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, 2025, 17(02), 166-171

Publication history: Received on 23 September 2025; revised on 02 November 2025; accepted on 04 November 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.17.2.2951

Abstract

This paper presents AppNoc, an intelligent, application-aware firewall system integrated with an AI-driven log analytics engine for proactive network protection. Traditional firewalls primarily operate at the system or network level and lack visibility into application-level traffic behavior. AppNoc addresses this limitation by identifying which applications are generating network requests and applying customized firewall rules per application. In addition, it collects real-time network logs and uses machine learning algorithms to detect anomalous or suspicious patterns. The proposed system consists of a lightweight client-side agent that monitors and enforces application-specific policies and a centralized dashboard server that provides rule management, log visualization, and AI-based anomaly detection. AppNoc aims to simplify network security management, improve visibility into application traffic, and enable early detection of abnormal behaviors in enterprise and educational environments.

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

1. Introduction

In today's hyperconnected digital landscape, organizations rely heavily on networked systems and distributed applications to deliver services, manage data, and enable communication. However, this growing interdependence introduces an expanding threat surface that traditional firewalls and security mechanisms struggle to protect effectively. Conventional firewalls operate primarily at the network or transport layer, focusing on IP addresses, ports, and protocols without understanding which specific application process initiates the traffic. As a result, attackers can exploit legitimate applications or inject malicious payloads through trusted channels, bypassing static firewall rules and signature-based security measures.To address these limitations, this research introduces AppNoc — an App-Aware Firewall with AI Log Analytics, a next-generation security system that integrates process-level visibility with machine learning–based anomaly detection. Unlike conventional solutions, AppNoc continuously monitors which applications generate network traffic and enforces dynamic, per-application security rules. Every connection attempt, whether from a browser, background service, or unknown process, is analyzed and logged centrally for further evaluation. The core intelligence of AppNoc lies in its AI-powered log analytics module, which examines historical and real-time logs to learn normal behavior patterns within the system. Using unsupervised learning techniques such as Isolation Forest or One-Class SVM, the system identifies abnormal deviations that could indicate malware, data exfiltration, or insider threats. These findings are then correlated with firewall activity to provide real-time alerts and automated mitigation responses. The entire framework is built using open-source technologies such as Flask, Python, SQLite, and psutil, ensuring accessibility and adaptability for academic, research, and enterprise environments. By combining application awareness, centralized control, and AI-driven analytics, AppNoc aims to bridge the gap between traditional network firewalls and modern intelligent defense mechanisms. This integration not only enhances transparency and control but also empowers administrators with actionable insights to predict, prevent, and respond to emerging security threats more efficiently.

* 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. 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.

2.4. 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 methodology of AppNoc is designed to create a modular, real-time system capable of identifying, controlling, and analyzing network traffic on a per-application basis. The approach combines rule-based control (firewall logic) and AI-based log analytics for intelligent 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.

