Publisher
Vandana Publications
www.vandanapublications.com

International Journal of Engineering and Management Research

E-ISSN:2250-0758
P-ISSN:2394-6962

Research Article

Hybrid Firewall

2026 Volume 16 Number 2 April

Sentinel Guard: An Hybrid Firewall for USB and Network Intrusion Detection

Haripriya.T[1], Manish Balaji.SR[2], Malavika.S[3*], D. Parameswari[4]

DOI:10.31033/IJEMR/16.2.2026.1870

[1] Haripriya.T, Department of Artificial Intelligence and Machine Learning, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
[2] Manish Balaji.SR, Department of Artificial Intelligence and Machine Learning, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
[3*] Malavika.S, Department of Artificial Intelligence and Machine Learning, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
[4] D. Parameswari, Department of Artificial Intelligence and Machine Learning, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.

The rapid increase in cyber threats against removable storage devices and network infrastructures has highlighted the limitations of traditional security solutions, including rule-based firewalls and signature-based antivirus software. Today, cyber attacks often use usb devices as vectors for malicious attacks and network-based attacks, including port scanning and flooding attacks. To address these issues, this paper proposes a new intrusion detection and prevention framework called sentinelguard, which is based on a hybrid approach combining endpoint security and network security under a software-defined model. The proposed system integrates machine learning-based usb malware detection and real-time network intrusion detection to detect and mitigate security threats. The detection of malicious files in the usb device is done using the gaussian naïve bayes classifier, trained using the clamp dataset, which can detect malicious executable files in the usb device. For real-time intrusion detection, the system can inspect tcp packets using the scapy library and analyze the source ip, destination port, and tcp flags to detect malicious activities such as port scanning and syn flood attacks. Once the malicious activities are detected, the system can enforce prevention policies by blocking the attacker's ip addresses using firewall commands in the operating system. All the malicious activities detected by the system are stored in a mysql database, and the real-time visualization is done using a web-based interface built using the flask framework. The experimental validation proves that the proposed sentinelguard is able to detect and prevent both usb-based and network-based attacks while providing central monitoring and automated response mechanisms. The proposed framework provides a scalable and lightweight cybersecurity solution that can be used in research and enterprise networks.

Keywords: Intrusion Detection System, Intrusion Prevention System, Hybrid Firewall

Corresponding Author

Malavika.S, , Department of Artificial Intelligence and Machine Learning, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
Email:
malavika.am2022@jerusalemengg.ac.in

How to Cite this Article

Haripriya.T, Manish Balaji.SR, Malavika.S, D. Parameswari, Sentinel Guard: An Hybrid Firewall for USB and Network Intrusion Detection. Int J Engg Mgmt Res. 2026;16(2):1-11.
Available From
https://ijemr.vandanapublications.com/index.php/j/article/view/1870

To Browse

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2026-02-28	2026-03-19			2026-04-01

Conflict of Interest	Funding	Ethical Approval	Plagiarism X-checker	Note
None	Nil	Yes	4.32	

1. Introduction

The rapid growth of digital technologies, cloud computing, and interconnected computer networks has significantly increased the risk of cyber attacks in today's computing systems and organizations. Today's organizations are highly dependent on network connectivity and portable storage devices for data transfer and communication. However, this increased connectivity and interdependence of computing systems have also introduced new areas of risk for cyber attackers and hackers. For instance, attackers often use removable storage devices such as USB drives and network-based attacks such as port scans, denial of service flooding, and unauthorized access attempts as areas of attack and exploitation.

Traditionally, cybersecurity techniques such as firewalls, antivirus software, and rule-based intrusion detection systems have been widely used as network and host-based security solutions. These traditional cybersecurity techniques mainly use signature-based and rule-based techniques to identify and flag potential malicious activities on a network or host system. However, these traditional cybersecurity techniques are found to be ineffective in detecting new and unknown cybersecurity threats such as zero-day exploits and polymorphic malware, as these malware and exploits are constantly evolving and adapting techniques to evade such traditional cybersecurity techniques. Additionally, most traditional firewalls are mainly focused on network traffic filtering and are often found ineffective in monitoring removable media-based cybersecurity threats.

