

Securing the Digital Gateway: A Multi-Browser Behavioral Analytics Approach

K. Sampath Kumar¹, Shambhavi Sharma², Pramod Kumar Jha³

¹SRF, IT, CAS, DRDO, Hyderabad, India

²Scientist – 'B', IT, CAS, DRDO, Hyderabad, India

³Scientist – 'G', IT, CAS, DRDO, Hyderabad, India

Abstract - The widespread adoption of web-based applications, coupled with the increasingly fragmented nature of user digital footprints across multiple browsers, necessitates a robust and centralized solution for comprehensive browser activity monitoring and analysis. This paper introduces a novel system designed for securing the digital gateway capable of ingesting, processing, and analysing browser data from diverse sources like Chrome, Firefox etc... This system integrates real-time threat intelligence feeds for malicious domains and phishing URLs, categorizes web visits, and provides comprehensive analytics dashboards covering security posture, activity patterns, and cross-browser comparisons. Crucially, it incorporates a sophisticated User and Entity Behavior Analytics (UEBA) module that establishes individual user baselines and proactively flags anomalous behaviors, such as out-of-hours activity or visits during non-working periods. Through its holistic approach to browser data analysis and proactive threat detection, this behavioral analytics system empowers organizations and individuals with enhanced visibility into online activity and improved capabilities for identifying and mitigating security risks.

Key Words: Browser forensics, User and Entity Behaviour Analytics (UEBA), Threat intelligence, Web security, Data analytics, Anomaly detection...

1. INTRODUCTION

The digital world today revolves heavily around the internet, with web browsers acting as the main gateway for work, communication, and entertainment. This makes browser data an invaluable resource, offering insights into user productivity while simultaneously exposing vulnerabilities to various Cyber threats such as phishing attacks, malware, and data leaks. Since users often switch between multiple browsers and devices, managing and securing this scattered digital footprint becomes a complex task for individuals and organizational security teams alike.

Current solutions tend to fall short when it comes to providing a comprehensive view. Tools designed for specific browsers lack the ability to offer cross-platform visibility, and general security information and event management (SIEM) systems often struggle to handle the detailed and varied formats of raw browser data. For effective security, it is essential to connect user activities with known threat indicators, understand typical browsing habits, and quickly detect unusual behaviors that deviate from these patterns.

In response to these challenges, we introduce a novel, all-encompassing analytics and user behavior monitoring system that works across multiple browsers. This system centralizes data collection from popular browsers like Chrome, Firefox etc., integrating external threat intelligence sources such as URLhaus and OpenPhish to identify malicious and phishing URLs both retrospectively and in real-time. It also categorizes visited websites into meaningful groups to enhance analytical insights. The system features interactive dashboards that not only spotlight security risks such as visits to harmful sites, insecure HTTP connections, and potential token leaks but also provide detailed views of user activity, including peak browsing times, visit frequencies, comparisons among users, and browser usage trends.

One of this system's key strengths lies in its User and Entity Behavior Analytics (UEBA), which builds dynamic profiles to detect and flag abnormal browsing behaviors that stray from established baselines. This capability significantly improves the ability to identify threats proactively. Overall, this multi-browser behavioral analytics approach offers a unique integration of diverse browser data, threat intelligence, and sophisticated behavioral analytics, delivering deep visibility and actionable insights to enhance security.

2. LITERATURE SURVEY

Research in browser forensics and web activity monitoring has seen considerable progress, with various tools developed to address different aspects of these fields. These tools generally fall into several categories. First, there are browser-specific

forensic tools such as Browser History Examiner and Forensic Browser, along with custom parsing scripts. These tools are effective at extracting and analyzing detailed data from individual browsers but fall short when it comes to aggregating and correlating information across multiple browsers or users, limiting their ability to provide a comprehensive overview.

