ISSN 2277-2685
IJESR/April-June. 2026/ Vol-16/Issue-2/975-982
Dr P Deepthi *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

Dark web Leak and Malware Detector

Dr P Deepthi[1], K Poojitha[2], V Uma[3], E Vignetha[4]
[1] Associate Professor; Department of Computer Science and Engineering Bhoj Reddy Engineering College for Women Hyderabad India.
[2,3,4]B.Tech Student's; Department of Computer Science and Engineering Bhoj Reddy Engineering College for Women Hyderabad India.
Mail Id; eedunurivignetha17@gmail.com[4]
Accepted 01-06-2026

Abstract
The rapid expansion of digital platforms has significantly increased the risk of cybersecurity threats, particularly in the form of credential leakage and malware-based attacks. A major concern among users is the repeated use of passwords across multiple services, which amplifies vulnerability to large-scale data breaches. Additionally, stealthy malicious programs such as keyloggers, spyware, and script-based attacks can silently capture sensitive information, further compromising system integrity. Existing security solutions typically address either credential breach detection or malware identification independently, resulting in fragmented protection mechanisms that are less effective in educational and general user environments.To overcome these limitations, this study proposes DARK WEB LEAK AND MALWARE DETECTOR, an integrated, simulation-driven cybersecurity framework designed to identify both exposed credentials and potential malware threats within a unified platform. The system utilizes cryptographic hashing techniques, specifically SHA-256, to securely verify user credentials against a simulated compromised database without exposing plaintext information. In parallel, it incorporates heuristic analysis and pattern-based detection methods to identify suspicious files, behaviors, and processes.The proposed platform supports multiple scanning modes, including Quick Scan, Deep Scan, and Full System Analysis, enabling flexible threat detection based on user requirements. A risk scoring mechanism is introduced to evaluate overall cybersecurity exposure by combining credential leak status and malware detection results. Based on this assessment, the system generates personalized security recommendations to enhance user awareness and promote safer digital practices.Unlike conventional tools, the proposed framework prioritizes user privacy by avoiding direct interaction with real dark web sources and ensuring that sensitive data is never stored in an unencrypted format. By integrating simulation-based learning with practical security analysis, the system serves both as a protective tool and an educational platform, fostering improved cybersecurity awareness among users.

Keywords
Cybersecurity, Credential Leakage, Malware Detection, Dark Web Simulation, SHA-256 Hashing, Heuristic Analysis, Keylogger Detection, Spyware Detection, Risk Scoring System, Security Awareness Platform, Digital Safety, Intrusion Detection.

Introduction

The rapid expansion of internet-based services has significantly increased the exposure of users to cybersecurity threats, particularly credential leakage and malware attacks. Many individuals reuse passwords across multiple online platforms, which greatly amplifies the risk of unauthorized access in the event of a data breach. In addition to credential-related vulnerabilities, systems are increasingly targeted by malicious software such as keyloggers, spyware, and trojans that operate covertly to extract sensitive information without user awareness.

To address these concerns, the proposed system, DARK WEB LEAK AND MALWARE DETECTOR, is developed as a web-based cybersecurity platform that integrates both credential leak detection and malware analysis into a unified framework. The system applies cryptographic hashing techniques such as SHA-256 to securely transform user credentials into encrypted representations, which are then compared against a simulated database of compromised credentials. Simultaneously, the platform performs malware analysis by evaluating uploaded files or inputs using predefined signatures, heuristic rules, and behavioral patterns.

In addition to detection capabilities, the system provides users with a comprehensive risk assessment along with personalized security recommendations. This enables users to take corrective actions such as strengthening passwords or avoiding potentially harmful files. The platform is designed primarily for educational and awareness purposes, ensuring that no real dark web sources are accessed and that sensitive user data is not stored in plaintext, thereby maintaining privacy and ethical compliance while enhancing cybersecurity understanding.