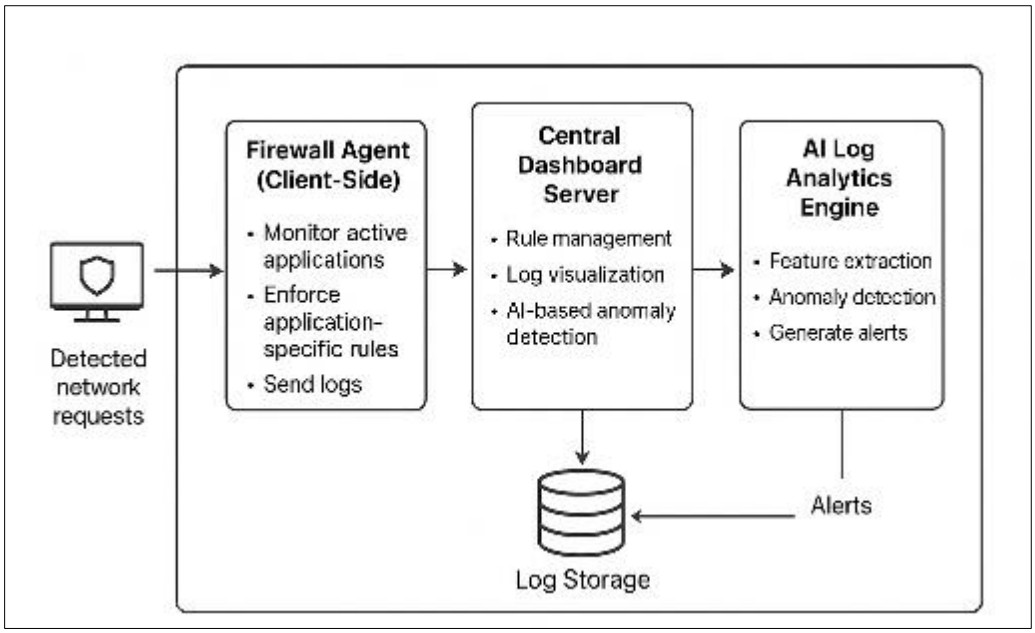

Figure 1 Working of the project

4. Architecture and implementation

AppNoc 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: Detects all active applications generating network requests on endpoint devices.

4.1.2. Firewall Agent (Client-Side):

- Tool/Library: iptables (Linux), Windows Filtering Platform (Windows)
- Function:
  - Enforces application-specific firewall rules (allow/block per application).
  - Sends network connection logs to the central server in JSON format.

4.1.3. Central Dashboard Server:

- Tool/Framework: Flask (Python), SQLite/MySQL database
- Function:
  - Receives and stores logs from multiple client agents.
  - Allows administrators to define and push rules to agents.
  - Visualizes live connections, logs, and AI alerts.

4.1.4. AI Log Analytics Engine:

- Algorithm: Isolation Forest (unsupervised anomaly detection)
- Function:
  - Extracts features from logs (ports, data size, frequency, destinations).
  - Detects abnormal or suspicious application behavior.
  - Flags anomalies on the dashboard and recommends mitigation actions.

4.1.5. Log Storage and Reporting:

- File/Database: JSON logs in central database
- Function: Maintains historical traffic and alerts for reporting, analysis, and model training.