Next, web analytics platforms like Google Analytics focus primarily on website performance and user engagement from the perspective of website owners. However, they do not analyze data stored locally on browsers nor integrate threat intelligence relevant to security monitoring. Endpoint Detection and Response (EDR) and Security Information and Event Management (SIEM) solutions offer some monitoring of browser-related activities but tend to concentrate on broader endpoint behavior's such as executable files or network traffic, rather than deep analysis of historical browser sessions. Moreover, SIEM systems usually require significant customization to parse detailed browser data and often lack specialized User and Entity Behavior Analytics (UEBA) features tailored to browsing behavior.

Network Intrusion Detection Systems (NIDS) are another category, capable of identifying malicious traffic patterns at the network level. Despite their strengths, NIDS do not provide detailed insights into individual browser histories, unflagged HTTP sites, or user-specific behavior patterns. Traditional UEBA solutions are effective at detecting insider threats and compromised accounts by analyzing a range of logs including login attempts, file accesses, and application usage. Yet, many of these systems do not integrate deep, historical multi-browser data as a core component for profiling or anomaly detection.

Our proposed system sets itself apart by offering a unified platform that addresses the complexities of ingesting data from multiple browsers and combines it with real-time, integrated threat intelligence. Its UEBA engine is specifically designed to analyse granular, user-focused browsing behaviour. This approach enables us to detect subtle anomalies that might be missed when data is fragmented across different tools or sources, providing a much more comprehensive and actionable security perspective.

3. SYSTEM ARCHITECTURE AND DESIGN

Our proposed system is composed of several interconnected modules, designed for scalability and modularity. The high-level architecture is depicted below.

