International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Safe Browser Extension for Real Time Safe Browsing

Perpetual Sheryl Fernando P*, Durga Veni G**
*(Student, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: perpetualsherylfernandop.ug22.cs@francisxavier.ac.in)
** (Assistant Professor, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: durgavenig@francisxavier.ac.in)

----------**********----------

Abstract:

Because web-based cyber threats are rapidly evolving, traditional browser security mechanisms often fail to provide real-time protection against malicious websites, phishing attacks, and unsafe downloads. This paper presents Safe Guardian, a browser-based security extension designed to proactively safeguard users by detecting and mitigating online threats during browsing activities. The system continuously monitors user interactions, including URL access and file downloads, and analyzes them using integrated threat intelligence services such as VirusTotal APIs.Upon identifying suspicious or malicious behavior, Safe Guardian immediately triggers a protection mechanism that blocks access to harmful resources and alerts the user through real-time warning notifications. The extension also provides a controlled browsing experience by maintaining a secure interaction layer that prevents further engagement with potentially dangerous content. This ensures that the user's system remains uncompromised while maintaining seamless browsing performance.The proposed implementation leverages event-driven monitoring, lightweight browser extension architecture, and external API-based threat analysis without requiring deep system-level modifications. Experimental observations demonstrate that the system effectively identifies threats with minimal latency and enhances user awareness of cybersecurity risks. The results validate the feasibility of browser-level proactive defense mechanisms as a scalable and user-friendly solution for improving everyday web security.

Keywords — **Browser Security, Cyber Threat Detection, Phishing Prevention, Safe Browsing, Real-Time Monitoring, Web Protection**

----------**********----------

I. INTRODUCTION

Over the past decade, web-based cybersecurity threats have evolved significantly, with attackers employing advanced techniques such as phishing attacks, malicious scripts, drive-by downloads, and deceptive URLs to exploit users. Traditional browser security mechanisms, including basic URL filtering and signature-based detection, often rely on predefined threat databases and reactive responses. As a result, modern threats can bypass these defenses, leading to data breaches, identity theft, and system compromise. To address these challenges, researchers and developers have increasingly focused on proactive web security strategies that emphasize real-time detection and prevention. One such approach involves integrating threat intelligence services and behavioral monitoring into browser environments to identify suspicious activities as they occur. By analyzing URLs, downloads, and user interactions dynamically, these systems aim to prevent users from accessing harmful content before damage occurs. The emergence of lightweight browser extensions has further enabled the deployment of user-centric security solutions. These extensions operate directly within the browsing environment, providing an additional layer of protection without requiring complex system-level modifications. However, many existing solutions lack real-time responsiveness, seamless user interaction, or

integration with advanced threat analysis platforms. This study introduces Safe Guardian, a browser-based cybersecurity solution designed to enhance user protection through continuous monitoring and intelligent threat detection. The system observes browsing activities such as website access and file downloads, and evaluates them using external threat intelligence services like VirusTotal. Upon detecting potentially malicious content, the extension immediately blocks access and notifies the user, ensuring a safe browsing experience while preventing system compromise. Built using modern web technologies and browser extension APIs, Safe Guardian operates with minimal performance overhead while delivering effective real-time protection. The proposed approach demonstrates that browser-level proactive defense mechanisms can serve as a practical and scalable solution for everyday cybersecurity challenges. Furthermore, it establishes a foundation for future advancements, including machine learning-based threat prediction and adaptive security responses, supporting the transition from reactive to intelligent web security systems.

II. OBJECTIVE

The primary objective of the Safe Guardian project is to design and develop an intelligent browser-based cybersecurity solution that enhances user protection through proactive web defense mechanisms. Unlike traditional browser security systems that rely on static filtering and reactive detection, this system aims to identify, analyze, and block malicious activities in real time, ensuring a safer browsing experience.

One of the key objectives is to create a robust threat detection mechanism capable of analyzing URLs, websites, and downloadable content. By integrating external threat intelligence services such as VirusTotal, the system evaluates potential risks associated with web resources and prevents users from accessing harmful or suspicious content, thereby reducing the chances of phishing attacks, malware infections, and data breaches.