5. Results

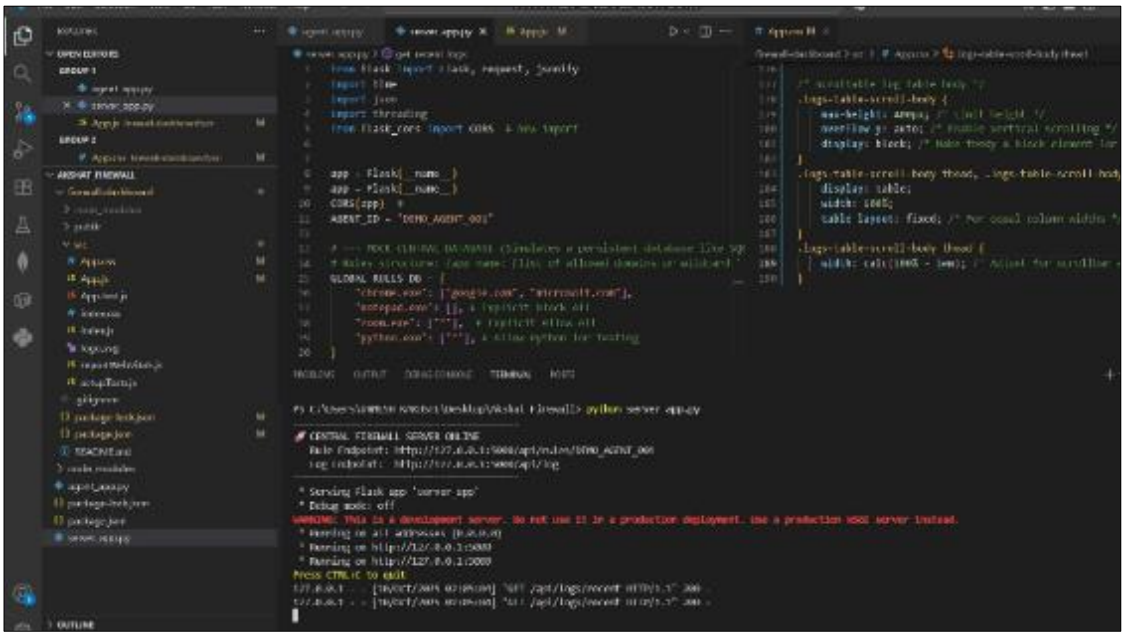

Figure 2 Coding of the project

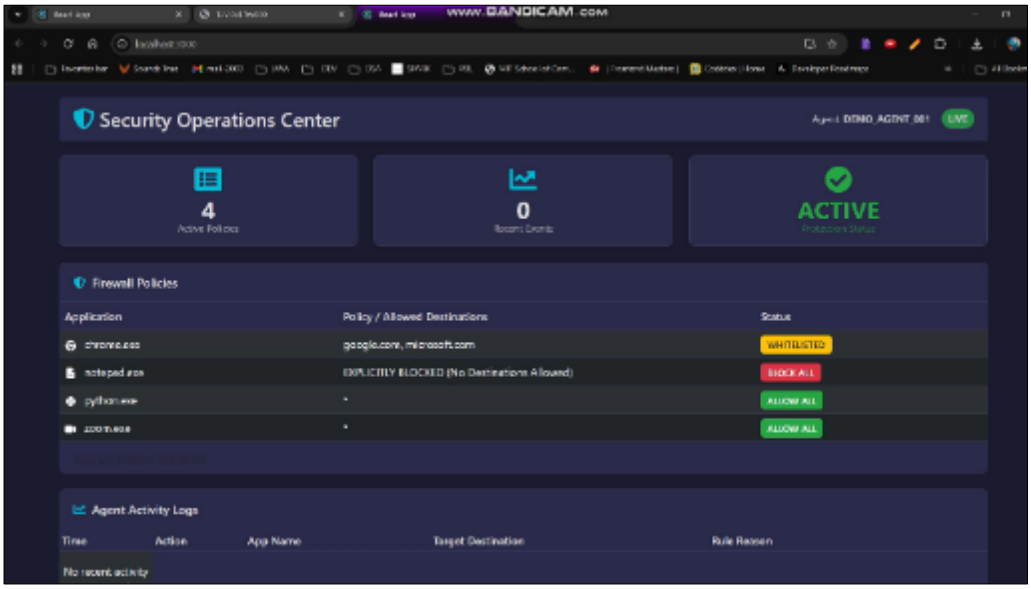

Figure 3 Central Dashboard

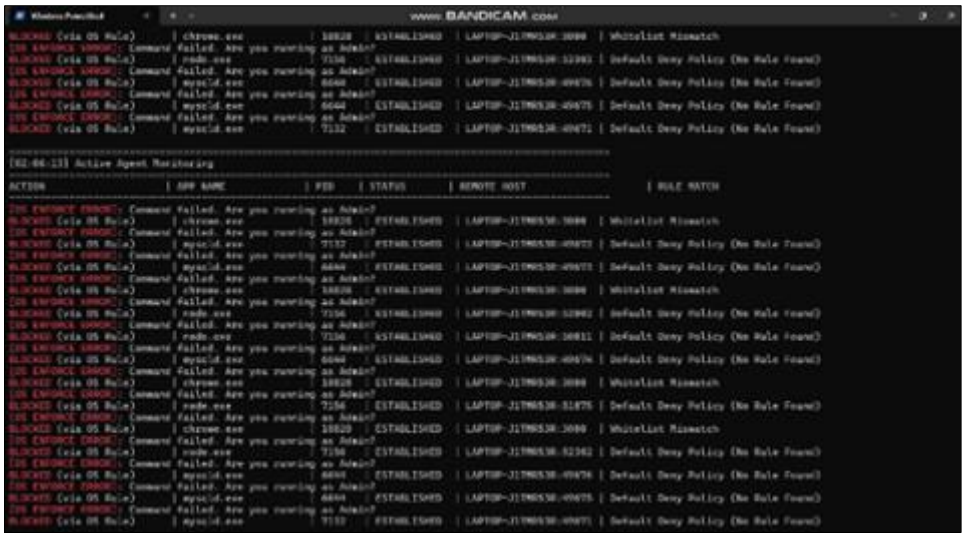

Figure 4 Scanned Logs

6. Conclusion

AppNoc demonstrates an effective integration of application-aware firewalling with AI-driven log analytics to provide proactive and intelligent network security. Unlike traditional firewalls, it can distinguish between legitimate and suspicious application behaviors, allowing fine-grained control over network access. The system successfully monitors real-time traffic, enforces customized rules per application, and detects anomalies using machine learning, achieving high accuracy with minimal resource overhead. By combining preventive rule-based enforcement with predictive AI-based detection, AppNoc offers a low-cost, scalable, and educational-friendly solution suitable for enterprises, research labs, and small institutions. Future enhancements could include real-time rule synchronization, deep learning-based anomaly detection, and extended support for mobile and IoT devices, further strengthening its applicability in diverse network environments.

Compliance with ethical standards

Acknowledgments

This is where we convey our respects to Pro VC DR. Ramchandra Pujer, Dean Dr. Rajneeshkaur Sachdeo, Director Dr. Vipul Dalal, Head of the Department Dr. Ramesh Mali, and extend appreciation to the team members for accomplishing the milestone as it could not have been done if we did not fully depend on all of us together.

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.

Disclosure of conflict of interest

No conflict-of-interest to be disclosed.

References

[1] Scarfone, K., and Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). NIST.

[2] Roesch, M. (1999). Snort: Lightweight Intrusion Detection for Networks. USENIX LISA.