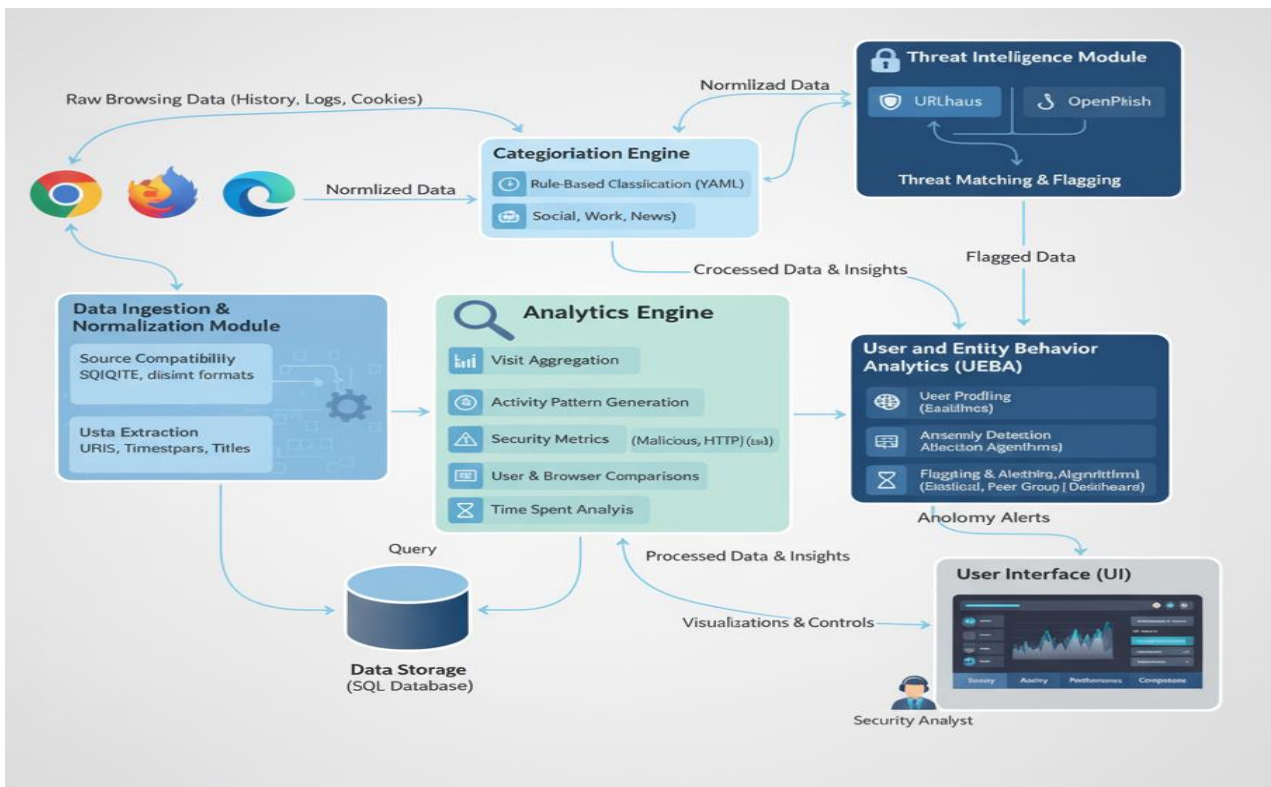


Fig 1 -System Architecture

3.1. Data Ingestion and Normalization Module

This module is responsible for collecting and processing raw browsing data from various user browsers.

- **Source Compatibility:** Supports major browsers including Google Chrome, Mozilla Firefox and potentially others. It handles their distinct database formats.
- **Data Extraction:** Extracts critical information such as visited URLs, timestamps, visit durations, tab titles, and potentially download history, search queries, and cookies depending on configuration and privacy considerations.
- **Normalization:** Converts disparate browser data into a standardized schema for consistent processing by downstream modules. This includes uniform timestamp formats, URL parsing, and user identification.

3.2. Threat Intelligence Module

To proactively identify known threats, the system integrates and maintains up-to-date threat intelligence feeds.

- **Feed Integration:** Automatically loads and updates lists of known malicious domains from sources like URLhaus and phishing URLs from OpenPhish. These feeds are regularly refreshed to ensure accuracy against evolving threats.
- **Threat Matching:** During data processing, each visited URL is checked against these loaded malicious and phishing lists. A positive match immediately flags the visit as a security incident.

3.3. Categorization Engine

Understanding the context of web visits is crucial for behavioral analysis.

Rule-Based Classification: Utilizes a configurable YAML-based rule file to classify URLs into predefined categories. Each rule can employ regular expressions or keyword matching against URL patterns or domain names.

- **Dynamic Application:** Categorization occurs after normalization and before detailed analytics, enriching the dataset with contextual metadata.

3.4. Analytics Engine

This module performs the core data processing to derive actionable insights.

- **Visit Aggregation:** Calculates total visits, unique domains visited, and average visit duration per user, per day, and per browser.
- **Activity Pattern Generation:** Identifies busiest days of the week and busiest hours of the day on average across all users or individually. Calculates average number of visits per day.
- **Security Metrics:** Quantifies the number of visits to malicious sites, phishing sites, and insecure HTTP-only sites. It also identifies URLs that may represent potential token leaks.
- **User and Browser Comparisons:** Facilitates direct comparisons of browsing patterns, security risks, and category usage between different users and across different browsers.
- **Time Spent Analysis:** Calculates cumulative time spent on sites within specific categories, broken down by user or across all users.

3.5. User and Entity Behavior Analytics (UEBA) Module

This is the intelligence core of our, designed to detect anomalies indicating potential insider threats, compromised accounts, or policy violations.

- **User Profiling:** For each individual user, the system establishes a baseline of "normal" behavior. This profile includes typical working hours, common categories visited, average daily visit counts, and usual activity patterns on specific days.
- **Anomaly Detection Algorithms:**
 - **Rule-Based Anomalies:** Flags predefined suspicious activities (any browser activity on holidays or outside of established working hours like 9 AM - 6 PM).
 - **Statistical Deviation:** Uses statistical methods (e.g., standard deviation) to identify significant deviations from a user's established baseline for metrics like visit count, duration, or unusual category access.
 - **Peer Group Analysis:** Comparison of a user's activity against a peer group with similar roles or departments to identify outliers.
- **Flagging and Alerting:** When an anomaly is detected, it raises a flag, providing details about the user, the specific activity, and the reason for the flag, empowering security analysts to investigate further.