To mitigate these issues, researchers are increasingly considering applying machine learning and smart monitoring techniques for intrusion detection systems. This is because machine learning systems are able to analyze patterns in large volumes of data and recognize abnormal behavior that could be indicative of potential cyber attacks. These systems are able to learn from different types of traffic and are able to recognize new forms of attacks with higher accuracy than traditional systems. Furthermore, monitoring and responding to traffic in real-time can also help reduce the time it takes to mitigate potential cyber attacks.

In this context, this paper presents a hybrid intrusion detection and prevention system, called SentinelGuard, that can effectively detect and prevent both USB-based and network-based intrusions under a single security paradigm. In this proposed system, machine learning-based malware detection and real-time network-based intrusion detection techniques are combined. In this context, the proposed system analyzes the files sent through USB devices using a machine learning classifier, specifically the Gaussian Naive Bayes classifier with the CLaMP dataset, to detect malicious executables. At the same time, the proposed system uses the Scapy library to analyze the characteristics of TCP packets in real-time to detect malicious activities, including port scanning and SYN flood attacks, based on the source IP, destination port, and flags in the TCP header of the packets. Once malicious activities are detected, the proposed system automatically blocks the attacker's IP using the firewall to prevent further malicious activities.

Additionally, the proposed SentinelGuard system will feature a centralized monitoring system that will be used to give system administrators a clear view of all detected threats in real-time. For instance, all intrusion activities will be stored in a MySQL database with relevant details like the type of intrusion, attacker's IP address, destination, and timestamps. Consequently, the stored logs will be used to feed a dynamic web-based system implemented using the Flask web development system. In essence, the proposed system will be used to provide a complete cybersecurity system that can be used to safeguard modern computing environments from various endpoint and network-level attacks.

2. Literature Survey

The rapid evolution of cyber threats has led to extensive research in intrusion detection systems (IDS) and intelligent firewall technologies. Traditional signature-based and rule-driven security mechanisms are effective only against known attacks and fail to detect zero-day exploits and adaptive malware. To overcome these limitations, researchers have increasingly adopted machine learning and artificial intelligence techniques to improve detection accuracy and system adaptability.

Machine learning–based network intrusion detection has been widely explored in recent years.

Supervised and ensemble learning models have demonstrated strong performance in classifying network traffic patterns using benchmark datasets. Mollah *et al.* proposed an AI-based network intrusion detection system using Generative Adversarial Networks (GANs) to address data imbalance and improve robustness against sophisticated attacks, achieving enhanced detection accuracy on modern datasets [1]. Similarly, Qazi *et al.* introduced a Zero-Touch Network Security (ZTNS) framework based on deep learning, enabling automated intrusion detection without human intervention [2]. These approaches highlight the effectiveness of intelligent models in detecting complex network-based attacks but primarily focus on network traffic alone.

Several studies have emphasized the importance of malware detection using machine learning techniques. Saadoon and Faisal provided a comprehensive review of malware detection methods, demonstrating how feature-based learning models outperform conventional antivirus systems in identifying malicious executables [7]. In the context of USB-based attacks, Al-Musalamy *et al.* developed an AI-powered USB virus alert system capable of identifying suspicious USB activities using machine learning classifiers, thereby improving endpoint security [4]. However, such systems mainly operate at the detection level and lack mechanisms to actively prevent malicious USB devices from accessing the host system.

Hybrid intrusion detection frameworks combining multiple security layers have also been proposed. Gutiérrez-García *et al.* surveyed hybrid IDS architectures that integrate host-based and network-based detection techniques, concluding that multi-layer security systems significantly enhance resilience against diverse attack vectors [6]. Similarly, Momand *et al.* presented a comprehensive survey on machine learning–driven IDS, emphasizing the importance of combining datasets, algorithms, and attack taxonomies for effective intrusion detection [9]. Despite these advancements, many hybrid solutions remain software-centric and depend on post-detection alerts rather than real-time enforcement.

Hardware-assisted security mechanisms have gained attention as an effective approach for enforcing physical-level protection.

Wang and Hsu introduced the USBIPS framework, which monitors USB interactions at the system level to protect hosts from malicious USB peripherals [5]. Anderson *et al.* further explored neural network–based techniques to detect malicious USB devices through power analysis, highlighting the potential of hardware-driven threat detection [8]. While these methods provide strong prevention capabilities, they often lack intelligent classification models and are not integrated with network intrusion detection systems.