Existing System

Current cybersecurity solutions typically address credential breaches and malware detection as separate functionalities. Services that monitor data breaches generally notify users when their credentials appear in known leak databases, while antivirus software focuses on identifying and removing malicious programs from computer systems. However, these tools operate independently and do not provide a unified security framework.

Furthermore, many existing solutions are commercial in nature, requiring subscriptions or advanced technical knowledge, which limits their accessibility for students and academic learners. Privacy concerns also arise, as certain platforms transmit user data to external servers for analysis. As a result, there is a clear lack of simple, integrated, and educational cybersecurity systems that combine both credential leak detection and malware analysis in a controlled and safe environment.

Proposed System

The proposed DARK WEB LEAK AND MALWARE DETECTOR system is designed to overcome the limitations of existing solutions by providing an integrated and user-friendly cybersecurity platform. It combines credential leak detection and malware analysis within a single system, enabling users to evaluate their digital security posture efficiently.

In this system, user credentials are securely processed using cryptographic hashing techniques such as SHA-256. These hashed values are then compared against a simulated database of leaked credentials to determine possible exposure. Concurrently, the system performs malware detection by analyzing uploaded files or inputs using signature-based methods, heuristic analysis, and multiple scanning modes, including Quick Scan, Deep Scan, and Full System Analysis.

The platform also incorporates a risk assessment module that evaluates both credential exposure and malware detection results to generate an overall security score categorized as Low, Medium, or High risk. Based on this evaluation, the system provides actionable recommendations to improve user security behavior. Importantly, the platform ensures privacy by avoiding real dark web interaction and by not storing sensitive data in plaintext format, making it both safe and educational.

Requirement Analysis

The functional architecture of the system is divided into user-side and system-side modules. The user module manages authentication and secure access to the platform, allowing users to register by providing basic details such as username, email, and password. The system also supports secure login and logout functionality to maintain session integrity. Users are required to input credentials for analysis during system operation.On the system side, multiple core functionalities are implemented. The credential leak detection module verifies whether user credentials appear in a simulated database of compromised data. The malware detection module analyzes uploaded files or inputs to identify potential threats. The system further generates SHA-256 hash values for secure credential processing and compares them with stored records. Finally, the results are displayed to the user in a clear format indicating whether the credentials or files are safe or compromised.

Life Cycle Model

The development of the system follows the Waterfall Model, a structured and sequential software engineering approach. In this model, each phase is completed before the next begins, ensuring clarity and disciplined development.The process begins with requirement analysis, where all functional and non-functional requirements are identified, including authentication, malware detection, and risk analysis. This is followed by system design, where the architecture is defined, including database structure, frontend layout, and backend logic.

In the implementation phase, the system is developed using Python Flask for backend logic and web technologies for the user interface. The testing phase ensures that all modules function correctly and that detection accuracy is maintained. After successful validation, the system is deployed for user interaction. Finally, maintenance is performed to fix issues, update detection patterns, and improve system performance over time.

Design

Architecture

The architecture of the proposed Dark Web Leak and Malware Detection System describes the overall structural design of the application and illustrates how different components interact during request processing. It provides a clear representation of how user inputs are received, processed through multiple functional modules, and finally transformed into meaningful outputs. The workflow typically involves sequential operations such as user authentication, credential verification, malware analysis, and generation of risk-based results, ensuring an organized and systematic execution of system functionalities. An architectural description serves as a formal blueprint of the system, helping in understanding the relationships and interactions among its core components**.**

Software architecture

Fig 1 Software architecture

The system architecture is broadly classified into two categories: software architecture and technical architecture. Software architecture refers to the high-level design of the system that defines its core structure and the interaction between different modules. It focuses on organizing system components in a way that enhances scalability, maintainability, flexibility, and reliability. In the proposed system, major modules such as authentication, credential leak detection, malware scanning, and risk assessment are designed to function independently while maintaining seamless integration, ensuring efficient overall performance.