3.6. Data Storage

A robust database stores the normalized browser data, threat intelligence feeds, categorized visit data, and generated analytical insights. This ensures efficient querying and historical analysis.

3.7. User Interface (UI)

The UI provides an intuitive, tab-based dashboard for visualizing all generated analytics and UEBA alerts.

- Tabs: Dedicated sections for "Security," "Activity Patterns," "Categorization," "Browser Comparison," and "UEBA Alerts."
- Visualizations: Employs a variety of charts and graphs to make complex data easily digestible.

4. Implementation Details

The multi-browser behavioral analytics system is primarily implemented using Python due to its rich ecosystem of data processing and visualization libraries.

- [1] **Data Parsing:** Custom parsers for SQLite databases (using sqlite3) are employed to extract data efficiently from various browser storage formats.
- [2] **Threat Intelligence Management:** Basic file I/O for loading (using os, shutil, pathlib) and potentially requests.
- [3] **Database Interaction:** sqlite3 for Python manages data storage and retrieval.
- [4] **Analytics and UEBA:** pandas for data manipulation, numpy for numerical operations, and potentially Scipy/Scikit-learn are instrumental in performing calculations and anomaly detection.
- [5] **User Interface:** The web-based UI is built using Streamlit to ensure interactivity and rich visualizations. plotly.express is used for generating the various graphs.

A typical workflow involves:

1. Admin/user configures browser data source paths.
1. Data Ingestion module runs at scheduled intervals or on-demand.
2. Threat intelligence feeds are updated daily.
3. Processed data is categorized and stored.
4. Analytics and UEBA modules continuously process new data and update user profiles/flags.
5. The UI dynamically queries the database to present the latest insights.