Another important objective is to implement continuous real-time monitoring of user browsing behavior, including website visits and file downloads. The system is designed to detect anomalies and suspicious patterns using predefined rules and API-based analysis, enabling immediate response through warning notifications and automatic blocking of unsafe resources.

The project also focuses on developing a user-friendly interface that provides clear and instant feedback to users regarding potential threats. Through alert messages and visual indicators, the extension enhances user awareness of cybersecurity risks and encourages safe online practices without disrupting the overall browsing experience.

Additionally, the system aims to ensure lightweight and efficient performance by utilizing browser extension architecture. By avoiding deep system-level modifications and operating within the browser environment, Safe Guardian delivers effective protection with minimal resource consumption and seamless integration.

Finally, the project seeks to contribute to the advancement of web security by demonstrating how browser-level proactive defense strategies can improve everyday cybersecurity. It highlights the importance of integrating real-time threat intelligence and user-centric design to build scalable and practical security solutions.

In addition to these goals, the project aims to improve adaptability and scalability by supporting modular enhancements. The framework can be extended with advanced capabilities such as machine learning-based threat detection, behavioral analysis, and personalized security recommendations. This ensures that Safe Guardian remains effective against evolving web threats while supporting continuous innovation in modern cybersecurity practices.

III. METHODOLOGY

The overall processing pipeline of the proposed Safe Guardian system integrates real-time data acquisition, threat intelligence analysis, and user-centric protection mechanisms into a unified framework for secure web browsing. Information flows sequentially from user activity monitoring and preprocessing to threat evaluation, risk classification, and automated response generation, ultimately supporting real-time alerts and safe browsing decisions. This end-to-end transformation from raw browsing inputs to actionable security responses is illustrated in Fig. 1,which presents the complete system architecture of the proposed solution.

A. ***System Architecture:*** The operating workflow of the Safe Guardian system begins within the browser environment, where user activities such as website navigation and file downloads are continuously monitored. A Monitoring Engine captures browser events including URL requests, tab updates, and download triggers. If no suspicious activity is detected, the system allows normal browsing without interruption. However, once a potentially harmful URL or file is identified, the system immediately activates the Protection Protocol. At this stage, two processes occur simultaneously: the resource is analyzed using external threat intelligence services (such as VirusTotal APIs), and a warning or blocking interface is presented to the user. The architecture includes a modular design, where components such as monitoring, detection, and response operate independently while maintaining seamless interaction. Event-driven triggers ensure efficient use of system resources by activating security checks only when required. Additionally, logging mechanisms capture detected threats and user actions, enabling further analysis and improvement. Overall, the architecture enhances browser security by combining real-time detection, intelligent analysis, and proactive response within a unified framework. Furthermore, the system architecture is designed to support extensibility and interoperability with emerging security technologies. By adopting a modular and API-driven approach, the framework can seamlessly integrate additional services such as advanced threat intelligence platforms, machine learning-based classifiers, and cloud-based security analytics.

