International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

App Noc: App-aware firewall with AI log analytics

Omkar Gade *, Akshat Bhatt, Shreeja Gundlur, Vijaya S.Patil and Abhishek Wagavekar

Computer Science and Engineering MIT Art, Design and Technology University Pune, India.

International Journal of Science and Research Archive, 2026, 19(02), 251-257

Publication history: Received on 27 March 2026; revised on 02 May 2026; accepted on 05 May 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.2.0996

Abstract

This study introduces App Noc, an advanced firewall solution that is aware of application-level activity and enhanced with an AI-powered log analysis system for proactive security. Traditional firewall systems generally function at the network or system level and do not provide insights into how individual applications behave in terms of network usage. App Noc overcomes this challenge by detecting which applications are responsible for network communication and enforcing tailored security rules for each application. Furthermore, it gathers live network logs and applies machine learning techniques to identify unusual or potentially harmful patterns. The architecture includes a lightweight client-side agent responsible for monitoring and enforcing policies, along with a centralized dashboard server for managing rules, visualizing logs, and performing anomaly detection using AI. The objective of App Noc is to streamline network security operations, enhance visibility into application-driven traffic, and facilitate early identification of suspicious behavior in both enterprise and academic settings.

Keywords: Application-Aware Firewall; Log Analytics; Machine Learning; Anomaly Detection; Network Security; App Noc

1. Introduction

In the modern era of interconnected systems, organizations depend extensively on network infrastructures and distributed applications for communication, data handling, and service delivery. This increased dependency, however, expands the attack surface, making it difficult for conventional security systems to provide adequate protection. Traditional firewalls mainly operate at the network and transport layers, relying on parameters such as IP addresses, ports, and protocols, without recognizing which application process initiates the communication. Consequently, attackers can misuse legitimate applications or inject malicious data through trusted channels, effectively bypassing static firewall configurations and signature-based defenses. To overcome these limitations, this paper presents App Noc—an application-aware firewall integrated with AI-driven log analytics, designed as a modern security solution combining process-level monitoring with intelligent anomaly detection. Unlike traditional approaches, App Noc continuously tracks which applications are generating network traffic and applies dynamic, application-specific security policies. Every network request, whether initiated by a browser, background service, or unknown process, is recorded and analyzed centrally. The intelligence of App Noc lies in its AI-based log analytics module, which evaluates both historical and real-time data to understand normal system behavior. By utilizing unsupervised machine learning methods such as Isolation Forest and One-Class SVM, the system identifies deviations that may indicate malware activity, data leakage, or insider threats. These anomalies are correlated with firewall actions to generate alerts and enable automated responses. The system is developed using open-source tools including Flask, Python, SQLite, and pistil, making it flexible and accessible for research, academic, and enterprise use. By merging application awareness, centralized control, and AI-based insights, App Noc bridges the gap between traditional firewalls and intelligent security frameworks, empowering administrators with better visibility and proactive threat management capabilities.

* Corresponding author: Omkar Gade

2. Literature Review

2.1. Traditional-Firewalls

Conventional firewalls, such as iptables or Windows Firewall, operate at the network and transport layers of the OSI model. They control traffic based on IPs, ports, and protocols but lack context about which application initiated the traffic.

2.2. Application-Aware-Firewalls

Next-generation firewalls (NGFWs) add visibility into specific applications, yet they are often commercial and resource-heavy. Research systems like AppID and Palo Alto's Application Framework provide advanced control but are difficult to deploy in small-scale or academic environments.

2.3. C.AI-and-Log-Analysis

Machine Learning algorithms have been widely used in cybersecurity for anomaly detection. Techniques like Isolation Forest, One-Class SVM, and Decision Trees can identify outliers in network traffic, helping to flag zero-day or insider threats that traditional signature-based systems miss.

2.3.1. Combined- AI+ Firewall -Systems

Recent studies show that integrating log analytics with rule-based firewalls enhances early threat detection and reduces false positives. However, few open-source implementations combine both in a lightweight, decentralized manner suitable for small labs or institutions — motivating the development of AppNoc.

3. Methodology

The App Noc system is designed as a modular and real-time framework capable of monitoring, controlling, and analyzing network traffic at the application level. It integrates rule-based firewall mechanisms with AI-driven analytics for effective threat detection.

3.1. Firewall Agent (Client Side)

Installed on endpoint devices (Linux or Windows).

Monitors active network connections and maps them to running applications using system APIs (e.g., psutil, netstat, or Windows Filtering Platform).