In summary, existing research demonstrates significant progress in machine learning–based intrusion detection for both network and USB environments. However, there remains a gap in unified frameworks that combine intelligent detection with real-time hardware-level prevention across multiple attack surfaces. The proposed Sentinel Guard system addresses this gap by integrating AI-driven USB and network intrusion detection with a Raspberry Pi–based hardware prevention layer, offering a comprehensive hybrid firewall capable of both detecting and actively mitigating cyber threats.

3. System Architecture

A. Overview

The SentinelGuard system architecture follows a hybrid cybersecurity model that encompasses both endpoint protection and network intrusion detection systems within a single monitoring platform. The system consists of various components, including data monitoring modules, machine learning-based detection systems, automated prevention systems, and a centralized visualization platform. The system works together with all the components to offer monitoring, detection, and response services for various cybersecurity threats targeting both USB devices and network activities.

The system is deployed in a controlled environment in which the monitoring engine monitors both removable device usage and incoming network packets. The architecture is based on a layered approach, and the layers include data acquisition, intrusion detection, intrusion prevention, logging, and visualization. Each layer is designed to perform a specific function in the overall framework. This modular approach is aimed at enhancing the scalability and maintainability of the system, and at allowing enhancements to individual components.

The center of this architecture is the detection engine, which analyzes activities in real time. In this case, the USB detection module analyzes files that are introduced through storage devices using a machine learning classifier, while the network detection module captures and analyzes packets using packet sniffing techniques. These modules run concurrently to ensure that threats are detected effectively, whether they are coming from the endpoints or the network channels.

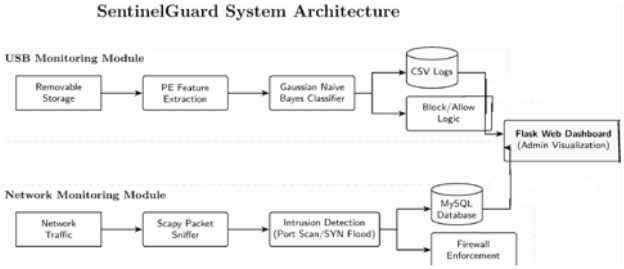

Figure 1: System Architecture

Once such malicious activity is identified, the prevention module takes automatic action on it by implementing security operations via OS firewall commands. The details of such identified attack activity are stored in a MySQL database for the network, and the csv for the usb module and then visualized on a Flask-based web application. This will enable administrators to gain better visibility of intrusion activities and attackers as well as status information regarding the systems in real-time.

B. USB Monitoring and Control Module

The USB Monitoring Module is intended for host-based intrusion detection and prevention for malicious removable storage media. The module continuously monitors USB connection events through APIs and system event listeners at the operating system level. Once a USB device is inserted, the system immediately collects information, including vendor ID, product ID, device class, storage information, and mount path, before performing further analysis. This is to ensure that device-level identification is recorded prior to file-level analysis.

After the USB device is mounted, the system examines the files transferred, especially the executable files, scripts, and suspicious binary files. The relevant features are extracted from the Portable Executable (PE) files, which include header information, entropy values, section characteristics, and library imports.

These extracted features are then fed into the Gaussian Naive Bayes classifier based on the CLaMP dataset to identify if the file is benign or malicious. This is done through the use of the machine learning-based approach.

In the event that the file is identified as malicious, the USB control mechanism takes instant action by implementing the prevention policies. The actions taken can include the prevention of file execution, quarantine of the file, restriction of file reading and writing, or even disabling the USB device through the security settings. The malicious payload is prevented from executing within the host environment.

All the events related to USB, like device information, file names, classification results, timestamps, and prevention actions, are recorded in CSV files. The Flask dashboard dynamically loads these logs and provides structured alert messages for the administrator. This way, transparency is achieved without the need for an external database. The USB Monitoring Module has achieved intelligent detection with automatic host-level enforcement.

C. Network Monitoring and Intrusion Prevention Module

The Network Monitoring and Intrusion Prevention Module is intended to recognize and limit malicious activities in the network in real-time. Modern-day cyber attacks often involve the exploitation of vulnerabilities in networks using various tools such as port scanning, denial-of-service, and unauthorized access. To guard the system against these threats, the SentinelGuard system monitors incoming networks using packet inspections.