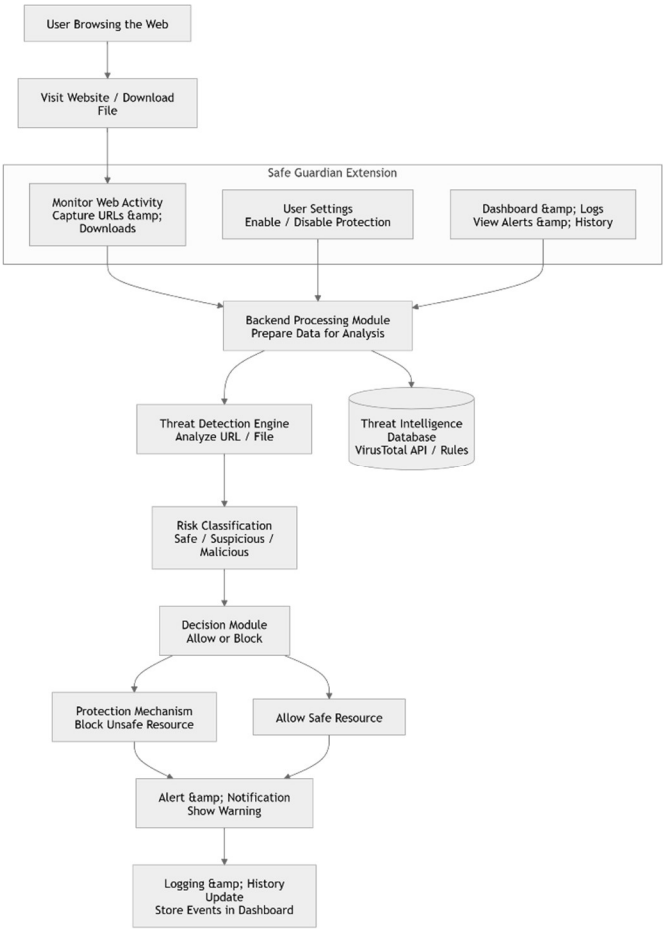

Fig. 1*. Block diagram of the proposed system*

B. ***Monitoring Engine:*** The Monitoring Engine serves as the primary data acquisition component of the Safe Guardian system, continuously observing user browsing behavior in real time. It captures activities such as URL access, page transitions, and file download requests using browser extension APIs.

Instead of relying on static filtering alone, the engine performs dynamic tracking of user interactions and forwards relevant data for analysis. A background process ensures continuous monitoring without affecting browser

performance. Designed as a lightweight and non-intrusive module, the Monitoring Engine operates within the browser environment without modifying system-level components. Structured logging of events provides traceability and supports further threat analysis. By continuously capturing and forwarding activity data, the engine enables early detection of suspicious behavior.

C. *Threat Detection Engine*: The Threat Detection Engine acts as the core analytical component of the Safe Guardian system, responsible for identifying potential security risks. It evaluates URLs, domains, and downloadable content using external threat intelligence services such as VirusTotal.