- Enforces security rules — e.g., blocking Notepad from accessing the internet while allowing browsers.
- Sends network logs to the central server periodically.

3.2. Central Server (Dashboard)

- Built using Flask (Python) with a SQLite or MySQL database.
- Receives and stores logs from all agents.
- Allows administrators to define and push application-specific firewall policies.
- Visualizes logs and connection histories.

3.3. AI Log Analytics Engine

- Processes incoming logs for feature extraction (bytes sent, ports, frequency, destinations).
- Trains an Isolation Forest model to identify unusual application behavior or traffic anomalies.
- Flags suspicious patterns on the dashboard and recommends blocking actions.

4. Architecture And Implementation

App Noc has a modular architecture designed for real-time application-level traffic monitoring, anomaly detection, and intelligent firewall enforcement in enterprise or educational lab environments. It integrates client-side agents, a central dashboard, and an AI-based log analytics engine.

4.1. Primary Components

4.1.1. Application Traffic Monitor

Tool/Library: socket (Python)

Function

- Continuously observes all outbound and inbound network activities on endpoint devices.
- Identifies and tracks applications that initiate network communication in real time.
- Maps low-level socket connections to high-level application processes for better visibility.
- Provides raw data to the firewall agent for further processing and policy enforcement.
- Ensures accurate detection even in multi-process or multi-threaded application environments.

4.2. Firewall Agent (Client-Side)

Tool/Library: iptables (Linux), Windows Filtering Platform (Windows)

4.2.1. Function

- Implements fine-grained, application-specific firewall rules such as allow, block, or restrict access.
- Dynamically updates rules based on instructions received from the central dashboard.
- Prevents unauthorized applications from accessing network resources or external servers.
- Captures detailed logs of every network connection, including metadata like timestamps, protocols, and data size.
- Sends these logs securely to the central server in JSON format for further analysis.
- Maintains a balance between security enforcement and system performance through optimized rule handling.

4.3. Central Dashboard Server

Tool/Framework: Flask (Python), SQLite/MySQL database

4.3.1. Function

- Acts as the centralized repository for logs collected from multiple client agents.
- Provides administrators with the ability to configure, update, and distribute firewall rules.
- Displays real-time and historical network activity through interactive dashboards and charts.
- Generates alerts based on AI-detected anomalies and predefined thresholds.
- Supports role-based access control for secure administration.
- Facilitates incident investigation through advanced filtering and querying of logs.

4.4. AI Log Analytics Engine

Algorithm: Isolation Forest (unsupervised anomaly detection)

4.4.1. Function

- Processes large volumes of log data to extract relevant behavioral features.
- Builds a baseline model representing normal application activity patterns.
- Detects deviations such as abnormal traffic spikes, unusual destinations, or irregular communication intervals.
- Assigns anomaly scores to each event to indicate the severity of potential threats.
- Integrates with the dashboard to visualize anomalies and suggest mitigation strategies.
- Continuously improves detection accuracy by learning from new incoming data.

4.5. Log Storage and Reporting:

File/Database: JSON logs in central database

4.5.1. Function

- Stores structured logs for long-term retention and historical analysis.
- Supports reporting features for auditing, compliance, and performance evaluation.
- Provides datasets for training and refining machine learning models.

- Enables traceability of network events for forensic investigations.
- Ensures data integrity and consistency across distributed components.

5. Results

This is the main dashboard of our firewall UI.

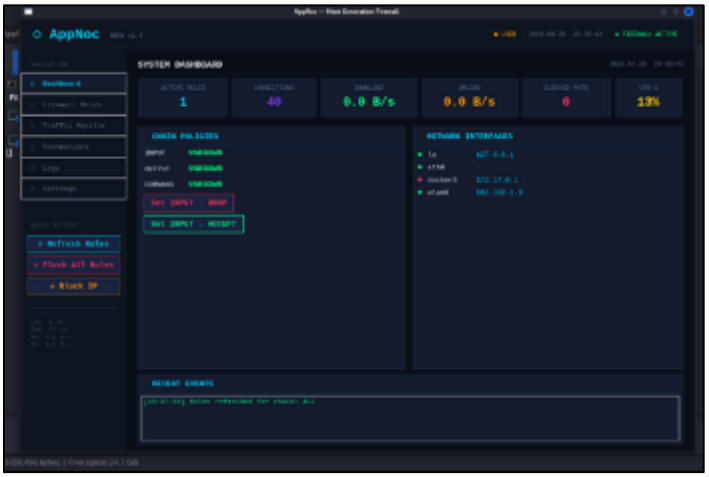

Figure 1 System Dashboard (Initial State)