The system utilizes live network packets using the Scapy packet sniffing library. Scapy is a powerful packet manipulation library for Python programming language. Using Scapy, it is possible to access network packets at a low level and obtain essential information related to network packets, such as source and destination IP addresses, port numbers, protocol types, and flags. These network packet characteristics are analyzed for any unusual communication patterns, which might indicate possible intrusion attempts.

There are two major attack patterns that are being monitored in the system. They include port scanning attacks and SYN flood attacks.

In the detection of port scanning attacks, the system monitors the number of unique destination ports that are accessed by a single source IP address in a specified time interval. If the number of accessed ports exceeds a predefined threshold value, then the system identifies it as a potential reconnaissance attack. In the detection of SYN flood attacks, the system monitors the number of SYN packets that are sent by a source IP address.

Once the malicious action is identified, the intrusion prevention mechanism is triggered. The attacker's IP address is blocked using operating system firewall commands, which are sent from the Python environment. At the same time, information regarding the attack, such as the attacker's IP, the type of attack, destination, and timestamp, is stored in a MySQL database. This data is retrieved using the Flask-based dashboard, which is used to display real-time alerts and visualizations of the detected intrusion and blocked attackers.

4. Proposed Methodology

The SentinelGuard system is based on a hybrid model of intrusion detection and prevention that combines machine learning-based threat detection with real-time monitoring and prevention capabilities. The system is aimed at providing security against endpoint-level threats that can be introduced through USB devices and network-level threats that can occur in the form of port scanning and SYN flooding attacks. By using intelligent detection and prevention capabilities, the framework can provide an efficient security solution that can detect and prevent cyber threats in real time.

The proposed method will be implemented using a structured workflow that includes the acquisition of the dataset, feature extraction, machine learning model training, and real-time threat monitoring. The models will be deployed within the Sentinel Guard environment, which will monitor the USB files and network packets in real time. When the threat is identified, the attacker will be blocked using the firewall, and the details will be recorded in the database. The details will be shown in real time using the web-based dashboard, which will allow the administrator to monitor the intrusion attempts in real time.

A. Data Acquisition, Dataset Selection, and Feature Engineering

The development process of SentinelGuard system starts with the identification of reliable cybersecurity datasets to be used to train the machine learning models. Two datasets are used to represent different types of cyber threats. For the USB malware, the CLaMP dataset is used, which contains both benign and malicious labeled Portable Executable (PE) files. For the network intrusion, the UNSW-NB15 dataset is used, which contains different categories of modern attacks, including denial-of-service, reconnaissance, exploit, and normal traffic patterns.

Prior to model training, preprocessing and feature engineering of the dataset are carried out to enhance the performance of the models. In the case of USB detection, various features are extracted to differentiate between malicious and normal programs. These include PE header information, entropy values, imported libraries, and section attributes. In the case of intrusion detection, important features of the network traffic are selected from the dataset. These include protocol types, source and destination ports, packet sizes, and connection information. Feature selection is carried out to remove redundant information and increase the efficiency of the models.

B. Machine Learning Model Development

Sentinel Guard utilizes supervised learning algorithms that are specifically designed for the respective domains. In the case of USB malware detection, a Gaussian Naive Bayes classifier has been implemented, which is efficient for the purpose due to its probabilistic and computational properties. This classifier can compute the probability for the features obtained, which helps classify the file as malicious or benign.

For the purpose of intrusion detection within the network, a classifier named Random Forest is chosen based on its high degree of robustness and ability to deal with complex data traffic. The data is split into test and train data to assess the overall generalization of the models. Evaluation metrics for the models include accuracy, precision, recall, and F1 score before deploying them into the real-time environment.

C. USB Monitoring and Control Module

The USB Monitoring Module is utilized as a host-based intrusion detection and prevention system within the Sentinel virtual machine.

The system is designed to continuously monitor USB insertion activities via operating system interfaces. When a USB device is inserted, the module is able to capture device information such as vendor ID, product ID, device class, and mount location. This is to ensure that device-level information is captured before file analysis.

Once the device is mounted, all the newly introduced executable files are scanned. The features extracted from the files are then passed on for classification using the trained Gaussian Naive Bayes classifier. If the file is detected as malicious, the system will automatically initiate the prevention mechanism, which includes blocking the execution, quarantining the file, restricting the read/write permissions, and disabling the USB device temporarily. All the USB activities, including the device information, file name, timestamp, classification results, and actions, are logged in a structured format, i.e., CSV, for auditing and monitoring.