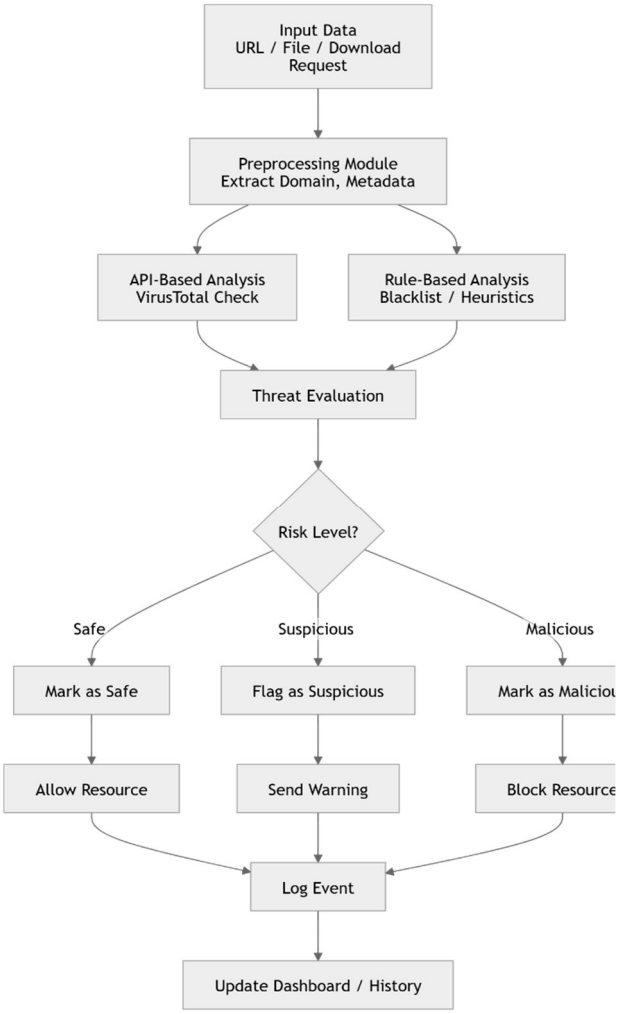

Fig.2. *Threat Detection Engine – Risk Analysis*

Unlike traditional signature-based approaches, this engine combines API-based verification with rule-based filtering to detect phishing websites, malicious domains, and unsafe downloads. Each resource is analyzed in real time, and a risk level is assigned based on threat intelligence results. The engine operates in an event-driven manner, ensuring rapid analysis and minimal latency. Upon detecting suspicious or malicious content, it triggers the protection mechanism. Additionally, structured logs are maintained for each analyzed event, enabling auditability and continuous improvement of detection strategies.

D. *Protection and Alert Mechanism:* The Protection and Alert Mechanism functions as the response layer of the Safe Guardian system. When a threat is confirmed, the system immediately prevents further interaction with the unsafe resource. This is achieved by blocking access to malicious websites, cancelling unsafe downloads, or displaying warning notifications to the user. The alert interface provides clear information about the detected threat, helping users understand the risk and avoid unsafe actions. Designed to be user-friendly and minimally disruptive, this module ensures that security measures do not negatively impact the browsing experience. By combining automated blocking with real-time alerts, the system effectively mitigates threats while enhancing user awareness.

E. *Logging and Safe Browsing Mechanism:* The Logging and Safe Browsing module represents the final stage of the Safe Guardian workflow. It maintains detailed records of detected threats, blocked URLs, and user interactions for analysis and future improvements. In addition to logging, the system ensures safe browsing by preventing harmful resources from being executed or accessed. Instead of allowing threats to reach the system, Safe Guardian isolates them at the browser level by blocking interaction at the source. The module is designed with strict isolation principles, ensuring that malicious content does not affect the user's system. By combining real-time protection with comprehensive logging, this component

completes the security cycle while preserving system integrity.

IV. RESULTS AND DISCUSSION

The Safe Guardian prototype was experimentally evaluated to assess its effectiveness in detecting web-based threats and providing real-time protection during browsing activities. The evaluation focused on three major aspects: accuracy of threat detection, responsiveness of the protection mechanism, and overall impact on user experience. Additionally, the coordination between monitoring, detection, and alert generation was analyzed to determine system reliability and performance.

1. Browser Extension Interface

This interface represents the primary interaction point between the user and the Safe Guardian system. It provides easy access to core functionalities such as enabling/disabling protection, viewing alerts, and checking browsing safety status. The clean and user-friendly design ensures that users can quickly understand and interact with the system without technical complexity.

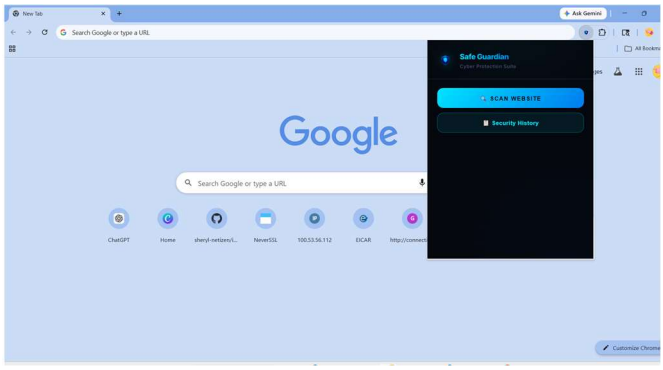

Fig.3. *Safe Guardian Extension Interface*

2. Security Dashboard

The dashboard demonstrates the system's capability to present security-related data in a centralized and organized manner. It displays information such as detected threats, blocked websites, and recent activity logs. The presence of real-time updates confirms the effectiveness of continuous monitoring and enhances user awareness of potential risks.