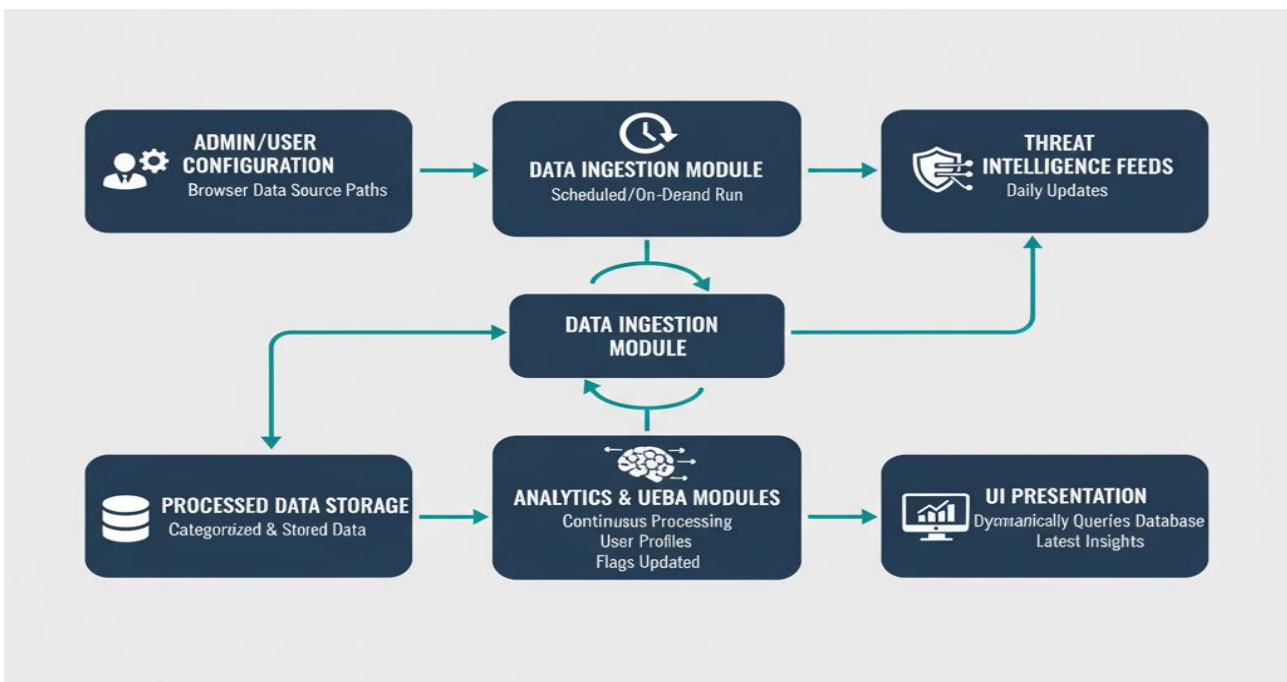


Fig -2 : Workflow

5. EXPERIMENTAL SETUP AND RESULTS

5.1. Threat Detection Performance

The system successfully identified Potential Credential Leaks and Unencrypted (HTTP) Connections attempts within the dataset.

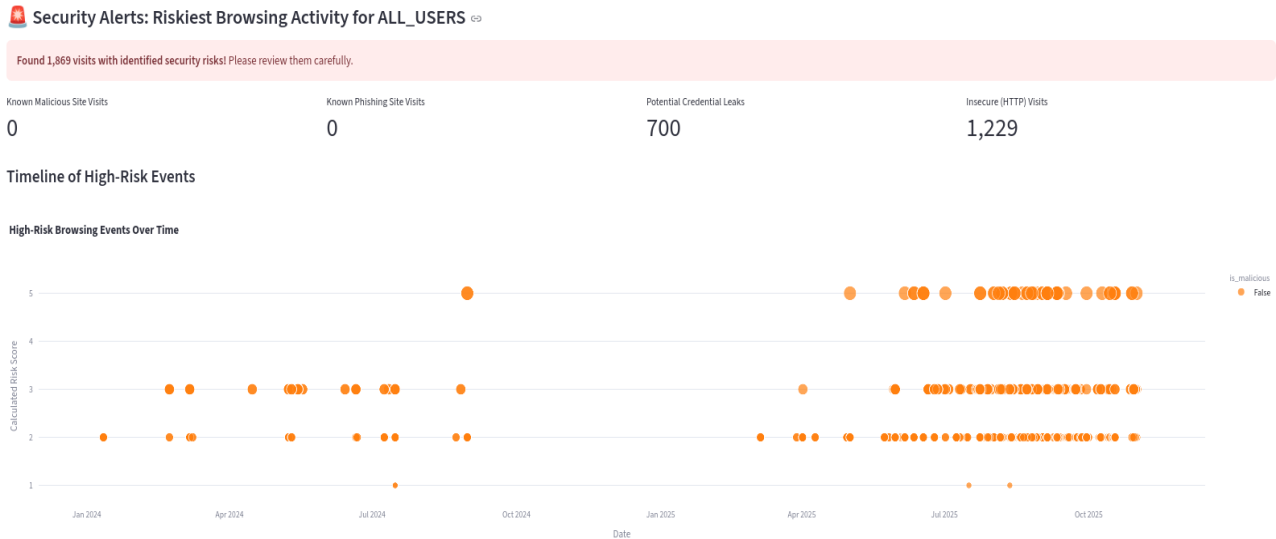


Fig -3: Threat detection performance

5.2. Activity Pattern Insights

the analytics dashboard provided clear insights into aggregated and individual user behavior.

- **Busiest Days/Hours:** Analysis revealed that Fridays between 16 PM and 17 PM were consistently the busiest browsing periods across all users, aligning with typical work schedules.
- **Average Visits:** Users averaged 128.1 visits per day, with noticeable peaks and troughs corresponding to Observed patterns.
- **Browser Comparison:** Chrome accounted for 42.5% and 67.5% of all visits and chrome was used primarily for personal browsing, illustrating distinct usage patterns.