Technical architecture

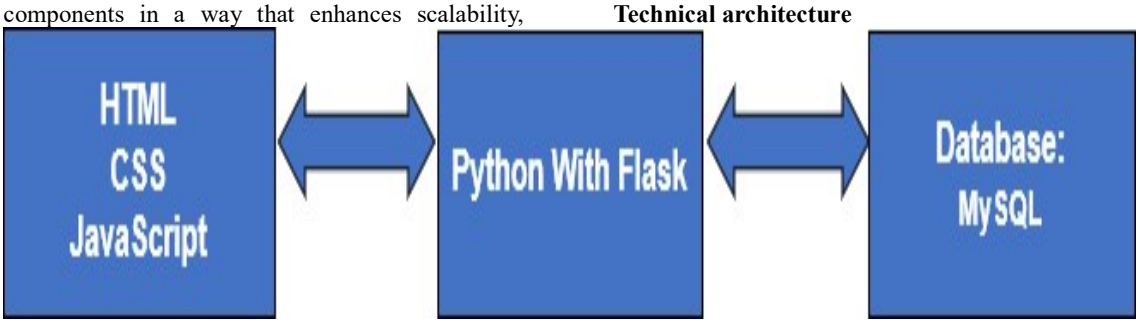


Fig 2 Technical architecture

Technical architecture, on the other hand, represents the implementation-level structure of the system. It specifies the technologies, frameworks, and tools used to develop and deploy the application. This includes frontend technologies such as HTML, CSS, and JavaScript, backend processing using Python Flask, and database management using MySQL. The technical architecture ensures proper integration of all system components, secure data handling, and efficient execution of operations. It also provides the foundation for system deployment, performance optimization, and future scalability, making the application robust and adaptable to evolving requirements.

Implementation

Technologies

The implementation of the proposed cybersecurity awareness platform is based on modern web technologies that ensure efficiency, security, and user-friendliness. The system is developed using HTML, CSS, JavaScript for the frontend, Python (Flask) for backend processing, and MySQL for database management. These technologies collectively support seamless integration between user interaction, data processing, and storage operations.HTML is used as the foundational markup language to structure the web application. It defines the layout of essential components such as login forms, registration forms, credential input fields, file upload sections, and result dashboards. CSS is used alongside HTML to enhance the visual presentation of the platform, ensuring a clean, responsive, and professional user interface. It controls design elements such as colors, spacing, typography, and layout alignment.JavaScript plays a key role in adding interactivity and dynamic behavior to the system. It is responsible for client-side validation, form handling, and real-time user interaction, improving the overall user experience. On the server side, Flask, a lightweight Python framework, handles backend operations such as user authentication, SHA-256 hashing of credentials, malware detection logic, file handling, and risk analysis computation. It acts as the bridge between the frontend interface and backend database operations, ensuring secure and efficient data processing.

MySQL is used as the relational database management system to store user credentials and simulated leaked data securely. It supports structured data storage and efficient querying, enabling fast comparison during credential verification processes.

Pseudocode (Workflow Description)

The system workflow begins with the login process, where users access the authentication page styled using a secure dark-themed interface. When a user submits login credentials, the system sends a POST request to the backend Flask server. The password is immediately converted into a SHA-256 hash to ensure secure handling of sensitive information. This hashed value is then compared with stored records in the user database. If a match is found, the system authenticates the user and initiates a session; otherwise, an error message is displayed.

During registration, the system validates user inputs such as full name, email format, and password confirmation. Once validated, the password is hashed using SHA-256 before being stored in the database, ensuring that no plaintext credentials are saved. Duplicate email entries are checked to prevent multiple registrations with the same identity.In the scanning module, users can input credentials or upload files for analysis. The system first performs credential leak detection by comparing hashed values against a simulated breach database. Simultaneously, uploaded files undergo malware analysis using pattern-based detection and heuristic rules. Different scan modes such as Quick Scan, Deep Scan, and Full Analysis determine the depth of inspection.Finally, the system aggregates both credential and malware analysis results to generate a risk score. This score is categorized into Low, Medium, or High risk levels, and presented to the user along with visual charts and security recommendations.

