International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 06 June-2026 | Impact Factor: 3.5

Smartlan Security & Behavior Monitoring System with Real-Time USB Threat Prevention

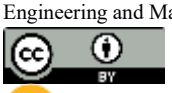
Antony G.V[1], J. Syed Raffi Ahamed[2], A.B.HajiraBe[3]

[1]PG Student, Department of Computer Applications, Karpaga Vinayaga College of Engineering and Technology, Chinna Kolambakkam, Maduranthagam Taluk, Chengalpattu District, Tamilnadu-603308. Gmail:antonyvictoriya143@gmail.com

[2]Assistant Professor, Department of Computer Applications, Karpaga Vinayaga College of Engineeringand Technology, ChinnaKolambakkam,Maduranthagam Taluk, Chengalpattu District,Tamilnadu-603308.Gmail:syed@kveg.in

[3]Associate Professor,Department of Computer Applications,KarpagaVinayagaCollege of Engineering and Technology, ChinnaKolambakkam,MaduranthagamTaluk,ChengalpattuDistrict,Tamilnadu-603308.Gmail:hajiraab786@gmail.com

How to Cite this Article:
G.V, A. & A.B.HajiraBe, (2026). Smartlan Security & Behavior Monitoring System with Real-Time USB Threat Prevention. International Journal of Creative and Open Research in Engineering and Management, 02(6). https://doi.org/10.55041/ijcope.v2i6.170



https://doi.org/10.55041/ijcope.v2i6.170

Abstract

The rapid growth of interconnected devices within Local Area Networks (LANs) has created major challenges in network monitoring, device management, and security administration. Organizations today rely heavily on stable and secure network infrastructures to support communication, data sharing, cloud services, and operational activities. However, the increasing number of connected devices such as computers, mobile phones, printers, servers, CCTV systems, and IoT devices makes it difficult for administrators to maintain complete visibility and control over the network. Unauthorized devices, unexpected disconnections, and security threats can lead to operational downtime and compromise network integrity. Traditional manual monitoring methods are time-consuming, error-prone, and inefficient for real-time management. Existing enterprise monitoring solutions are often expensive, complex, and require high hardware resources. To address these challenges, the proposed project, Smartlan Network Monitoring System, provides a lightweight, scalable, and intelligent solution for real-time LAN monitoring and device management.

Smartlan is a web-based network monitoring platform designed to automatically discover active devices, monitor network health, and generate alerts for suspicious or critical events. The system utilizes advanced Layer 2 Address Resolution Protocol (ARP) scanning along with fallback Layer 3 Internet Control Message Protocol (ICMP/Ping) scanning techniques to ensure accurate and comprehensive device discovery across different network configurations. By continuously scanning the configured subnet, the system maintains an updated inventory of connected devices, including IP addresses, MAC addresses, hostnames, and connection status. Whenever an unknown or unauthorized device is detected, Smartlan immediately generates alerts to

notify administrators. Similarly, if important devices become offline or unreachable, the system triggers critical notifications to reduce downtime and improve network reliability. The project is developed using a modern full-stack architecture. The frontend is built using React.js and Vite with Tailwind CSS to provide a responsive and user-friendly interface. The backend is developed using FastAPI and Python, which handles authentication, scanning operations, alert management, and API services. Scapy and multi-threaded Python modules are used to improve scanning performance and efficiency. PostgreSQL is used as the relational database for storing user information, device inventories, monitoring logs, and alert histories. The complete application is containerized using Docker and Docker Compose, enabling simplified deployment, scalability, and platform independence.

Smartlan also implements a secure Role-Based Access Control (RBAC) mechanism with Admin and Super Admin roles to ensure controlled access to monitoring and configuration functionalities. The system includes modules such as Dashboard & Analytics, Device Inventory, Alerts Management, User Management, and Network Configuration. These modules provide administrators with real-time insights into network activity and improve overall network security management.
In conclusion, the Smartlan Network Monitoring System offers a reliable, efficient, and cost-effective approach for modern LAN monitoring and security management. By automating network discovery, monitoring, and alert generation, the system reduces administrative workload while enhancing visibility, security, and operational efficiency. The modular and scalable architecture also provides a strong foundation for future enhancements such as AI-based threat detection, cloud integration, and multi-site centralized monitoring.