- Active Rules: 1 → Only one firewall rule is currently applied
- Connections: 40 → Total active network connections
- Download/Upload: 0.0 B/s → No current traffic (idle state)
- Blocked Packets: 0 → No packets blocked yet
- CPU: 13% → System usage is low
- Set INPUT → DROP → blocks incoming traffic
- Set INPUT → ACCEPT → allows incoming traffic

Figure 2 System Dashboard (Active Traffic)

- Network activity has started (likely browsing or background apps)
- Firewall is now monitoring real-time traffic
- Still 0 blocked packets → nothing suspicious detected

Figure 3 Traffic Monitor

This module functions as a real-time network traffic analyser, similar to tools like Wireshark or Netstat, providing

- Identification of the active network interface (wlan0 in this case)
- Visibility into data flow rates and packet transmission
- Monitoring of real-time traffic patterns and trends
- Ability to detect sudden spikes or anomalies in network usage

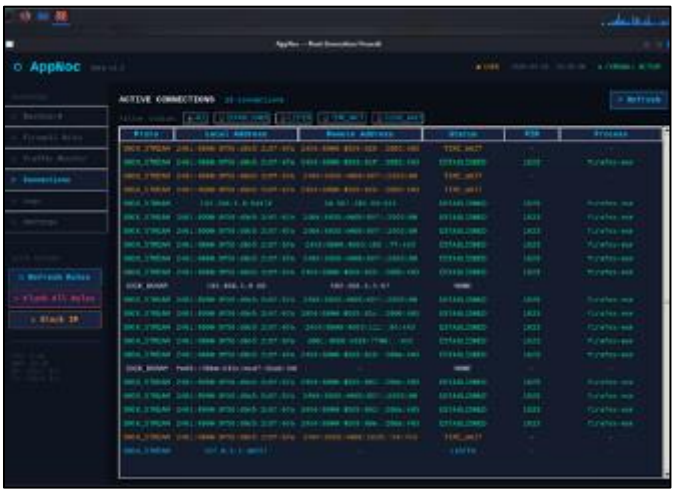

Figure 4 Active Connections

5.1. Columns

- Proto: SOCK_STREAM (TCP)
- Local Address: your system IP + port
- Remote Address: destination IP + port (mostly port 443 = HTTPS)

5.1.1. Status

- ESTABLISHED → active connection
- TIME_WAIT → recently closed connection
- LISTEN → waiting for incoming connection
- PID and Process: e.g., fireboxes
- Tracking all outgoing connections (mostly browser traffic)
- Shows which process is responsible

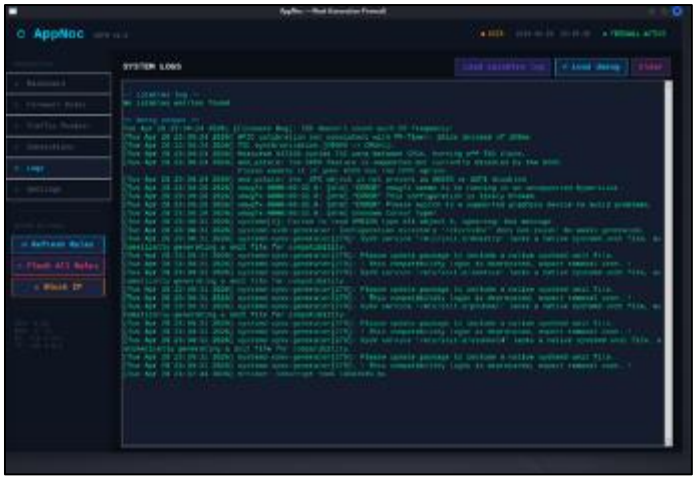

Figure 5 System Logs

5.1.2. Key observations

- No iptables logs found → no firewall rule activity yet
- Kernel messages like:
- CPU timing issues
- BIOS/virtualization warnings
- System service messages
- Collecting low-level system events
- Helps in troubleshooting + forensic analysis

