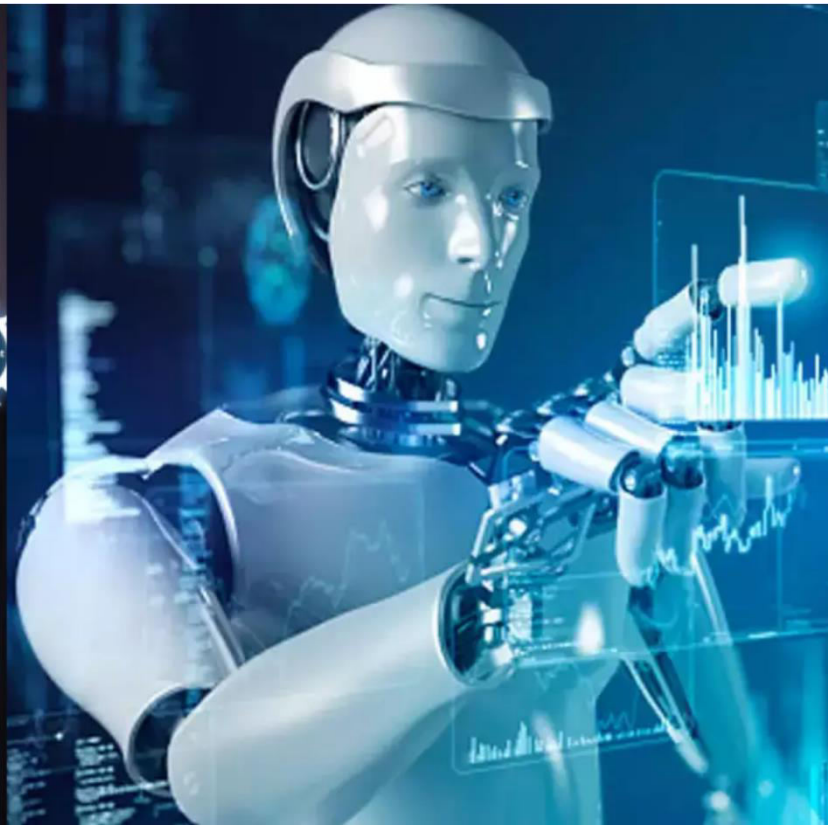
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

Agent-Less Windows System Vulnerability and Network Scanner

Prof. Suma Manjunath N, Priya Prabhakar, Yashaswini K

Assistant Professor, Department of Information Science and Engineering, S.J.C. Institute of Technology, Chickballapur, Karnataka, India

U. G Student, Department of Information Science and Engineering, S.J.C. Institute of Technology, Chickballapur, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S.J.C. Institute of Technology, Chickballapur, Karnataka, India

ABSTRACT: This project addresses the growing need for intelligent, non-intrusive cybersecurity solutions by introducing an "Agentless Windows System Vulnerability and Network Scanner" that identifies and predicts potential threats in real-time. The system leverages a comprehensive dataset of packet-level communication data, including protocol types (TCP, UDP), flags (SYN, ACK, RST, FIN), services (HTTP, DNS, SSH, FTP, NTP), source and destination IP addresses, port numbers, packet sizes, and sender/receiver identifiers. These features enable the detection of diverse attack types such as Phishing, DoS, DDoS, Man-in-the-Middle, SQL Injection, Cross-Site Scripting (XSS), Ransomware, Password Attacks, and Zero-Day Exploits. Machine learning models—Decision Tree, Random Forest, and XGBoost—are employed to analyze patterns and anomalies within the data, improving the accuracy and speed of threat identification. The front-end is developed using HTML, CSS, and JavaScript for seamless user interaction, while the back-end uses Python for data processing and model execution. By eliminating the need for endpoint agents, this project offers a scalable and efficient approach to securing Windows-based networks through automated, real-time threat detection and vulnerability assessment.