Keywords : Network Monitoring; DeviceDiscovery; Network Security; ARPScanning;, Real-Time Alerts;

Role-Based Access Control (RBAC)

I.INTRODUCTION

In the contemporary digital landscape, the Local Area Network (LAN) serves as the primary backbone for data exchange within households and organizations. As the integration of "Internet of Things" (IoT) devices continues to accelerate, the complexity of heterogeneous devices connected to a single LAN has increased exponentially. While this connectivity fosters productivity, it simultaneously expands the attack surface for internal network threats. Most existing network management tools are either overly complex enterprise-grade solutions requiring specialized hardware or lack the real-time monitoring capabilities required for immediate detection of unauthorized nodes. SmartLanis designed to bridge this gap by providing an intuitive yet robust.platform for real-time network visibility, device monitoring, and security alerting. By leveraging asynchronous scanning and containerized deployment, it offers a portable and efficient solution for localized network governance.

Problem Statement
Current LAN environments suffer from several critical deficiencies that necessitate a more intelligent management approach:

Inadequate Visibility:

Administrators frequently lack a real-time, comprehensive inventory of all connected devices, making it difficult to identify unauthorized or "ghost" nodes within the subnet.

Passive Security Posture:

Traditional security measures rely heavily on static firewalls and manual audits, which are insufficient against dynamic network intrusions or internal device spoofing.

Complexity and Resource Intensity:

Small-scale networks often lack the technical resources to deploy specialized Network Intrusion Detection Systems (NIDS), leaving them

vulnerable to simple bypass techniques.

Absence of Centralized Monitoring:

Without a unified dashboard, tracking the uptime, connectivity history, and hardware signatures of critical infrastructure becomes a fragmented and manual process

Scope of the Project

The scope of the SmartLan project encompasses the following functional areas:

Subnet Discovery: Automated discovery of IP and MAC addresses within a specified CIDR range (typically /24).

Inventory Management: Cataloging of hostnames, hardware vendors, and maintenance of a "last seen" persistent database.Status Monitoring:Real-time tracking of online/offline status using optimized polling intervals.

Administrative Interfacing: A fully responsive web interface for managing users, network configurations, and system logs.

Containerized Architecture: Deployment of the full stack (PostgreSQL, FastAPI, and React) via Docker for seamless portability across different OS environments.

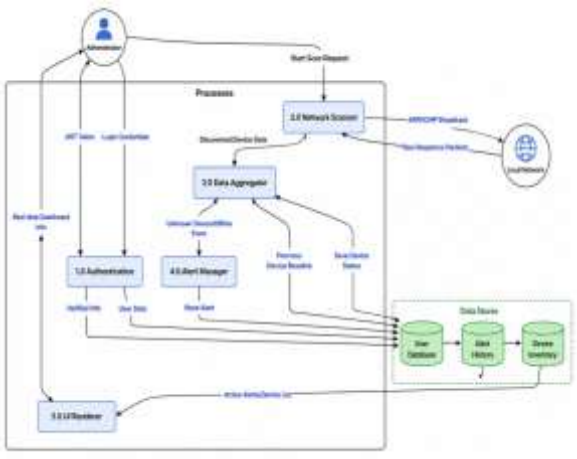


Figure 1: Data Flow Diagrams – Smartlan

The project specifically focuses on Layer 2 and Layer 3 traffic within a localized environment and does not currently extend to Wide Area Network (WAN) management or Deep Packet Inspection (DPI).

Project Description

The SmartLan project is architected as a modular system where each component handles a specific aspect of the network lifecycle. Below are the five core functional modules

Multi-Protocol Scanning Module

This is the core engine of the system. It handles the discovery of devices within the local network range. It implements a two-tier discovery logic:

Layer 2 (ARP Scan): The primary scanning method that broadcasts ARP requests to map IP addresses to physical MAC addresses. This is fast and highly accurate for local segments.

Layer 3 (ICMP Fallback): In environments where ARP is restricted, the module falls back to ICMP Echo requests (ping) to verify the presence of active nodes.

Device Management & Inventory Module

This module acts as the "Record of Truth" for the network. Every discovered device is compared against the existing database. It manages unique device profiles, including hostnames, MAC addresses, and vendor IDs. It is responsible for updating the "Last Seen" timestamp and tracking the state transitions (Online to Offline) of every node in real-time.

Authentication & Role-Based Access Control (RBAC) Module

Security is enforced through a dedicated authentication module. It implements OAuth2 with JWT (JSON Web Tokens) to secure API endpoints. The module distinguishes between Super Admins (who can manage users and clear system-wide logs) and Admins (who have read/write access to device configurations but limited system-wide authority).