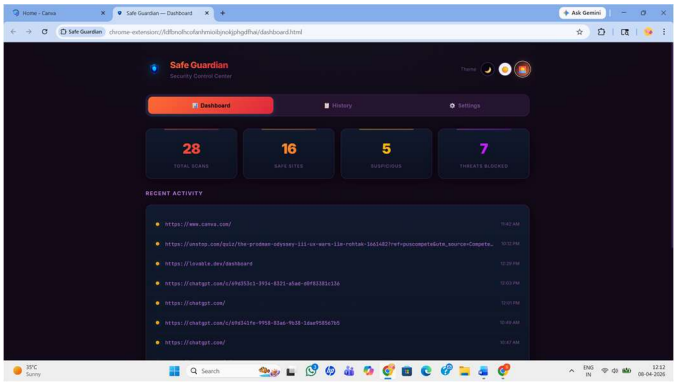

Fig. 4. *Security Dashboard – Monitoring Overview*

3. URL Threat Detection Result

The results indicate that the system successfully identifies and classifies URLs based on their safety levels. By integrating external threat intelligence services, Safe Guardian accurately detects phishing websites and malicious domains. The classification into safe, suspicious, and malicious categories validates the effectiveness of the detection engine in real-world browsing.

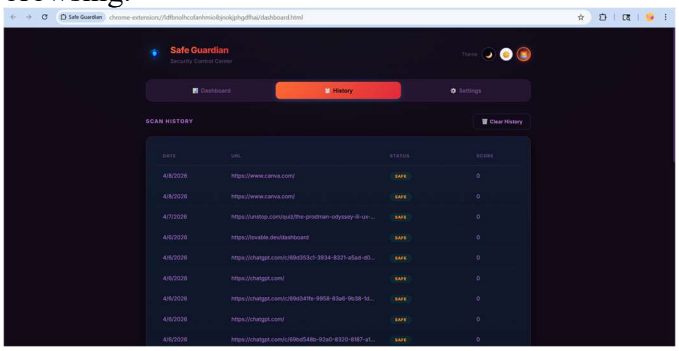

FIG. 5. *URL THREAT DETECTION OUTPUT*

4. Protection Settings and Threat Prevention

This module demonstrates the system's ability to provide customizable security controls for enhanced user protection. Features such as Web Shield, Download Scanner, and Notifications enable real-time detection and prevention of malicious websites and unsafe files. The Web

Shield continuously scans visited URLs and blocks harmful content, while the Download Scanner ensures that files are analyzed before being accessed. Additionally, notification alerts inform users immediately when a threat is detected, improving awareness and response. The inclusion of Child Safety Mode further strengthens protection by restricting access to inappropriate or unsafe content, making the system suitable for controlled browsing environments.

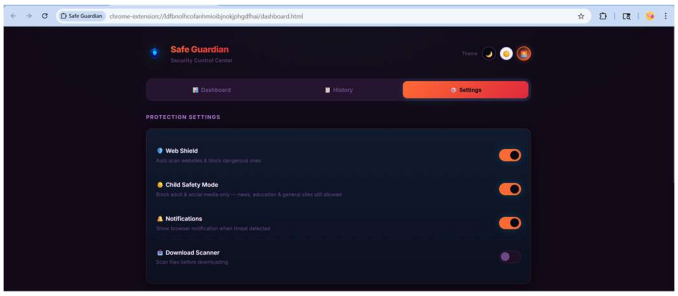

Fig. 6. *Protection Settings – Threat Prevention Interface*

5. Productivity and Focus Management

This section highlights the system's capability to promote focused and distraction-free browsing. Features such as Block Gaming Sites and Screen Time Reminder help users manage their online habits effectively. The system can restrict access to entertainment platforms like gaming and streaming websites, ensuring improved productivity. Additionally, the screen time reminder feature provides timely alerts after prolonged usage, encouraging healthier browsing behavior. The customizable reminder duration adds flexibility, allowing users to personalize their experience according to individual needs.