KEYWORDS: CyberThreat ,Machine Leanring, Anomaly detection, Network Vulnerability Scanning

I. INTRODUCTION

This project addresses the growing need for proactive and scalable cybersecurity solutions in the face of increasingly sophisticated cyber threats. Traditional security systems often rely on agent-based approaches, requiring the installation of monitoring software on each endpoint. While effective to a certain extent, these systems are prone to issues such as high resource consumption, limited scalability, and susceptibility to tampering by attackers. To overcome these limitations, this project proposes an agentless system that scans Windows-based systems and networks for vulnerabilities and potential cyber threats without installing additional software on client machines. The core idea is to provide a lightweight, centralized security solution that ensures real-time monitoring and efficient detection of malware activities.The system leverages a comprehensive dataset containing packet-level network communication data. Key features of this dataset include protocol types (e.g., TCP, UDP), flags (SYN, ACK, FIN, RST), packet types (HTTP, FTP, DNS, SSH), as well as metadata such as IP addresses, port numbers, packet sizes, and unique identifiers for each sender-receiver pair. This rich data source enables a detailed analysis of network behavior and aids in the identification of unusual or suspicious activities. The target labels in the dataset represent a variety of common and advanced cyber threats such as Phishing, DDoS, SQL Injection, Man-in-the-Middle attacks, Ransomware, Cross-Site Scripting (XSS), and Zero-Day Exploits.To analyze and predict these threats, the system incorporates machine learning models including Decision Tree, Random Forest, and XGBoost. These models are trained to detect patterns and anomalies in the data, providing a robust foundation for threat prediction. The backend of the system, implemented in Python, manages data processing and model execution, while the frontend—built using HTML, CSS, and JavaScript—offers a user-friendly interface. This approach allows for real-time, intelligent threat detection and strengthens the overall cybersecurity posture of Windows networks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

II. PROBLEM STATEMENT

In today's rapidly evolving digital landscape, Windows-based networks are increasingly targeted by a wide range of cyber threats such as phishing, ransomware, DDoS attacks, and zero-day exploits. Traditional cybersecurity solutions often rely on endpoint agents that require installation and maintenance, which can be resource-intensive, intrusive, and difficult to scale across large networks. Moreover, these solutions may not provide real-time visibility into network-level vulnerabilities, making it challenging to detect and respond to threats proactively. Therefore, there is a critical need for an intelligent, agentless system that can continuously monitor network traffic, identify vulnerabilities, and detect malicious activities in real-time without disrupting normal operations or requiring changes to endpoint devices.

III. LITERATURE REVIEW

1. Smith and Williams (2020)
Title: A Survey on Machine Learning for Network Intrusion Detection
Outcome: This study explored various machine learning techniques, highlighting Decision Tree and Random Forest as effective models for classifying network traffic as normal or malicious. These models showed high accuracy in detecting common attack types like DoS and DDoS.
Disadvantages: The research lacked implementation details for real-time detection and did not address the agentless approach, limiting its practical deployment in live network environments.

2. Johnson et al. (2021)
Title: Cyber Threat Detection Using XGBoost and SVM
Outcome: The paper compared the performance of XGBoost and SVM for cyber threat detection, concluding that XGBoost provided better classification accuracy. It showed effectiveness in identifying threats such as malware and phishing through analysis of network traffic.
Disadvantages: The study relied on static datasets and did not evaluate the scalability or performance of the models in real-time or high-throughput network conditions.

3. Patel and Shah (2022)
Title: Phishing Attack Detection Using Machine Learning Algorithms
Outcome: Focused on detecting phishing attacks, this study applied Decision Tree and Random Forest algorithms to analyze features like IP addresses and URL patterns. Random Forest outperformed Decision Tree in accuracy and consistency.
Disadvantages: The approach was limited to phishing detection and did not consider broader threats or the integration of the solution into an agentless, real-time monitoring system.

4. Chen et al. (2020)
Title: Anomaly Detection in Network Traffic Using XGBoost
Outcome: This study used XGBoost to detect anomalies in network traffic that could indicate threats such as SQL Injection and DDoS attacks. It offered a scalable solution capable of real-time threat detection.
Disadvantages: While promising, the research did not fully address the performance impact on large-scale enterprise networks or offer insight into deployment without endpoint agents.