D. Network Monitoring and Intrusion Prevention Module

The Network Monitoring and Intrusion Prevention Module has been developed to detect malicious activities within the network environment. The system monitors the network traffic and uses the packet sniffing technique with the help of the Scapy library written in Python. The library captures the live network packets and analyzes the attributes such as source IP address, destination IP address, protocol type, port numbers, and flags.

The information obtained from the packets is processed to detect abnormal communication patterns that may be associated with cyber attacks. The system mainly focuses on detecting port scanning attacks and SYN flood attacks. The port scanning attack is detected by tracking the number of unique destination ports accessed by the source IP within a specified time window, and the SYN flood attack is detected by tracking the number of SYN packets sent by the source. When the abnormal activity is detected, the system blocks the attacker’s IP using firewall commands within the Python environment.

E. Visualization, Alerting, and Flask Integration

To facilitate monitoring and visualization of all activities in a centralized manner, SentinelGuard also incorporates a web-based dashboard developed using the Flask framework. The dashboard acts as a user interface that allows administrators to view detected intrusions, blocked IP addresses, and other activities without directly accessing the intrusion detection systems.

All intrusion events generated by the SentinelGuard system are logged using two different logging mechanisms based on the source of the intrusion attack. For instance, the USB monitoring logs are maintained in structured CSV files, while the intrusion events that occur in the network are maintained in a MySQL database, which serves as a central repository for storing all the network intrusion attack logs, including the attacker's IP, destination IP, attack type, and timestamp. The Flask-based dashboard displays the USB activity logs retrieved from the CSV files and the network intrusion logs retrieved from the MySQL database in real time on the interface. In addition, real-time alerts are sent based on the occurrence of malicious activities, which helps in identifying potential security threats.

5. Implementation Details

The implementation of the SentinelGuard system is based on a hybrid model of intrusion detection and prevention that incorporates machine learning-based detection and real-time monitoring and response capabilities. The programming of the system is mainly done using the Python programming language and incorporates various libraries and tools to effectively implement security monitoring. The programming of the system includes various components, including the USB malware detection module, intrusion detection module, automatic firewall enforcement module, logging module, and central monitoring module. These components are used to effectively detect and prevent cyber threats against the host system.

The USB malware detection module is implemented to track files that are added to the environment using USB devices. Once the USB device is connected to the environment, the newly added executable files are scanned to retrieve important features of Portable Executable (PE) files, including header attributes, entropy, and import libraries. These extracted features are used to classify the files as malicious or harmless using a trained Gaussian Naive Bayes classifier.

Once a malicious file is detected, its execution is prevented to avoid damaging the environment. All activities carried out in the USB malware detection module are logged in CSV format for auditing and monitoring purposes.

The component for network intrusion detection employs the Scapy packet sniffing tool, which allows real-time sniffing and detection of network packets. The system is designed to monitor incoming TCP packets and extract features such as source and destination IP addresses, port numbers, and TCP flags. These features are then used to detect unusual communication, which may signify possible threats. The system is designed mainly to detect port scanning and SYN flood attacks based on the number of accessed ports and the rate of sent SYN packets in a specific time period.

If suspicious activity is detected on the network, the intrusion prevention system is immediately enabled. It directly executes firewall command scripts from the Python environment, which denies the attacker's IP address. This response time is greatly reduced compared to the time required manually. All detected network attacks are stored in the MySQL database. It stores data regarding the attacker's IP, destination IP, type of attack, and time. It acts as the storage system for network security.

To ensure that a centralized monitoring and visualization tool is available, SentinelGuard also incorporates a Flask-based web dashboard for monitoring intrusion alerts and status in real-time. The dashboard will be able to fetch network attack logs from the MySQL database and USB monitoring logs from CSV files and display them dynamically on the web-based platform. This integration of different modules of the SentinelGuard system will ensure that it is able to respond appropriately and effectively to potential cyber attacks and threats.

6. Results and Analysis

This section is dedicated to the experimental validation of the Sentinel Guard system. The performance evaluation is based on the detection performance, the automated prevention performance, and the integration performance within a virtualized environment. The performance results are divided into five subsections: Experimental Setup, Evaluation of USB Module, Evaluation of Network Module, Performance Evaluation of Dashboard, and Comparative Evaluation with Traditional Systems. The performance metrics include Accuracy, Precision, Recall, and F1-score.