Frontend Design Overview

The frontend design consists of multiple user interfaces, including login, registration, scanning dashboard, and result visualization pages. Each page is designed with a consistent dark-themed layout to enhance readability and user focus. The login interface provides secure authentication with password visibility toggling and error handling mechanisms. The registration page ensures proper input validation and user-friendly account creation.The scanning dashboard serves as the main interface where users can perform credential checks and malware analysis simultaneously. It includes input fields for email, username, and password verification, along with a file upload section supporting multiple formats. A scan execution button triggers backend processing, and a loading interface displays real-time progress updates such as hashing, database comparison, file scanning, and risk evaluation steps.The result page presents the final output in a structured format, displaying risk levels, detection outcomes, and graphical representations using Chart.js. This ensures that users can easily interpret security status and take necessary preventive actions.

Testing

Software testing is a crucial phase in the development of the DARK WEB LEAK AND MALWARE DETECTOR, ensuring that all system functionalities operate correctly, securely, and efficiently. The primary objective of testing is to validate credential leak detection accuracy, malware identification reliability, and risk assessment consistency.Given the increasing dependency on digital systems for personal and academic use, ensuring system correctness is essential to prevent security vulnerabilities. Even minor errors in detection logic can significantly impact system reliability. Therefore, structured testing methodologies are applied to verify system behavior under different conditions and inputs.Testing also ensures cost efficiency, enhances product quality, improves system reliability, and strengthens overall security by identifying and resolving defects early in the development cycle.

Dimensions of Testing

The testing process is evaluated across multiple dimensions to ensure comprehensive validation. These include different layers of the application such as database operations, backend logic, and user interface performance. Testing is also categorized based on scale, including unit testing, module testing, integration testing, and system-level testing.Additionally, functional testing is performed to verify system outputs against expected results, while security testing ensures that sensitive user data is protected throughout the process. Manual testing methodologies are primarily used to validate system behavior from a user perspective.

Stages of Testing

Testing begins with unit testing, where individual components such as login authentication, credential verification, malware detection, and risk scoring modules are tested independently. This ensures that each unit functions correctly in isolation.Integration testing follows, where multiple modules are combined to evaluate their interaction and data flow. This phase ensures that authentication, database queries, and detection modules work seamlessly together without conflicts.System testing evaluates the complete application as a unified system. It verifies whether the platform meets all functional and security requirements, including proper credential analysis and malware detection in real-time scenarios.Finally, acceptance testing is conducted to validate system readiness from an end-user perspective. This ensures that the system meets user expectations and performs reliably before deployment.