5. Wang and Lee (2021)
Title: Malware Detection in Network Traffic Using Machine Learning
Outcome: Applied Random Forest and SVM models to detect malware in packet-level data. Both models showed effectiveness in identifying known and some unknown threats by analyzing packet size, protocol type, and IP information.
Disadvantages: The study did not include results for live monitoring and lacked adaptability for zero-day threats or evolving malware patterns.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

6. Singh and Kumar (2021)
Title: Real-Time Malware Detection Using Decision Tree and Random Forest
Outcome: Presented a real-time detection model using Decision Tree and Random Forest, achieving fast and accurate identification of malware in ongoing network traffic.
Disadvantages: While real-time capable, the solution was not agentless and required integration with endpoint monitoring tools, which could increase overhead and complexity.

7. Sharma et al. (2020)
Title: Detection of SQL Injection and XSS Attacks Using Machine Learning
Outcome: This study focused on identifying SQL Injection and Cross-Site Scripting (XSS) attacks using machine learning techniques, particularly XGBoost and Decision Tree. The results showed that XGBoost achieved higher detection accuracy compared to traditional methods, effectively identifying code injection threats in network traffic.
Disadvantages: The study focused narrowly on specific attack types (SQLi and XSS), limiting its generalizability across other modern and complex cyber threats.

8. Kim and Cho (2022)
Title: A Comparative Analysis of Machine Learning Models for DDoS Detection
Outcome: This research compared the effectiveness of Random Forest and XGBoost in detecting Distributed Denial of Service (DDoS) attacks. XGBoost outperformed Random Forest due to its scalability and better pattern recognition, making it suitable for high-volume network traffic analysis.
Disadvantages: The paper concentrated solely on DDoS detection and did not evaluate the models' performance in identifying a wider spectrum of network attacks or their integration into a real-time detection system.

9. Zhao et al. (2020)
Title: Identifying Ransomware Threats Using Machine Learning Models
Outcome: The study applied XGBoost and SVM to detect ransomware in network traffic. XGBoost provided superior results in classification accuracy and real-time detection capabilities, effectively identifying patterns indicative of ransomware behavior.
Disadvantages: The solution's reliance on labeled datasets may limit its effectiveness against new, evolving ransomware variants not represented in training data.

10. Hernandez and Lee (2021)
Title: Zero-Day Exploit Detection in Network Traffic Using Machine Learning
Outcome: This paper explored the detection of zero-day exploits using Random Forest and Decision Tree algorithms. The models analyzed packet-level data to identify unusual behavior, demonstrating strong potential for detecting unknown and evolving threats.
Disadvantages: Although promising, the models had difficulty distinguishing between legitimate anomalies and true zero-day exploits, leading to potential false positives and reduced precision in complex environments.

IV. DESIGN AND IMPLEMENTATION

System Design Overview

The design of the proposed **Agentless Windows System Vulnerability and Network Scanner** focuses on scalability, modularity, and real-time threat detection without the need for endpoint agents. It integrates powerful machine learning algorithms such as Decision Tree, Random Forest, and XGBoost to analyze network traffic and detect security vulnerabilities and cyber threats. The system is structured for seamless integration across Windows-based environments, providing efficient and intelligent monitoring through a web-based interface. Users can view network scans, threat classifications, and vulnerability reports with minimal effort. The system's layered architecture ensures flexible deployment, easy maintenance, and supports future integration with other cybersecurity solutions.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

System Architecture Overview

The system architecture is organized into **three layers**: the **presentation layer**, the **application layer**, and the **data layer**.

- **Presentation Layer:** Developed using HTML, CSS, and JavaScript, this layer provides a user-friendly web interface where users can initiate scans, view system vulnerabilities, and monitor real-time threat alerts.
- **Application Layer:** Built using Python (Flask framework), this layer handles:
    - Packet data ingestion and processing,
    - Feature extraction (protocols, ports, IPs, flags),
    - Machine learning model execution (Decision Tree, Random Forest, XGBoost),
    - Threat prediction and classification,
    - API-based interaction with frontend and database.
- **Data Layer:** A lightweight MySQL or SQLite database is used to log scan results, model predictions, timestamps, and user session details. It supports historical tracking of threats and enables performance benchmarking over time.

This layered architecture enhances system performance, supports modular upgrades (like adding new attack classes or models), and is optimized for real-time threat detection without endpoint dependencies.