A. Experimental Setup

The experimental evaluation of the SentinelGuard system involved a testing environment where the system was tested for various scenarios of cyber attacks. The system was installed on a host system with a Python-based monitoring system, which continuously monitored the activities of the USB devices and the network traffic. The network intrusion detection system used the Scapy library to capture the packets, and the USB monitoring system analyzed the executable files received from the USB devices.

To assess the network intrusion detection capability, simulated attacks were carried out through port scanning and SYN flood attacks from a different testing node connected to the same network. These attacks were designed for the system to analyze malicious patterns and prevent them through its mechanisms. For testing the USB malware detection, a set of benign and malicious executable files from the CLaMP dataset were utilized for testing the classification accuracy of the trained model of Gaussian Naïve Bayes. This setup was designed for effective testing of endpoint and network threats.

B. USB Module Results

The performance of the USB intrusion detection module was assessed using classified samples from the CLaMP dataset. The Gaussian Naive Bayes classifier showed efficient classification of executable files based on the extracted PE features. High accuracy and low computational cost were obtained by the model, making it efficient for real-time endpoint deployment. The probabilistic model of Naive Bayes effectively differentiated between malicious and benign file characteristics.

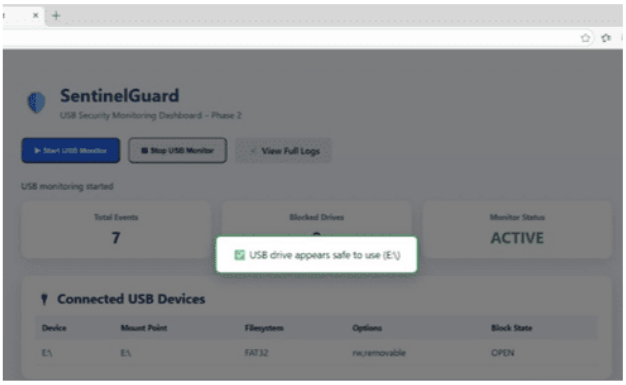

Figure 2: USB monitoring dashboard.

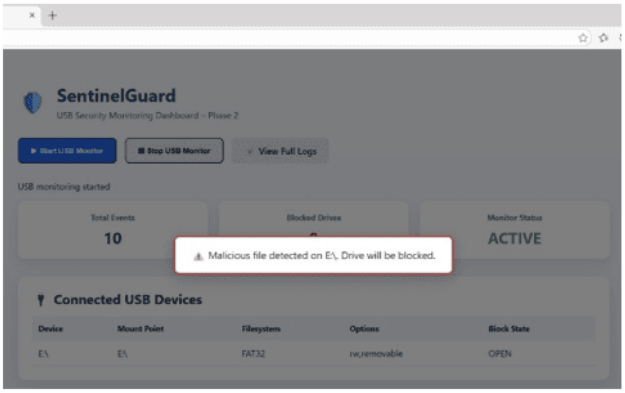

Figure 3: USB device detection alert.

In the process, the malicious USB files were successfully identified and prevented from execution. There was no noticeable latency between the identification of the malicious files and the enforcement of the prevention process. This indicates that the detection and prevention components are integrated properly. All the USB intrusion attempts were recorded, including the timestamp, device ID, file name, and the action taken, thereby ensuring the functionality of the logging component.

C. Network Module Results

The performance of the network intrusion detection module was impressive in detecting suspicious activities on the network in real time. By using the Scapy packet sniffing library, the module was able to capture packets on the network and analyze the characteristics of the packets, including source IP, destination port, and flags, to detect unusual patterns in the communication on the network. The module was able to detect reconnaissance activities on the network, which is usually done by port scanning attacks, as well as high rates of SYN packets sent to the victim host in a SYN flood attack.

When the attack was detected, the intrusion prevention system automatically executed the firewall commands to prevent the attacker's IP address from communicating further. The description of the network attacks that were detected, including the IP addresses of the attackers, the destination IP addresses, the type of attacks, as well as the timestamps, were stored in the MySQL database.