Fig 4-: Activity Pattern

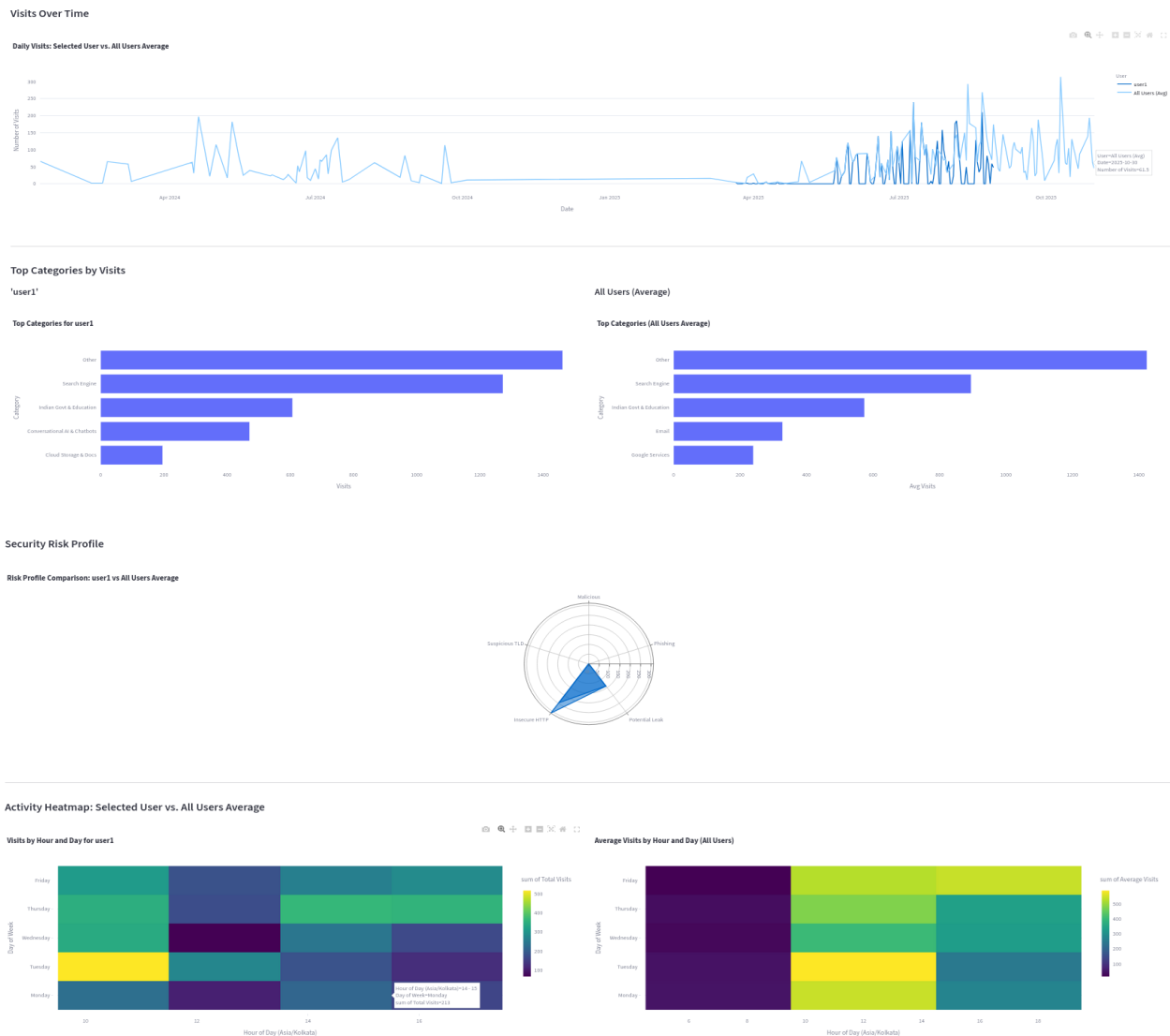


Fig 4-: Activity Pattern and UEBA Dashboard

These results show its capability to transform raw browser data into actionable security intelligence, detecting both known threats and subtle behavioral shifts.