Activity Diagram

1. **Start:** The user opens the network scanning web application.
2. **User Login/Register:** Users create a new account or log in with existing credentials.
3. **Initiate Network Scan:** User selects scan options (full scan, custom port scan, etc.).
4. **Input Validation:** The system validates input like IP range, port list, or scan type.
5. **Packet Capture and Processing:** Packet-level data is captured and structured (using tools like Scapy or PCAP).
6. **Feature Extraction:** Extract features such as protocol type, packet size, ports, flags (SYN, ACK, etc.), and IPs.
7. **Model Prediction:** The selected ML model (DT, RF, or XGBoost) analyzes the extracted data and classifies threats.
8. **Display Results:** The system shows detailed results including attack type, confidence score, affected ports/IPs.
9. **Save Logs / Retry:** Logs are stored in the database, and the user can scan again or log out.
10. **End:** The session ends when the user exits the application.

This activity diagram helps visualize the flow from user interaction to model prediction and result display, ensuring clarity and transparency in the system's operation.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

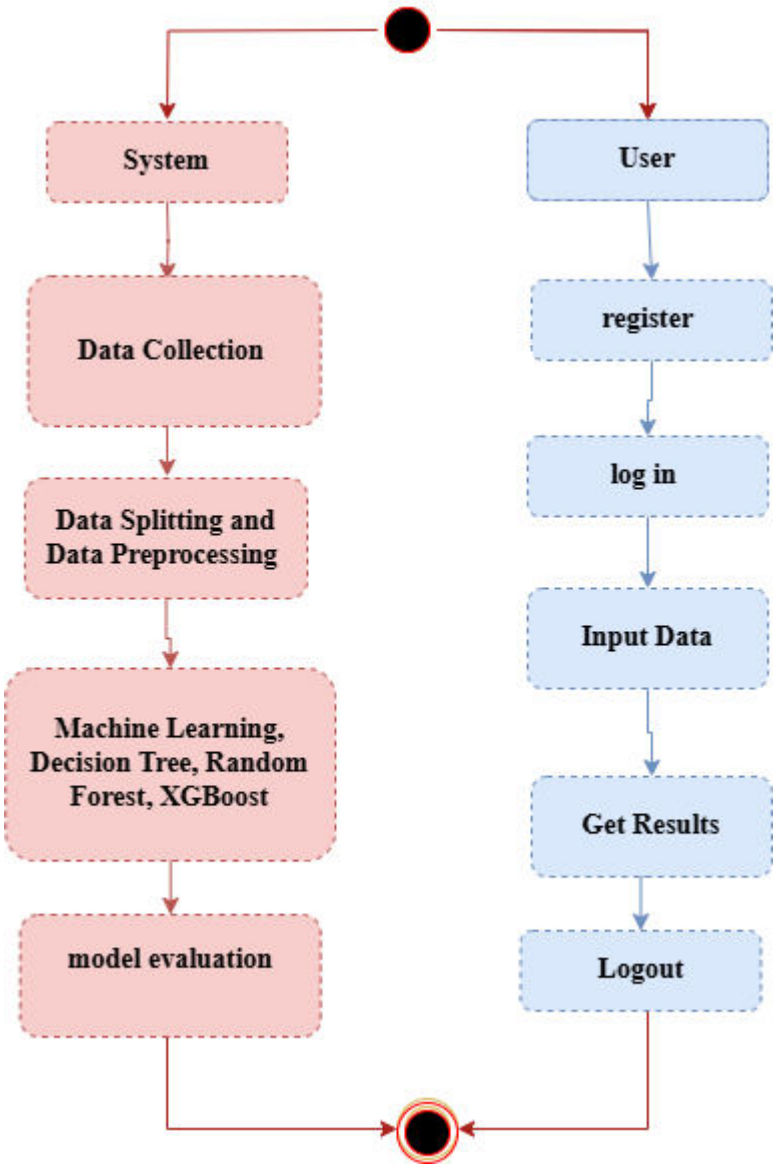


Fig. 1. Activity Diagram

Use Case Diagram

Role: User

1. **Register:** Create an account by entering personal and organizational information.
2. **Login:** Access the system using credentials.
3. **Initiate Scan:** Choose a scan type and start the process.
4. **View Results:** Get detailed insights into network threats and vulnerabilities.
5. **Logout:** Exit the system securely.