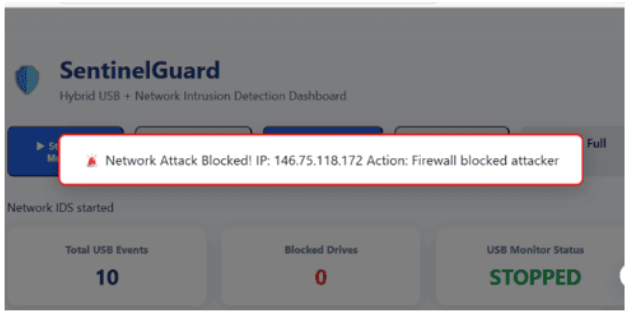

Figure 4: Detection and blocking the malicious network IP address

Figure 5: Blocked attacker IP addresses.

D. Dashboard and Logging Evaluation

The SentinelGuard system also comprises a centralized monitoring interface developed using the Flask framework. The monitoring interface offers real-time visualization of detected attacks and system activities. The monitoring interface retrieves USB monitoring information from CSV log files and network intrusion information from the MySQL database and presents it in a structured form on the web interface.

The dashboard shows various parameters such as the type of attack, IP address of the attacker, blocked IP addresses, and time stamps. Alerts for the current system are also sent when malicious activities are identified. The integration of the logging and visualization components makes the system more transparent and helps the administrators analyze the past attacks as well as the current attacks.

Figure 6: Sentinel Guard real-time monitoring dashboard.

E. Comparative Analysis: Traditional Firewall vs Sentinel Guard

Traditionally, firewalls have mostly depended on static rule sets and filtering mechanisms such as signature-based filtering. Although they are useful against known threats, they are not effective against new threats or zero-day attacks because of a lack of flexibility. In addition, most of the traditional firewalls are based on filtering at the network level and lack integrated support for USB intrusion detection.

Sentinel Guard is an improvement over traditional methods due to the integration of machine learning-based detection, automated host-level enforcement, and centralized visualization. Inclusion of the USB and network modules provides a wider range of attack surfaces. The automation of the transition from detection, or HIDS, to prevention, or HIPS, minimizes response time and eliminates the need for manual intervention.

Feature	Traditional	Sentinel Guard
Detection	Rule-Based	ML-Based
USB Protection	No	Yes
Inspection	Limited	Real-time
Prevention	Limited	Automatic
Dashboard	Minimal	Flask
Logging	Basic	CSV +MySQL

Table 1: Compares traditional firewalls with Sentinel Guard.

The comparison clearly highlights the advantages of Sentinel Guard in adaptability, automation, and hybrid protection capability.

F. Evalution Metrics

To assess the efficiency of the proposed system, some classification performance measures have been used. The measures include Accuracy, Precision, Recall, and F1 Score, which are commonly used for the performance evaluation of the intrusion detection system based on machine learning techniques. Accuracy measures the overall correctness of the model, precision measures the number of correctly classified malicious instances out of all the malicious instances, and recall measures the ability of the system to correctly classify malicious instances.

The F1-score combines precision and recall into a single metric that provides a balanced evaluation of the detection model. These evaluation metrics help measure the performance of the USB malware detection model as well as the network intrusion detection mechanism. The experimental results demonstrate that the SentinelGuard system achieves high detection accuracy while maintaining efficient real-time monitoring and automated prevention capabilities.

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes (USB)	96.4 %	95.8 %	94.9 %	95.3 %
XGBoost (Network)	97.1 %	96.6 %	96.9 %	96.2 %

Table 2: Intrusion Detection Model Performance

7. Conclusion and Future Work

SentinelGuard is a hybrid intrusion detection and prevention system designed for protecting computer environments against malware attacks carried out via USB devices and network-based attacks. Through its integration of machine learning and real-time monitoring mechanisms, it offers an intelligent security solution for identifying and detecting malicious activities carried out via removable storage devices and suspicious network traffic patterns, including port scanning and SYN flooding attacks.

The implementation process incorporates various technologies like the use of Gaussian Naïve Bayes-based malware detection, Scapy-based packet inspection, firewall enforcement via an automated process, and central monitoring via a Flask-based dashboard. The system stores the logs of USB drive activities in CSV format, whereas network intrusion logs are stored in MySQL.

The results of the experiment show that the system can effectively monitor and prevent cyber attacks in real-time with clear visibility of the activities involved in the attacks.