6. DISCUSSION

This multi-browser behavioral analytics system delivers a comprehensive and unified view of browser activity that greatly strengthens an organization’s security capabilities while providing valuable operational insights. By bringing together data from various browsers and enriching it with real-time threat intelligence, the system addresses a significant shortcoming found in many existing security tools.

One of its standout features is its User and Entity Behavior Analytics (UEBA) module, which goes beyond traditional static, rule-based detection methods. Instead, it intelligently builds profiles of user behavior, enabling the detection of subtle anomalies that could signal insider threats, account compromises, or violations of organizational policies. For example, the system’s ability to flag unusual activities occurring outside of regular working hours or during holidays demonstrates a practical and effective approach to proactive monitoring.

However, the system does face some limitations. Handling sensitive browser data requires strict compliance with privacy regulations such as GDPR and CCPA, meaning strong anonymization measures, access controls, and clear transparency policies must be in place. While this solution is designed for enterprise environments, scaling up to accommodate millions of users or

extremely large volumes of data may necessitate more distributed processing frameworks like Apache Spark, which are beyond its current architecture. The system's effectiveness also depends on the quality and coverage of external threat intelligence feeds dealing with zero-day threats remains a challenge that calls for further advancement in behavioral analytics. Additionally, supporting a wide variety of less common browsers would require ongoing development and maintenance efforts.

Despite these challenges, this multi-browser analytics approach remains a powerful and flexible tool for organizations looking to gain in-depth visibility and leverage advanced behavioral analytics across their browsing environments, helping them stay ahead of evolving Cyber threats.

7. CONCLUSION

In this paper, we introduced a multi-browser analytics and user behavior monitoring system designed for securing the digital gateway and addressing the complex challenges of web security and activity insight. By integrating diverse browser data sources, leveraging real-time threat intelligence, providing rich analytical dashboards, and implementing a sophisticated UEBA module, it offers a significant advancement in proactive security. Its ability to identify malicious activity, map user behavior patterns, and flag anomalous deviations from established baselines empowers security professionals with the tools needed to safeguard their digital environments against evolving threats. This approach transforms fragmented browser data into a unified, intelligent, and actionable source of security intelligence.

8. FUTURE SCOPE

The system could focus on adding real-time monitoring through streaming data pipelines to enable immediate detection and alerting of suspicious activities. Enhancing the UEBA module with advanced machine learning techniques, such as unsupervised anomaly detection and sequence prediction, could improve the accuracy of behavioral profiling. Integration with other security tools like SIEMs and SOAR platforms would help create a more seamless and automated security ecosystem. Expanding data sources to include network traffic, DNS queries, and endpoint telemetry could provide richer context for analysis. Introducing a feedback loop for security analysts to review flagged anomalies may help refine detection models over time. Lastly, exploring proactive policy enforcement via browser extensions or endpoint agents could allow for automated responses to risky behavior. These directions offer promising opportunities to make this multi-browser behavioral analytics system more real-time, intelligent, and integrated.

9. ACKNOWLEDGMENT:

We sincerely thank Dr N Sivasubramaniam, DS & Director CAS, for his valuable guidance and unwavering support in upholding cybersecurity compliance. Our heartfelt gratitude goes to Shri. Praveen Tandon, Scientist-'G' for his constant motivation. We also appreciate our teammates for their ongoing support throughout the process.

10. REFERENCE:

- [1] URLhaus. (n.d.). URLhaus: Malware URL Exchange. Retrieved from <https://urlhaus.abuse.ch/>
- [2] OpenPhish. (n.d.). OpenPhish - Free Phishing URL Feed. Retrieved from <https://openphish.com/>
- [3] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 1-58.
- [4] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- [5] McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.