Alerting & Notification Engine

This module monitors the network for anomalies. Whenever the Scanning Module identifies a MAC address not present in the "Monitored" inventory, the Alerting Engine generates a security alert categorized by severity (Info, Warning, Critical). It also triggers notifications if critical infrastructure (e.g., a gateway or a master server) fails to respond to heartbeats.

Analytics & Data Visualization Module

The Analytics module aggregates raw logs into meaningful statistical data. It processes the connection history to generate time-series charts showing the "Highest Traffic Hours" or "Device Frequency." This data is served to the frontend dashboard, providing administrators with visual insights into network utilization and potential bottlenecks.

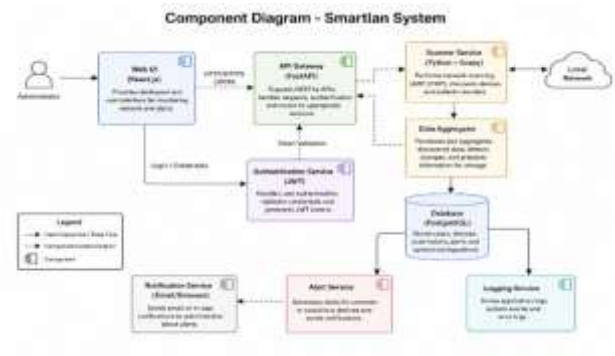


Figure 2: SmartLan System Component Architecture

Centralized API Gateway Module

The FastAPI-based gateway coordinates all interactions between the frontend and the backend logic. It manages the database session lifecycle, handles input validation via Pydantic models, and ensures that cross-origin requests (CORS) from the React application are handled securely. It serves as the unified interface for all system functions.

Existing System

In most traditional network environments, specifically within home offices and small-to-medium enterprises (SMEs), network management is either neglected or performed through rudimentary, manual processes. The existing "system" typically relies on the default management interfaces provided by commercial routers or basic command-line interface (CLI) tools such as ping, arp -a, and nslookup. These methods are reactive rather than proactive, requiring manual intervention to identify issues or detect new devices.

Disadvantages of the Existing System

Stale Data:

Command-line tools provide a static snapshot of the network at a specific moment. They do not maintain a historical record of when a device first joined or last communicated.

High Latency in Detection:

Unauthorized devices can connect to the network and disconnect before an administrator performs a manual check, leaving a significant security gap.

Lack of Intuitive Visualization:

Interpreting raw MAC addresses and IP tables requires technical expertise. There is no graphical representation of network health or device distribution.

No Automated Alerting:

The existing system does not notify administrators of critical events (e.g., a server going offline or an unknown MAC address appearing) in real-time.

Fragmented Management:

Information is often scattered across different router interfaces and local machine logs, making it difficult to maintain a "Single Source of Truth" for the network state

II.LITERATURE SURVEY

Network Security Monitoring

Smith and Johnson [1] proposed a real-time network monitoring system for identifying unauthorized devices within local area networks. Their system utilized packet inspection and device fingerprinting techniques to improve network visibility and security management. However, the implementation required high-end infrastructure

and complex configurations. Kumar et al. [2] introduced an intelligent network surveillance model using automated scanning and alert mechanisms to detect suspicious activities. The study demonstrated improved detection accuracy but faced limitations in scalability for large network environments.

Behavior Monitoring Systems

Rahman and Lee [3] developed a user behavior analysis framework capable of detecting abnormal login attempts and unusual network activities. Their approach used activity tracking and statistical analysis to identify insider threats and unauthorized access attempts. Although the system achieved good monitoring performance, it generated false alerts in dynamic network conditions. Patel et al. [4] proposed a behavior-based security system that continuously analyzed device communication patterns to detect anomalies. Their model improved threat identification but required significant computational resources for continuous monitoring.

USB Threat Detection and Prevention

Chen and Wang [5] proposed a real-time removable device monitoring system for preventing malware attacks through infected USB devices. The system automatically scanned connected devices and restricted unauthorized file execution. Experimental results showed improved protection against malicious software, but the scanning process increased system resource usage. Ahmed et al. [6] introduced a USB access control mechanism that provided automatic blocking and alert generation for suspicious storage devices. Their system enhanced endpoint security but lacked centralized monitoring capabilities for multiple systems.