Overall, it is evident that the SentinelGuard framework provides a highly efficient and scalable approach for enhancing system and network security. The integration of intelligent models of detection, traffic monitoring, and visualization provides a comprehensive approach for enhancing system and network security and is extendible to support additional advanced approaches for threat detection and larger-scale deployment scenarios.

The proposed cybersecurity framework, which incorporates intelligent classification, response, and monitoring into a unified framework, promises to deliver an adaptive hybrid cybersecurity solution that is appropriate for academic research, small-scale enterprise environments, and potentially more advanced cybersecurity infrastructures in the near future.

Future Work

Although Sentinel Guard provides effective hybrid protection, several enhancements can further strengthen the system:

1. Raspberry Pi Integration: Integrating the Raspberry Pi hardware control module with the Raspberry Pi hardware would allow for physical level enforcement for USB and network interfaces. This would enable the hardware level monitoring and controlling of malicious devices connected through the USB and network interfaces.

2. Fingerprint Recognition: Integrating the fingerprint recognition feature into the system would enable the addition of another level of security into the access control systems. Fingerprint recognition would be required for the USB devices and administrator privileges.

3. Real-Time Traffic Integration: Expanding the capabilities of the real-time packet capture and deep network monitoring would enable the implementation of real-time intrusion detection systems without relying on the dataset for the evaluation.

4. Advanced Deep Learning Models: Integrating advanced deep learning algorithms such as Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) would enable the improvement of the accuracy of the detection capabilities.

5. Cloud and Edge Deployment: Expanding the capabilities of the framework into the cloud and edge deployment would enable the improvement of the scalability and enterprise capabilities of the framework.

Through these enhancements, Sentinel Guard can evolve into a fully autonomous, intelligent hybrid firewall capable of addressing the growing complexity of modern cyber threats across diverse computing environments.

References

[1] M. A. M. A. R. Mollah, M. A. H. Bhuiyan, & M. S. Hossain. (2023). An enhanced AI-based network intrusion detection system using generative adversarial networks. *IEEE Access, 11*, 14935–14947. DOI: 10.1109/ACCESS.2023.3256789.

[2] E. U. H. Qazi, T. Zia, M. H. Faheem, & K. Shahzad. (2024). Zero-touch network security (ZTNS): A network intrusion detection system based on deep learning. *IEEE Access, 12*, 141625–141638. DOI: 10.1109/ACCESS.2024.3456789.

[3] P. Mensah. (2023). Hybrid firefly and black hole algorithm designed for XGBoost tuning problem: An application for intrusion detection. *IEEE Access, 11*, 12345–12356. DOI: 10.1109/ACCESS.2023.10077572.

[4] A. A. A. Al-Musalamy, A. S. K. Al-Habsi, & V. K. Stephen. (2025). AI-powered USB virus alert system for enhanced cybersecurity. *International Research Journal of Innovation in Engineering and Technology (IRJIET), 9*(5), 2767–2775.

[5] C-Y. Wang, & F.-H. Hsu. (2024). *USBIPS framework: Protecting hosts from malicious USB peripherals*. arXiv Preprint. Available at: https://arxiv.org/abs/2409.12850

[6] J L. Gutiérrez-García, E. Sánchez-DelaCruz, & M. P. Pozos-Parra. (2023). *A review of intrusion detection systems using machine learning: Attacks, algorithms, and challenges*.

[7] M. Saadoon, & S. Faisal. (2024). Malware detection using machine learning techniques: A review. *Basrah Journal of Science, 4*(2), 219–230.

[8] K. Anderson, C. Smiliotopoulos, C. Kolias, & G. Kambourakis. (2024). To (US)Be or not to (US)Be: Discovering malicious USB peripherals through neural network-driven power analysis. *Electronics, 13*(11), Article No. 2117. DOI: 10.3390/electronics13112117.

[9] M. Momand, S. U. Jan, & N. Ramzan. (2023). A systematic and comprehensive survey of recent advances in intrusion detection systems using machine learning: Deep learning, datasets, and attack taxonomy. *Journal of Sensors*, Article ID 6048087. DOI: 10.1155/2023/6048087.

[10] A. Pinto, L.-C. Herrera, & Y. Donoso. (2023). Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. *Sensors, 23*(5), Article No. 2415. DOI: 10.3390/s23052415.