© JAN 2026 | IRE Journals | Volume 9 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I7-1713339

Security Assessment Model for Software Systems

UZAMA TABAASSUM[1], SHREESAMPADA [2], LAKSHMI A[3], MANYA M[4], MANYA S E[5]
[1]*Asst Professor, Department of Computer Science & Engineering, Maharaja Institute of Technology, Mysore, Karnataka, India*
[2,3,4,5]*Student, Department of Computer Science & Engineering, Maharaja Institute of Technology, Mysore, Karnataka, India*

Abstract- ***The rapid growth of software systems and internet-based applications has increased the risk of cyber threats such as phishing attacks, unauthorized access, and malicious software execution. Traditional security mechanisms are often insufficient to detect dynamic threats in real time. This paper presents a *Security Assessment Model for Software System*, which integrates system monitoring, phishing detection, executable file analysis, and authentication log analysis into a unified web-based platform. The proposed system continuously monitors system processes, network activity, and resource usage while allowing users to analyze website URLs and uploaded executable files for potential threats. Authentication log analysis is used to detect suspicious login activities such as repeated failed attempts or unauthorized access. The system is developed using Python Flask for backend processing, SQLite for data storage, and machine learning–based decision logic for phishing detection. Experimental evaluation demonstrates that the system effectively identifies suspicious behavior and provides timely security alerts. The proposed model can assist system administrators in enhancing software system security by proactively detecting and mitigating threats.***

Index Terms- ***Authentication Log Analysis, Cyber Security, Phishing Detection, System Monitoring, Web Security.***

I. INTRODUCTION

With the increasing dependency on software systems and online services, cybersecurity has become a critical concern for organizations and individuals. Software systems are frequently targeted by cyber attackers through phishing websites, malicious executable files, and unauthorized login attempts. These attacks can result in data breaches, financial loss, and system compromise. Conventional security solutions such as antivirus software and firewalls are often limited in detecting emerging and sophisticated threats.

To address these challenges, this paper proposes a *Security Assessment Model for Software System* that combines real-time system monitoring with intelligent threat detection techniques. The model focuses on monitoring system processes, analyzing authentication logs, detecting phishing URLs, and evaluating executable files uploaded by users. By integrating these components into a single platform, the system provides a holistic view of the security status of a software system.

The proposed system uses a web-based interface for ease of access and usability. Python Flask is employed to manage backend operations, while SQLite is used to maintain security assessment records. Machine learning concepts are applied in the phishing detection module to classify websites as legitimate or malicious. This approach enables proactive identification of threats and supports effective security decision-making.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

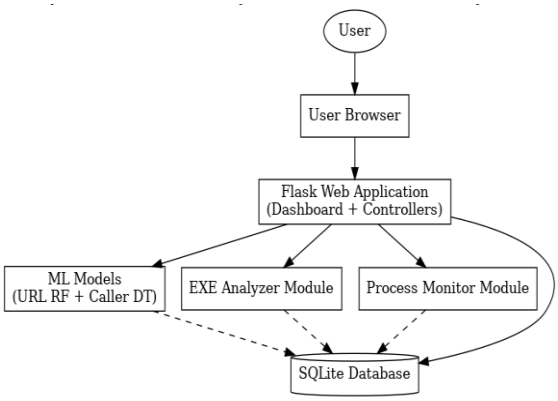
Several studies have explored different approaches for enhancing software system security. Phishing detection techniques using machine learning algorithms such as Decision Tree, Random Forest, and Support Vector Machines have shown promising results in classifying malicious URLs. System monitoring tools based on process and network behavior analysis are commonly used to detect abnormal system activities.

© JAN 2026 | IRE Journals | Volume 9 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I7-1713339

Authentication log analysis has also gained importance in identifying brute-force attacks and unauthorized access attempts. Researchers have proposed rule-based and machine learning–based methods to analyze login patterns and assess risk levels. However, most existing solutions focus on a single security aspect and lack an integrated framework.

The proposed system distinguishes itself by combining system monitoring, phishing detection, executable analysis, and authentication log analysis into a unified security assessment model. This integration improves threat visibility and enhances the overall security posture of the software system.

III. WRITE DOWN YOUR STUDIES AND FINDINGS



The system architecture consists of four major components:

1. User Interface Layer
Provides a web-based dashboard for user interaction, URL input, file uploads, and result visualization.

2. Application Layer (Flask Backend)
Handles request processing, feature extraction, rule-based and machine learning analysis, and system monitoring using psutil.

3. Security Analysis Modules
* Phishing Detection Module
* Executable File Analyzer
* Authentication Log Analyzer
* System Process and Network Monitor

4. Database Layer (SQLite)
Stores phishing detection history, log analysis results, and security assessment records.

The interaction between these components enables continuous security assessment and real-time threat detection.

IV. WORKING METHODOLOGY

4.1 System Monitoring Module
The system monitoring module continuously observes system processes, CPU usage, memory utilization, and network connections. Using the psutil library, it retrieves real-time system data and displays it on the dashboard. Users can identify suspicious processes and terminate them if required.

4.2 Phishing Detection Module
Users provide a website URL as input. The system extract features such as URL length, presence of special characters, HTTPS usage, and domain structure. These features are analyzed using decision logic based on machine learning principles. The system outputs whether the URL is phishing or legitimate along with a confidence score.

4.3 Executable File Analysis Module
The user uploads an executable file (.exe). The system computes the SHA256 hash and evaluates file metadata to determine potential risk. The result is displayed as a risk level indicator.

4.4 Authentication Log Analysis Module
Users upload authentication log files. The system parses log entries to extract usernames, IP addresses, timestamps, and login status. It identifies suspicious patterns such as repeated failed login attempts and generates a security assessment report.

V. DATASETS AND ALGORITHMS USED

Dataset:
The phishing detection module can be trained using publicly available datasets such as:
* UCI Phishing Websites Dataset
* Phish Tank Dataset
* Kaggle Phishing URL Dataset

Algorithm:
A *Decision Tree classification algorithm* is used for phishing detection due to its interpretability and efficiency. Rule-based logic is used for authentication log analysis and executable file assessment.

V. RESULTS AND OUTPUT

The system produces the following outputs:
* Real-time system process and resource usage display
* Phishing URL classification with confidence score
* Executable file risk assessment report
* Authentication log security assessment summary
* Historical records stored in SQLite database

These outputs assist administrators in identifying and mitigating security threats.

VI. CONCLUSION

This paper presents a Security Assessment Model for Software System that integrates multiple security mechanisms into a single platform. The proposed system effectively detects phishing attacks, suspicious executable files, abnormal system behavior, and unauthorized login activities. By combining system monitoring with intelligent analysis techniques, the model enhances the overall security of software systems. Future work may include integrating advanced machine learning models, real-time alerts, and cloud-based security analytics.

REFERENCES

[1] A. Alqahtani and M. Hussain, “An intelligent security assessment model for software systems using machine learning,” *IEEE Access*, vol. 9, pp. 1–12, 2021.

[2] H. Alqahtani and N. Alsharif, “A comprehensive security assessment framework for web applications using machine learning,” *IEEE Access*, vol. 11, pp. 1–15, 2023.

[3] R. Vinayakumar and K. P. Soman, “A comparative study of decision tree and random forest for cybersecurity intrusion detection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 876–888, 2020.

[4] Y. Du and J. Li, “Machine learning-based log analysis for security monitoring in software systems,” *IEEE Access*, vol. 10, pp. 1–13, 2022.

[5] M. H. Shahriar and A. Rahman, “Hybrid static and dynamic malware detection using random forest,” *IEEE Access*, vol. 9, pp. 1–14, 2021.