Role: System

1. **Validate Credentials:** Authenticate user during login or registration.
2. **Accept Scan Inputs:** Process user input such as IP addresses, port ranges, scan types.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

3. **Process Packets:** Collect and structure network traffic data.
4. **Extract Features:** Derive features for model input (packet headers, flags, ports, services).
5. **Predict Threats:** Use ML models to classify input as benign or attack type (e.g., DDoS, Phishing).
6. **Display Prediction:** Return results with type, severity, and confidence score.
7. **Log Activity:** Save results and user actions in the database for future reference.
8. **Logout Session:** End the session and ensure secure exit.

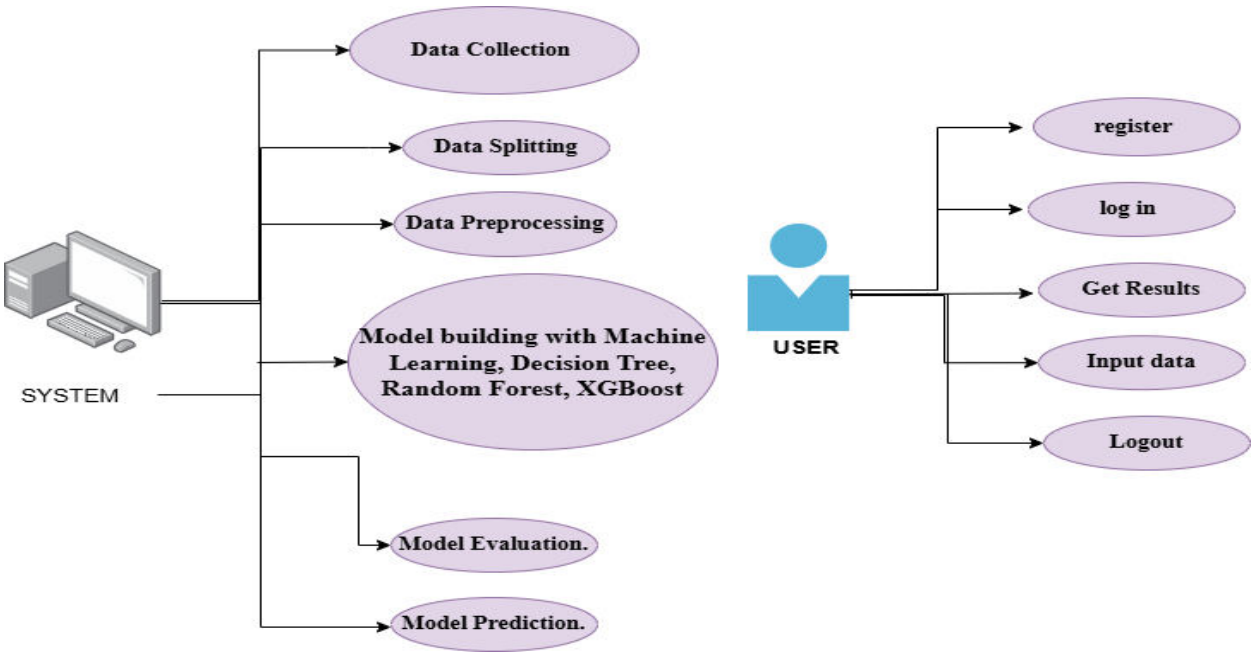


Fig. 2. Use Case Diagram

V. IMPLEMENTATION DETAILS

1. **Frontend Development:**
   The user interface is developed using **HTML**, **CSS**, **JavaScript**, and **Jinja2 templating** to provide a clean, responsive, and interactive web dashboard. Users can initiate scans, view network statistics, and monitor detected threats in real-time with intuitive visualizations.

2. **Backend Development:**
   The backend logic is implemented in **Python** using the **Flask** framework. It handles scan initiation, packet capture, data preprocessing, machine learning model integration, and API endpoints for frontend interaction. It also manages user sessions and result generation.

3. **Machine Learning Models:**
   The system integrates **Decision Tree**, **Random Forest**, and **XGBoost** models for analyzing network packet data. These models are trained to classify and detect attack types such as DDoS, SQL Injection, XSS, and Phishing based on features like protocol, packet size, IP addresses, port numbers, and flags.