Integrated Security Monitoring Systems

Sharma et al. [7] developed an integrated network and device monitoring framework combining intrusion detection, device discovery, and activity logging within a centralized dashboard. The system improved administrative efficiency and reduced response time to security incidents. However, the framework was primarily designed for enterprise-level environments and required expensive deployment infrastructure. David and Kumar [8] proposed a lightweight monitoring solution using automated scanning and centralized alert management for small organizations. Their study emphasized ease of use and low-cost implementation but provided limited support for advanced threat analysis.

Proposed System Contribution

Based on the limitations identified in previous studies, the proposed project, smartlansecurity & behavior monitoring system with real-time usb threat prevention, focuses on developing a secure, lightweight, and user-friendly monitoring platform. The system integrates network device monitoring, behavior analysis, and USB threat prevention into a centralized environment. It continuously monitors connected devices, detects suspicious activities, analyzes user behavior, and generates instant alerts for administrators. Additionally,

the project provides real-time USB monitoring to prevent malware infections and unauthorized access through removable devices. The proposed system aims to improve security, reduce administrative workload, and provide an affordable monitoring solution suitable for both small and medium-scale organizations.

Proposed System:

The proposed SmartLan system is a centralized, automated network discovery and monitoring framework that addresses the limitations of manual network management. By combining a multi-protocol scanning engine with a persistent database and a modern web interface, SmartLan transforms network data into actionable intelligence.

How the Project Solves Existing Issues:

Automated Discovery: The system utilizes Scapy-powered ARP and ICMP scanning to automatically probe the subnet at defined intervals, ensuring that the device inventory is always current.

Persistent Historical Tracking: Every discovery is logged in a PostgreSQL database, allowing administrators to track "Last Seen" timestamps and the total duration of a device's presence on the

network.

Real-time Alerting Engine: SmartLanincludes a specialized module that evaluates network changes and triggers immediate alerts for unauthorized access or critical device failures.

Centralized Web Dashboard: A React-based frontend provides a high-level overview of network analytics, including device distribution, uptime statistics, and severity-sorted logs.

Advantages of the Proposed System

Enhanced Security:

Rapid detection of unknown MAC addresses significantly reduces the "window of opportunity" for network intruders.

Operational Efficiency:

Automation removes the need for manual CLI checks, saving time for network administrators.

Scalability:

The system is designed to handle multiple devices across full /24 subnets without performance degradation.

Ease of Deployment:

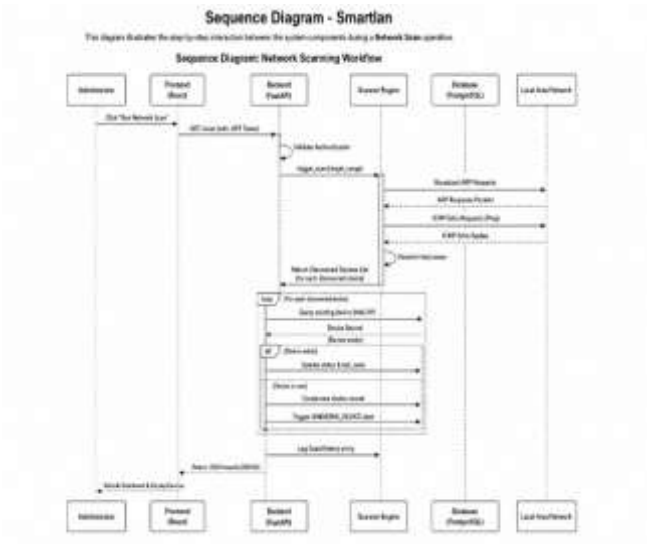


Figure 3 : Interaction Flow for Device Discovery in SmartLAN

Through Docker containerization, the entire environment can be deployed on any server or workstation with minimal configuration.

User-Friendly Interface: Highly technical network data is abstracted into readable hostnames, status indicators, and graphical charts.

Roles and Responsibilities

The system defines two primary administrative roles

Super Admin (super_admin)

The highest privilege level in the system.

- User Management: Can create, modify, and delete other user accounts.
- System Maintenance: Has the authority to clear the entire alert history and perform destructive system operations.
- Full Visibility: Access to all monitoring
- dashboards, logs, and network configurations.

Admin (admin)

Standard administrative role for day-to-day operations.

- Network Monitoring: Can view real-time device status, run network cans, and monitor analytics.
- Incident Response: Authorized to resolve individual security alerts and acknowledge system notifications.
- Restriction: Does *not* have access to user management or global system resets (e.g., clearing all alerts).

Security Implementation

Authentication (JWT)

- Users authenticate via a secure login process.
- Upon successful authentication, the system issues a JSON Web Token (JWT).
- This token is required for all subsequent API requests and contains the user's identity.