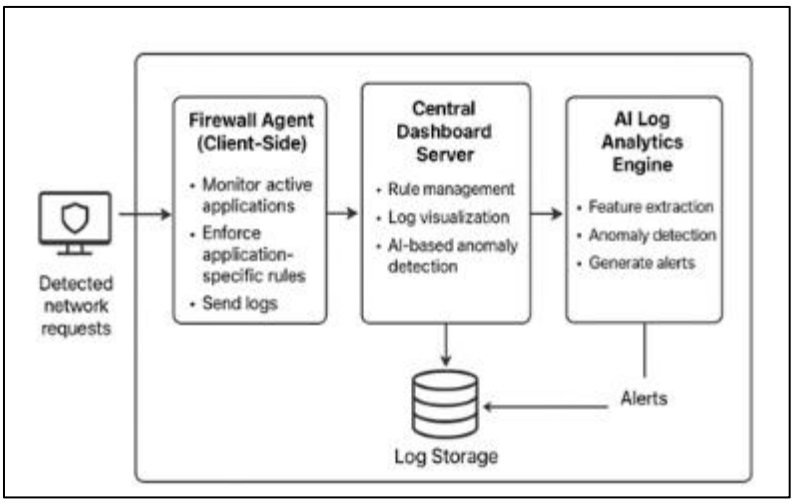

Figure 6 Process

6. Conclusion

App Noc successfully demonstrates the integration of application-aware firewall mechanisms with AI-driven log analysis to deliver a proactive network security solution. Unlike conventional firewalls, it differentiates between normal and suspicious application behavior, enabling precise control over network access. The system effectively monitors live traffic, enforces customized rules, and detects anomalies using machine learning techniques with minimal resource consumption. By combining preventive rule enforcement with predictive analytics, App Noc offers a scalable, cost-effective, and practical solution suitable for enterprises, research labs, and academic institutions. Future work may include real-time rule synchronization, adoption of deep learning models, and support for mobile and IoT ecosystems to further enhance its capabilities.

Compliance with ethical standards

Acknowledgments

We appreciate the provisions extended MIT ADT University, Pune in respect of School of computing, other unidentified facilities along with the immense support extended by the family and friends which has been imperative in achieving the project objectives.

References

[1] Denning, D. E. (1987). An Intrusion Detection Model. IEEE Transactions on Software Engineering, 13(2), 222–232.

[2] Axelsson, S. (2000). Intrusion Detection Systems: A Survey and Taxonomy. Technical Report, Chalmers University of Technology.

[3] Lippmann, R., Fried, D., Graf, I., et al. (2000). Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation. DARPA Information Survivability Conference and Exposition.

[4] Kim, G., Lee, S., and Kim, S. (2014). A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection. Expert Systems with Applications, 41(4), 1690–1700.

[5] Garfinkel, S., and Spafford, G. (2002). Practical UNIX and Internet Security (3rd ed.). O'Reilly Media.

[6] Behl, A., and Behl, K. (2017). Cybersecurity and Cyberwar: What Everyone Needs to Know. Oxford University Press.

[7] Patcha, A., and Park, J. M. (2007). An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Computer Networks, 51(12), 3448–3470.

[8] Sommer, R., and Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy, 305–316.

[9] Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. IEEE Symposium on Computational Intelligence in Security and Defense Applications.

[10] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., and Rajarajan, M. (2013). A Survey of Intrusion Detection Techniques in Cloud. Journal of Network and Computer Applications, 36(1), 42–57.

[11] Shone, N., Ngoc, T. N., Phai, V. D., and Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50.

[12] Sahu, S., and Mehtre, B. M. (2015). Network Intrusion Detection System Using J48 Decision Tree. IEEE International Conference on Advances in Computing, Communications and Informatics.

[13] Aljawarneh, S., Aldwairi, M., and Yassein, M. B. (2018). Anomaly-Based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model. Journal of Computational Science, 25, 152–160.

[14] Buczak, A. L., and Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials, 18(2), 1153–1176.

[15] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., and Hotho, A. (2019). A Survey of Network-Based Intrusion Detection Data Sets. Computers and Security, 86, 147–167.

[16] Ahmed, M., Mahmood, A. N., and Hu, J. (2016). A Survey of Network Anomaly Detection Techniques. Journal of Network and Computer Applications, 60, 19–31.

[17] Conti, M., Dehghantanha, A., Franke, K., and Watson, S. (2018). Internet of Things Security and Forensics: Challenges and Opportunities. Future Generation Computer Systems, 78, 544–546.

[18] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), 15.

[19] Zhang, Y., Chen, X., Li, J., et al. (2019). Network Intrusion Detection Based on Stacked Autoencoder and Isolation Forest. IEEE Access, 7, 123–134.

[20] Javaid, A., Niyaz, Q., Sun, W., and Alam, M. (2016). A Deep Learning Approach for Network Intrusion Detection System. Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies.