CyberPeace editor@cyberpeace.net

THE CYBERPEACE JOURNAL

Supported by United Service Institution of India

Academic Partner Shah & Anchor Kutchhi Engineering College

Maltrail: Intrusion Detection System

Dev Bhadra
Cyber security Department
Shah and Anchor Kutchhi Engineering College
Mumbai, India
dev.17305@sakec.ac.in

Akshat Mandot
Cyber security Department
Shah and Anchor Kutchhi Engineering College
Mumbai, India
akshat.mandot17573@sakec.ac.in

Satvik Trivedi
Cyber security Department
Shah and Anchor Kutchhi Engineering College
Mumbai, India
satvik.17041@sakec.ac.in

Dr. Shwetambari Borade
Cybersecurity
Shah and Anchor Kutchhi Engineering College
Mumbai, India
shwetambari.borade@sakec.ac.in

***Abstract*— MALTRAIL is an advanced intrusion detection system designed to safeguard networks against cyber threats in today's dynamic digital landscape. By integrating traditional methods with cutting-edge technologies, MALTRAIL offers a comprehensive defense mechanism against unauthorized access attempts, malicious activities, and emerging cyber threats. This research explores the methodologies, technologies, and comparative analysis employed in the development and implementation of MALTRAIL. Through a thorough examination of intrusion detection methods, including signature-based, anomaly-based, and machine learning-based approaches, the project identifies the strengths and weaknesses of each method, providing valuable insights into their effectiveness, scalability, and computational complexity.**

***Keywords*—Maltrail, Network security, Traffic analysis, Intrusion detection, Cybersecurity, Blacklist, Malware detection**

I. INTRODUCTION

Maltrail is an innovative open-source network traffic detection system designed to identify and monitor malicious activities and anomalies within a network. Leveraging a combination of publicly available blacklists, threat reports, and malware trails, Maltrail offers a comprehensive and robust solution for real-time threat detection [1]. This system is capable of identifying a wide range of malicious activities, including malware infections, command and control (C2) communications, data exfiltration, and other forms of cyber threats by meticulously analyzing network traffic and comparing it against a database of known malicious indicators.

Maltrail is designed with user accessibility and ease of deployment in mind, making it suitable for both small and large-scale network environments. Its detailed logging and alerting capabilities ensure that security professionals can quickly respond to potential threats, while its integration with other security tools enhances overall threat intelligence and incident response strategies. By providing a scalable, efficient, and powerful tool for network defense, Maltrail significantly strengthens an organization’s cybersecurity posture, helping to safeguard critical infrastructure against evolving cyber threats.

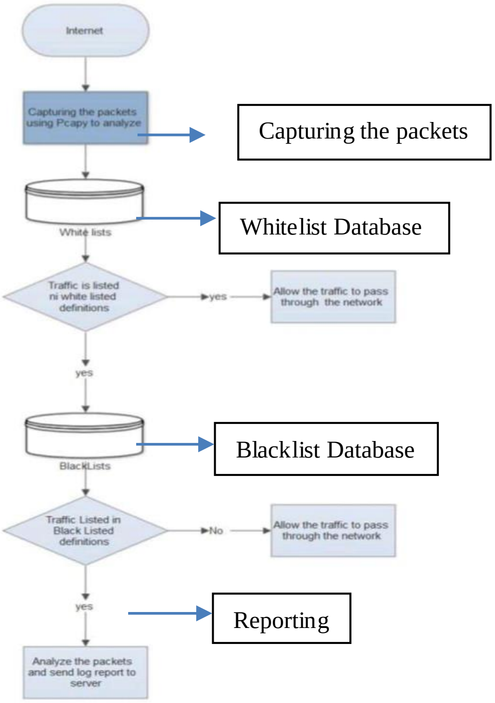
II. LITERATURE REVIEW

The literature review provides a comprehensive overview of prior studies and research efforts in the field of intrusion detection systems (IDS). It traces the evolution of IDS methodologies from early signature-based detection to more sophisticate anomaly-based and hybrid approaches [2]. Key advancements, such as machine learning techniques and behavioral analysis, are highlighted, along with the challenges and limitations faced by existing IDS solutions. The review identifies several knowledge gaps in the field, including the need for adaptive detection mechanisms, scalability and performance enhancements, improved contextual understanding of network behavior, and better integration with other security controls. These knowledge gaps underscore the importance of ongoing research and innovation in IDS development to address the evolving threat landscape effectively.

III. METHODOLOGY

The research aims to develop an AI-driven tool for real-time intrusion detection, providing immediate alerts to users while browsing the internet to mitigate the risk of attacks and safeguard sensitive information. The Chrome extension employs various features extracted from URLs and webpage content to detect phishing attempts, utilizing a prediction algorithm based on weighted features for classification. Upon detection, users receive pop-up alerts, and data, including the URL and user ID, is sent to a Node.js server for storage and further analysis in an Oracle Database. The server facilitates communication with the Chrome extension, which serves as an interface for user management and interaction with the backend server [3]. Communication among the components involves the exchange of data flows, including intrusion detection results and user information. Error handling mechanisms are implemented at various stages to ensure system robustness, addressing potential issues with intrusion detection, backend processing, and database interactions.

Overall, the methodology integrates phishing detection, backend communication, data handling, error handling, and Android app integration to provide a comprehensive solution for detecting phishing risks and enhancing user safety while collecting data for continuous analysis and improvement [4].


Internet
Capturing the packets using Pcapy to analyze
Capturing the packets
White lists
Whitelist Database
Traffic is listed ni white listed definitions
yes
Allow the traffic to pass through the network
yes
BlackLists
Blacklist Database
Traffic Listed in Black Listed definitions
No
Allow the traffic to pass through the network
yes
Reporting
Analyze the packets and send log report to server

IV. AIM

The project aims to develop a robust tool for phishing detection and prevention, with a focus on employing the Agile methodology for its iterative and collaborative approach. Agile's flexibility and adaptability will be crucial in navigating evolving requirements and challenges effectively. It will foster frequent communication, short development cycles, and continuous testing to ensure the project's success.

To manage the project's complexity, well-defined milestones will be established to guide the implementation process. These milestones include data collection and preparation, where a diverse dataset of phishing and legitimate URLs will be collected, preprocessed, and cleansed to remove inconsistencies. The Chrome extension development phase will focus on designing an intuitive user interface for seamless integration into the browser. The extension will process URLs in real-time, providing instant feedback to users while ensuring lightweight design to minimize resource consumption.

Integration with Kavach - The Shield, will be a critical component of the project, with meticulous attention to detail. A robust and secure API will be designed for communication between the extension and Kavach, facilitating real-time data synchronization for continuous threat updates.

Regarding the real-time images and GUI, the extension will prompt users for login upon loading, storing data in the Oracle server if logged in, and prompting new users for registration [5]. The Node.js server will assist in appending data into the database, where login credentials and phishing predictions will be stored along with the username and visited URLs.

Ongoing monitoring and maintenance will be essential for the tool's continued effectiveness. This includes regular threat database updates to incorporate new phishing patterns and malicious URLs, as well as performance monitoring to address any deviations promptly.

Scalability is integral to the project's design, with cloud-based infrastructure facilitating seamless scalability as the user base grows [6]. The project will also be flexible to accommodate future enhancements, such as additional threat detection capabilities or support for more platforms.

In conclusion, the project's approach encompasses Agile methodology, well-defined milestones, robust integration with Kavach, intuitive user interface design, ongoing monitoring and maintenance, and scalability for future enhancements, all aimed at developing an effective phishing detection and prevention tool.

V. RESULT

Maltrail will depends on the architecture with the flow like Traffic which is the real-time and then the sensor which monitorzs the traffic in the device and then the server which then logs the results. Sensor is an stand-alone script which is running in the end devices in moniter mode to inspect the packets which are moving in and out. It mainly works with an principle of PCapy an python script to inspect the web traffic . This then controls the low of logs to the server where the black listed traffic have to be necessarily forwarded to the server which is the centralized one accurately for example an honeypots.

VI. CONCLUSION

Reflecting on the project, valuable insights have been gained into the intricacies of intrusion detection systems and their role in bolstering cybersecurity defenses. By analyzing *the performance of the IDS in real-world scenarios*, important observations and findings have emerged, shedding light on the effectiveness of various detection techniques and algorithms.to mobile users, allowing them to manage accounts and access phishing.

ACKNOWLEDGMENT

The Completion of this research would not have been possible without the mentorship and guidance of several individuals. We would like to express my sincere gratitude to our mentor, Dr. Shwetambari Borade and Dr. Nilakshi Jain, Head of Cybersecurity Department for their constant support and insightful feedback throught this project Also, we are grateful to Dr. Bhavesh Patel, Shah and Anchor Kutchhi Engineering College for his unwavering support and for providing the resources necessary to conduct this research.

REFERENCES

[1] Anderson, J. P. (1980). Computer Security Threat Monitoring and Surveillance. https://csrc.nist.gov/files/pubs/conference/1998/10/08/proceedings-of-the-21st-nissc-1998/final/docs/early-cs-papers/ande80.pdf

[2] Denning, D. E. (1987). An Intrusion-Detection Model. https://ieeexplore.ieee.org/document/1702202

[3] Axelsson, S. (2000). The Base-Rate Fallacy and the Difficulty of Intrusion Detection. https://dl.acm.org/doi/pdf/10.1145/357830.357849

[4] Lippmann, R. P., et al. (2000). Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation. https://ieeexplore.ieee.org/document/821506

[5] Forrest, S., et al. (1996). A Sense of Self for Unix Processes. https://asu.elsevierpure.com/en/publications/sense-of-self-for-unix-processes

[6] Lee, W., & Stolfo, S. J. (2000). Data Mining Approaches for Intrusion Detection. https://www.researchgate.net/publication/2428488