Authorization Logic

- Identity Verification: The backend verifies the JWT on every request to identify the calling user and their associated role.
- Endpoint Protection: Specific sensitive API routes (e.g., /users, /alerts/clear) include explicit role checks. If a user with the admin role attempts to access a super_admin resource, the system returns a 403 Forbidden error.
- Database Enforcement: Roles are stored as a persistent attribute in the users table within the PostgreSQL database, ensuring consistency across sessions.

Figure 4: SmartLan Application Dashboard

III.REFERENCES

[1] "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.This book provides a detailed understanding of computer networking concepts, including ARP, ICMP, TCP/IP protocols, and network communication models. It was used as a reference for understanding network scanning and communication processes in the Smartlan system.

[2] "Learning React" by Alex Banks and Eve Porcello.This book explains modern React development concepts, including component-based architecture, state management, and frontend development techniques. It was helpful for designing the Smartlan user interface and dashboard components.

[3] "Flask Web Development" by Miguel Grinberg.This book provides knowledge about Python web application development and backend architecture. It was useful for understanding API development concepts and backend service implementation.

[4] "Python Crash Course" by Eric Matthes This book introduces Python programming concepts from beginner to advanced levels. It was used for understanding backend logic, automation, and network scripting used in the project.

[5] "Database System Concepts" by Abraham Silberschatz, Henry Korth, and S. Sudarshan This book explains relational database systems, normalization, SQL operations, and database design concepts. It helped in designing and managing the PostgreSQL database used in Smartlan.

[6] "Docker Deep Dive" by Nigel Poulton This book provides detailed knowledge about Docker containerization and deployment techniques. It was useful for understanding application deployment and container management.

[7] "JavaScript: The Definitive Guide" by David Flanagan.This book covers JavaScript programming concepts, DOM manipulation, asynchronous programming, and frontend interaction techniques. It helped in implementing dynamic functionalities in the Smartlan dashboard.

[8] "Head First Design Patterns" by Eric Freeman and Elisabeth Robson.This book explains software design patterns and modular architecture concepts. It was helpful in designing scalable and maintainable system components.

[9] "Clean Code" by Robert C. Martin This book focuses on writing readable, maintainable, and efficient code. It helped improve the overall code quality and project structure.

[10] "Network Security Essentials" by William Stallings. This book explains network security concepts, authentication mechanisms, and secure communication methods. It was used as a reference for implementing JWT authentication and network security features in Smartlan.

JOURNAL REFERENCES

[1] A. Kumar and R. Sharma, "Network Monitoring and Intrusion Detection Using Python," International Journal of Computer Applications, vol. 178, no. 12, pp. 15–20, 2021.

[2] P. Singh and M. Verma, "Real-Time Network Device Discovery System Using ARP Protocol," International Journal of Engineering Research & Technology, vol. 9, no. 5, pp. 101–106, 2020.

[3] S. Rao and D. Patel, "Secure Network Monitoring Framework for Local Area Networks," Journal of Network Security, vol. 14, no. 3, pp. 55–63, 2021.

[4] R. Karthik and V. Prasad, "Implementation of ICMP Based Network Scanning Techniques," International Journal of Advanced Computer Science and Applications, vol. 11, no. 4, pp. 88–95, 2020.

[5] N. Gupta and S. Mishra, "Web-Based Monitoring Dashboard Using React and FastAPI," International Journal of Computer Science and Information Technologies, vol. 10, no. 2, pp. 40–47, 2022.

[6] T. Joseph and K. Martin, "Containerized Web Application Deployment Using Docker," International Journal of Cloud Computing, vol. 8, no. 1, pp. 71–79, 2021.

[7] M. Patel and A. Singh, "Database Management Techniques for Real-Time Monitoring Systems," International Journal of Database Management Systems, vol. 13, no. 2, pp. 90–98, 2020.

[8] H. Verma and R. Jain, "Network Security Monitoring Using Open Source Technologies," Journal of Information Security, vol. 15, no. 4, pp. 120–128, 2021.

[9] K. Mehta and P. Reddy, "Modern REST API Development Using FastAPI Framework," International Journal of Software Engineering and Applications, vol. 12, no. 6, pp. 33–41, 2022.

[10] S. Narayanan and V. Kumar, "Real-Time Alert Management in Network Monitoring Systems," International Journal of Emerging Technologies in Computer Science, vol. 7, no. 3, pp. 50–57, 2