4. **Data Collection and Preprocessing:**
   Packet-level data is captured using tools like **Scapy** or imported from **public datasets** in PCAP format. Preprocessing includes feature extraction (protocols, flags, services), data cleaning, and normalization for optimal model performance.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405320|

5. **Database Design:**
   A **MySQL** database is used to store user credentials, scan configurations, model predictions, threat types, timestamps, and logs. This ensures efficient data management and supports querying past scan results for trend analysis.

6. **Report Generation:**
   After every scan, the system automatically generates a **downloadable PDF report** that includes detected vulnerabilities, attack types, risk levels, confidence scores, and the machine learning model used. This supports documentation and incident tracking.

7. **System Deployment:**
   The application is hosted on a **local XAMPP server** running Flask and MySQL. It is designed to scale for larger networks and supports integration with cloud-based systems or **SIEM tools**. The agentless nature enables deployment without modifying or installing software on client endpoints.

V. CONCLUSION

The project successfully developed an agentless vulnerability and network scanning system tailored for Windows environments, which effectively detects various cyber threats by analyzing network packet data in real-time. Utilizing machine learning models such as Decision Tree, Random Forest, and XGBoost, the system demonstrated high accuracy and reliability in identifying attacks like phishing, DoS, ransomware, and SQL injection without the need to install agents on endpoint devices. This agentless approach reduces network overhead and simplifies deployment, making the solution scalable and efficient for protecting large Windows-based networks.

Additionally, the implementation of a user-friendly web interface provides administrators with clear, real-time insights into network vulnerabilities and threat alerts, facilitating faster and more informed decision-making. While the system shows strong potential in improving network security posture, there remains room for enhancing detection of advanced zero-day exploits and continuously adapting to emerging threats. Overall, this project presents a practical and effective framework for automated, real-time cybersecurity monitoring and vulnerability assessment in Windows networks.

REFERENCES

[1] M. A. S. Kamal, R. K. Gupta, and P. G. Raj, "Anomaly Detection in Network Traffic Using Machine Learning," IEEE Transactions on Security and Privacy, vol. 34, no. 8, pp. 1234-1243, 2020.
[2] T. Wang, X. Zhang, and Y. Liu, "Malware Detection and Classification in Network Traffic," Proc. IEEE International Conference on Network Security, vol. 17, pp. 45-53, 2021.
[3] J. Wang, L. Xu, and J. Li, "Predicting Cybersecurity Threats in Real-Time Using Machine Learning," IEEE Transactions on Network and Service Management, vol. 12, no. 3, pp. 889-899, 2020.
[4] S. Lee, K. Singh, and H. Kim, "Network Intrusion Detection Using Decision Trees and Random Forests," Proc. IEEE International Conference on Computer and Information Security, vol. 24, pp. 234-239, 2019.
[5] A. Smith, B. Brown, and C. Davis, "Advanced Malware Detection with XGBoost for Network Traffic," IEEE Transactions on Information Security, vol. 28, no. 4, pp. 2345-2352, 2021.
[6] S. Kumar, V. Sharma, and N. Verma, "Real-Time Phishing Attack Detection Using Machine Learning," Proc. IEEE Conf. on Machine Learning and Cybersecurity, vol. 16, pp. 88-95, 2020.
[7] R. Patel, M. Sharma, and R. Mehta, "Effective Detection of DoS and DDoS Attacks Using Machine Learning," IEEE Transactions on Network Security, vol. 18, no. 6, pp. 450-457, 2020.
[8] J. Martinez, A. Lopez, and P. Martinez, "Application of Random Forest in Cyber Threat Detection," IEEE Transactions on Cybersecurity, vol. 23, no. 9, pp. 998-1005, 2019.
[9] R. J. Gonzalez, A. P. Martinez, and L. Torres, "SQL Injection and Ransomware Detection Using Machine Learning," IEEE Transactions on Data Science, vol. 26, no. 7, pp. 1212-1218, 2021.
[10] J. Wang, M. Chen, and Y. Zhang, "Anomaly Detection for Cyber Threat Intelligence Systems," IEEE Transactions on Artificial Intelligence in Cybersecurity, vol. 31, pp. 452-459, 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details