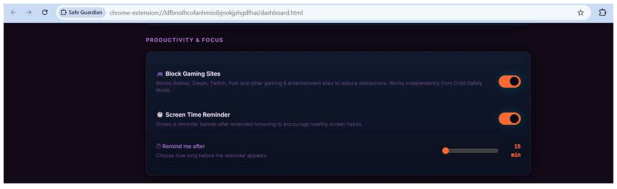

Fig. 7: *Productivity Settings –Focus Control Features*

6. Customization and User Preferences

The customization module enhances user experience by providing control over accessibility and interface preferences. The Allowed Sites feature enables users to whitelist trusted websites, ensuring uninterrupted access even if they are flagged by security filters. This improves usability while maintaining flexibility in protection mechanisms. Furthermore, the Appearance Settings, including theme selection (Dark, Light, and Sunset modes), allow users to personalize the interface according to their visual preferences. These features collectively ensure that the system remains user-friendly while offering adaptable security controls.

V. REFERENCES

[1] A. Kumar and K. Singh, "Real-time phishing detection using machine learning techniques," IEEE International Conference on Computing, Communication and Automation, pp. 1–6, 2018.

[2] J. Ma, L. Saul, S. Savage, and G. Voelker, "Learning to detect malicious URLs," ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 3, pp. 1–24, 2019.

[3] S. Wang, N. Zhang, and R. Beyah, "Behavior-based malware detection in virtualized systems," IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 4, pp. 1719–1732, 2020.

[4] Z. Dou, I. Khalil, A. Khreishah, and A. Al-Fuqaha, "A systematic review of software-based web phishing detection," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2797–2819, 2020.

[5] S. Syed, H. Banerjee, and M. Rahman, "Phishing detection using machine learning and feature extraction techniques," IEEE Access, vol. 9, pp. 124–135, 2021.

[6] S. Rohmat et al., "Phishing detection and prevention using Chrome extension," IEEE International Symposium on Digital Forensics and Security, pp. 1–6, 2022.

[7] Z. Yang, J. Liu, and K. Zhao, "Host-level anomaly detection using behavioral modeling," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1301–1313, 2022.

[8] V. Chavan et al., "Phishing detection using

MACHINE LEARNING AND CHROME EXTENSION," IEEE INTERNATIONAL CONFERENCE ON ADVANCES IN INFORMATION TECHNOLOGY, 2024.

[9] K. DHIVYA ET AL., "SIMULATING PHISHING AND DETECTING FAKE URLS USING BROWSER EXTENSION," JOURNAL OF CYBER SECURITY, VOL. 2, NO. 4, PP. 19–28, 2024.

[10] ARYA NADH T. S. ET AL., "PHISHSNAP: CHROME EXTENSION FOR PHISHING DETECTION USING MACHINE LEARNING," JOURNAL OF ARTIFICIAL INTELLIGENCE AND CAPSULE NETWORKS, VOL. 6, NO. 1, PP. 105–121, 2024.

[11] K. BARIK, S. MISRA, AND R. MOHAN, "WEB-BASED PHISHING URL DETECTION USING DEEP LEARNING TECHNIQUES," INTERNATIONAL JOURNAL OF DATA SCIENCE AND ANALYTICS, VOL. 20, PP. 4449–4471, 2025.

[12] SHYNI SHAJAHAN ET AL., "BROWSER EXTENSION FOR PHISHING WEBSITE DETECTION USING MACHINE LEARNING," INTERNATIONAL CONFERENCE ON INNOVATIONS IN COMPUTING, PP. 1–6, 2025.

[13] M. DANDOTIYA ET AL., "REAL-TIME IDENTIFICATION OF PHISHING ATTACKS THROUGH MACHINE LEARNING ENHANCED BROWSER EXTENSIONS," SCIENTIFIC REPORTS, VOL. 16, 2026.

[14] G. NERURKAR AND A. NAIK, "REAL-TIME PHISHING WEBSITE DETECTION USING BROWSER EXTENSION WITH RANDOM FOREST CLASSIFIER," INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY, VOL. 15, NO. 3, 2026.