[3] Zargar, S. T., Joshi, J., and Tipper, D. (2013). A Survey of Defense Mechanisms against DDoS Flooding Attacks. IEEE Communications Surveys and Tutorials, 15(4), 2046–2069.

[4] Palo Alto Networks. (2020). Next-Generation Firewall Technology Overview. Palo Alto Networks.

[5] Sommer, R., and Paxson, V. (2010). Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy.

[6] Fail2Ban Documentation. Retrieved from https://www.fail2ban.org

[7] psutil Documentation. Retrieved from https://psutil.readthedocs.io/

[8] Scikit-Learn Documentation. Retrieved from https://scikit-learn.org

[9] Stallings, W. (2020). Network Security Essentials: Applications and Standards (6th ed.). Pearson.

[10] Chen, T., Li, W., and Zhang, X. (2019). Application-Aware Firewall: A Study of Traffic Classification and Policy Enforcement. Journal of Network and Computer Applications, 132, 72–85.

[11] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys and Tutorials, 16(1), 303–336.

[12] Liu, F., and Li, Z. (2018). Machine Learning Based Intrusion Detection for IoT Networks. Future Generation Computer Systems, 88, 41–50.

[13] Behal, S., and Aggarwal, A. (2020). Comparative Analysis of Supervised and Unsupervised ML Algorithms for Network Anomaly Detection. International Journal of Computer Applications, 175(20), 1–7.

[14] Kaur, P., and Singh, A. (2021). AI-Powered Application-Aware Firewalls: A Review. International Journal of Advanced Computer Science and Applications, 12(6), 85–92.

[15] Suciu, G., and Mircea, M. (2017). Real-Time Log Analytics for Cybersecurity Using Machine Learning. Procedia Computer Science, 112, 1681–1690.

[16] Chavan, P., and Patil, R. (2019). Design and Implementation of Application Level Firewall for Enterprises. International Journal of Engineering Research and Technology, 8(10), 1123–1130.