Types of Testing

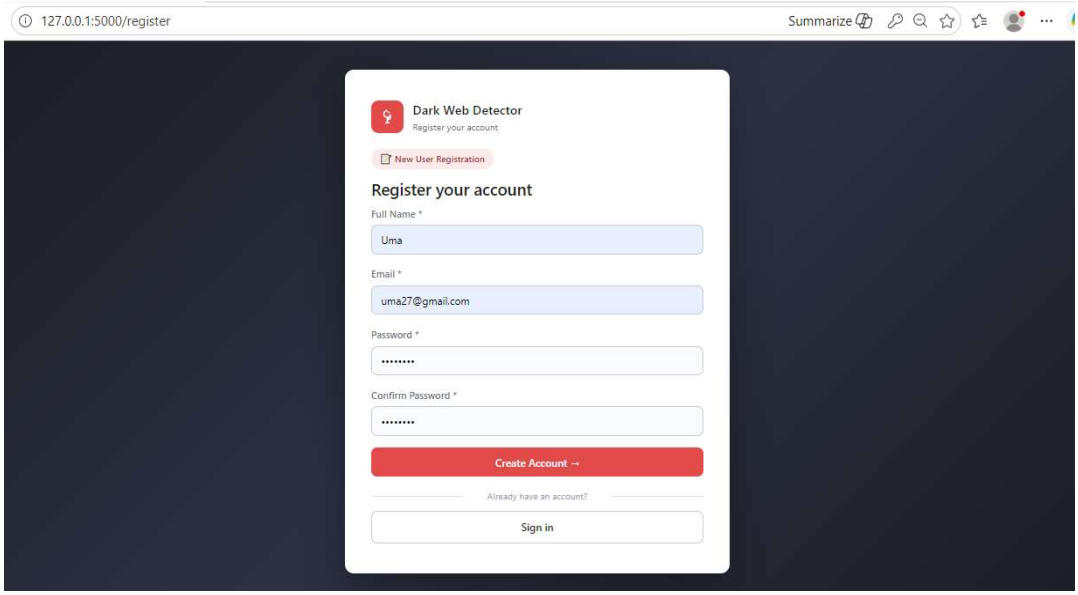
Black box testing is used to validate system functionality without considering internal code structure. Inputs such as credentials and files are provided, and outputs such as "Safe" or "Compromised" are verified for correctness.

White box testing focuses on internal logic and code structure. It is used to verify SHA-256 hashing, database comparison logic, malware pattern detection, and risk classification algorithms. Techniques such as statement coverage, branch coverage, and path coverage are applied to ensure complete logical validation.

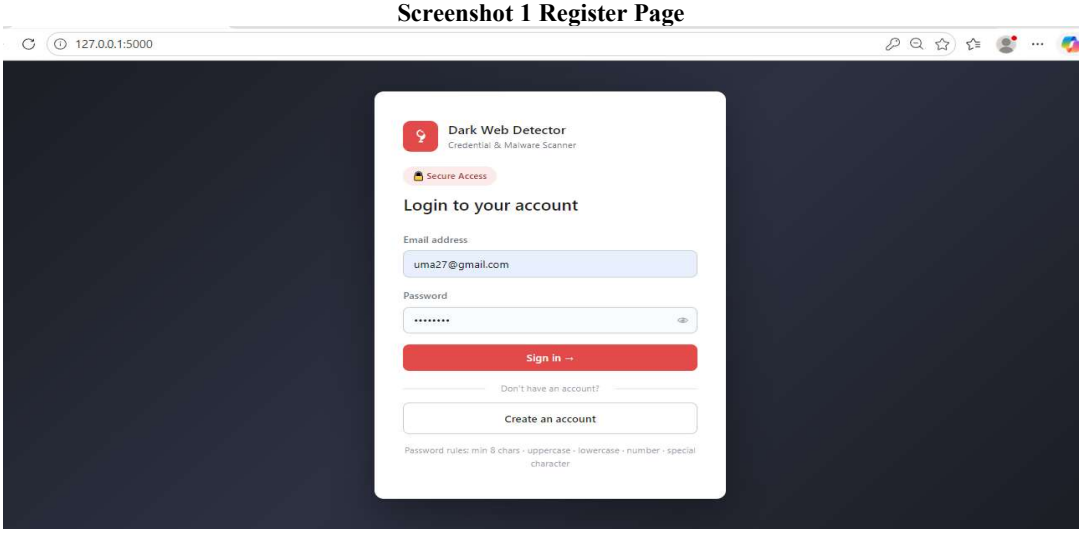
Test Cases

The system is evaluated using structured test cases covering authentication, credential detection, malware scanning, and risk classification. Functional verification includes validating correct login behavior, detecting leaked and safe credentials, identifying malicious and safe files, and ensuring accurate risk categorization. Additionally, dashboard visualization is tested to confirm proper graphical representation of results, including risk-level charts and analysis summaries. All test cases confirm expected behavior, ensuring that the system performs reliably under various input conditions.

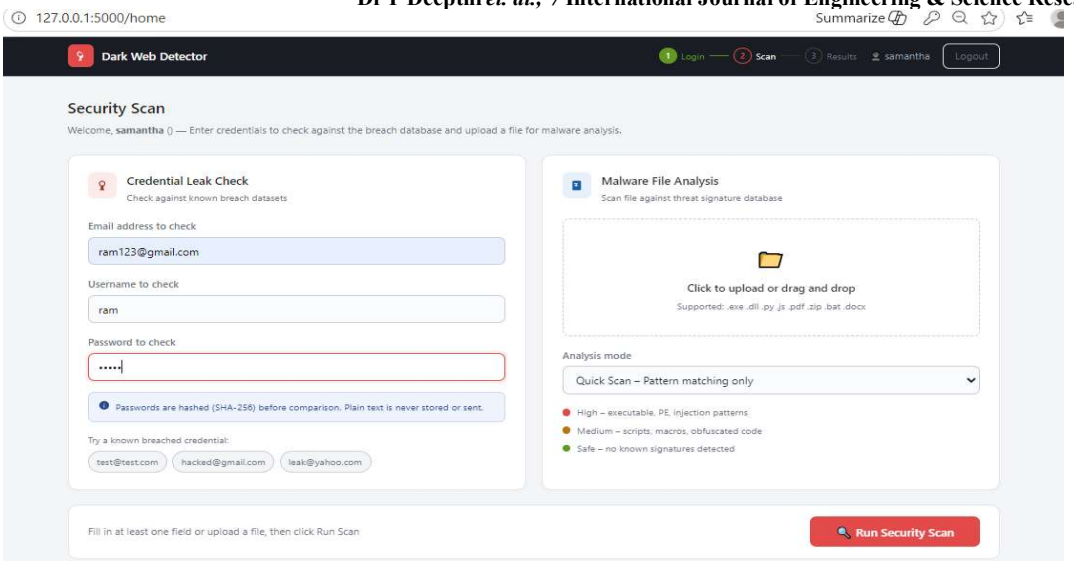
Screenshots



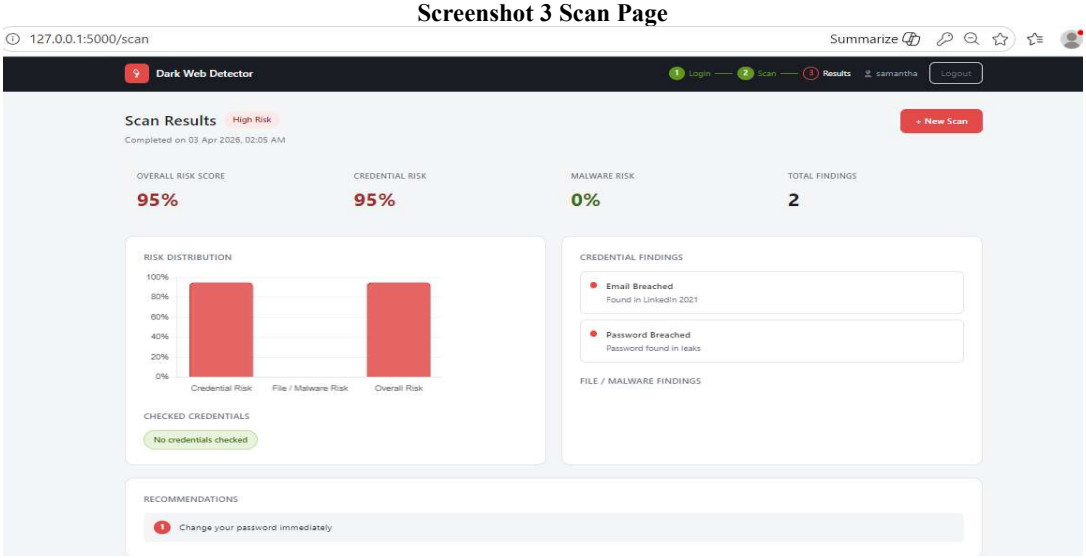
Screenshot 1 Register Page



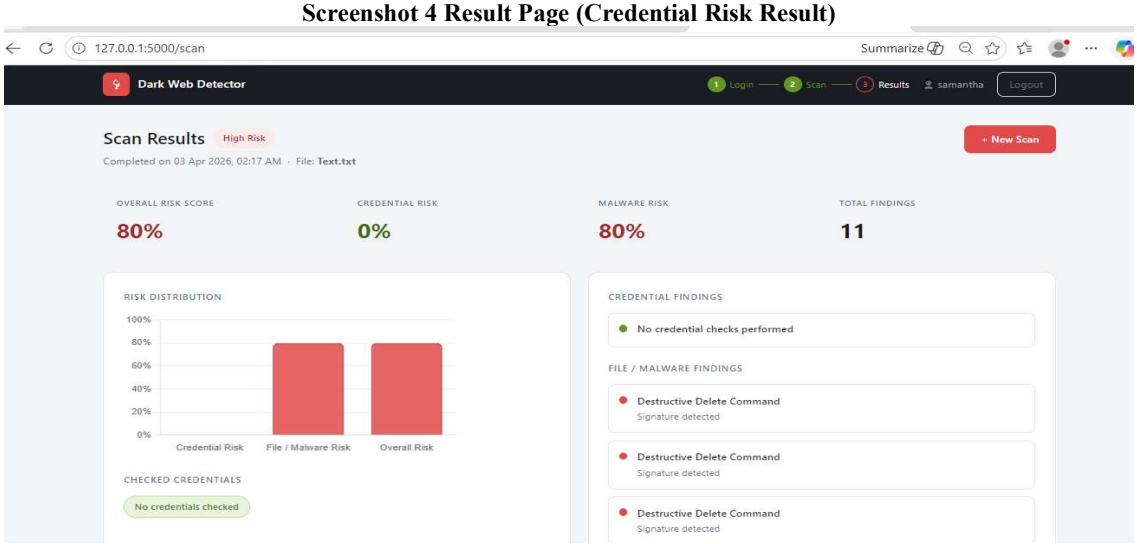
Screenshot 2 Login Page



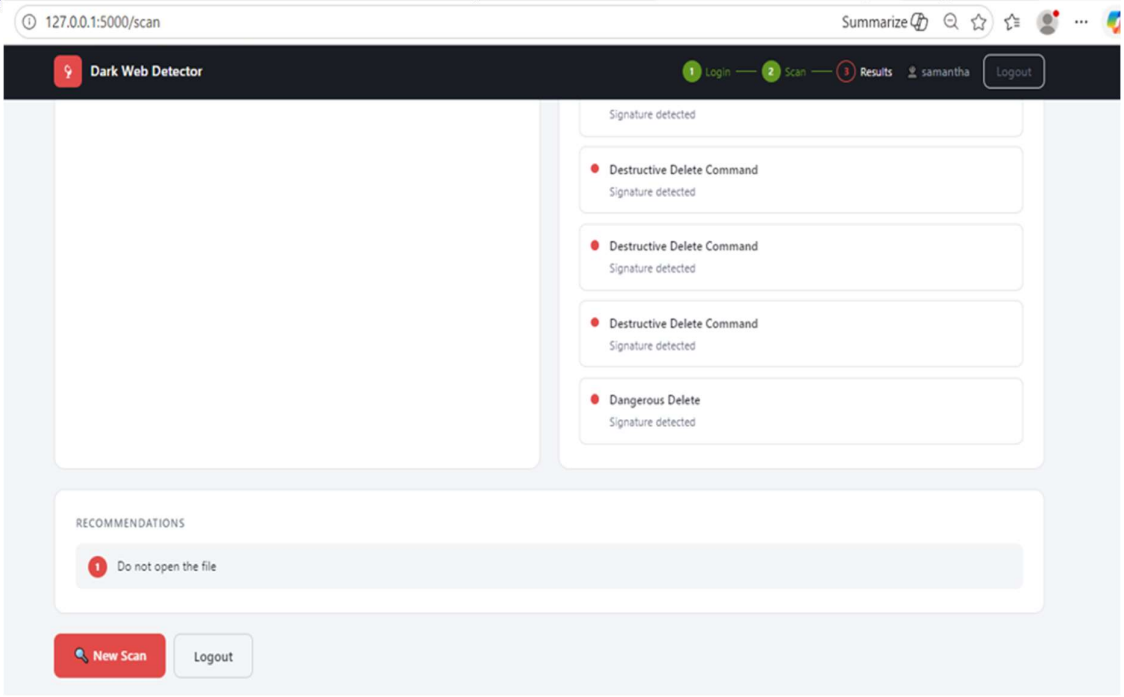
Screenshot 3 Scan Page



Screenshot 4 Result Page (Credential Risk Result)



Screenshot 5 Result Page (Malware Risk Result)



Screenshot 6 Result Page (Malware Risk Result)

Conclusion

The proposed Dark Web Leak and Malware Detection System presents an integrated cybersecurity framework aimed at improving digital safety through unified detection of credential leakage, malware threats, and risk evaluation. Unlike traditional security solutions that operate in isolation, the developed system consolidates multiple security functions into a single platform, thereby improving usability and effectiveness for end users.

The system successfully combines credential leak identification, file-based malware analysis, risk scoring, and personalized security recommendations within a structured and interactive environment. This integrated approach enhances user awareness regarding potential cyber threats while enabling proactive security management. By utilizing secure authentication mechanisms and cryptographic hashing techniques, the platform ensures that sensitive user information is processed safely without exposure of plaintext credentials.

Furthermore, the modular architecture of the system supports scalability and future enhancements, making it adaptable to evolving cybersecurity requirements. The inclusion of risk classification and recommendation features strengthens user understanding of digital threats and promotes safer online practices. Overall, the system contributes as both a practical security tool and an educational platform for improving cybersecurity awareness.

Future Scope

The future development of the system can be extended in several directions to improve its efficiency, scalability, and real-world applicability. One of the key enhancements includes the development of a mobile application version, enabling users to access cybersecurity protection services on smartphones and other portable devices. Additionally, deployment on cloud-based infrastructure can significantly improve system performance by supporting large-scale data processing, higher user traffic, and distributed computing capabilities.

Another important improvement involves the integration of real-time threat intelligence systems. By incorporating APIs connected to live cybersecurity databases and dark web monitoring services, the platform can be enhanced to detect newly emerging credential leaks and evolving malware threats in real time. This would significantly increase the system's responsiveness and accuracy.

Further advancements may also include the application of machine learning techniques for behavioral malware detection, enabling the system to identify unknown threats based on patterns rather than predefined signatures. These enhancements will transform the current simulation-based model into a more advanced, intelligent, and adaptive cybersecurity solution suitable for real-world deployment.

References

[1] A. Sharma, R. Verma, and S. Kulkarni, "Dark Web Monitoring and Data Breach Detection Using Machine Learning," *IEEE Access*, 2025.
[2] P. Reddy, V. Kumar, and S. Joshi, "Cyber Threat Intelligence Using OSINT for Dark Web Analysis," *IEEE Transactions on Information Forensics and Security*, 2024.
[3] K. Pandey, M. Singh, and R. Kaur, "Detection of Compromised Credentials Using Deep Learning Techniques," *IEEE Access*, 2023.
[4] S. Patel, A. Mehta, and D. Shah, "Malware Detection and Classification Using Static and Dynamic Analysis," *IEEE Access*, 2025.