NEXT QUANTUM 2.0

KERNEL PANIC

CYBER SECURITY WIRENET

Team ID- NQ 02:27

Topic Name – Centralized Application-Context Aware Firewall

SANT BABA BHAG SINGH UNIVERSITY

LEARN | ACHIEVE | SUCCEED

Proposed Solution

- AI Firewall: Monitors & restricts app network activity.
- Real-Time Tracking: Analyzes traffic & enforces policies.
- Anomaly Detection: Alerts cybersecurity teams.
- Central Dashboard: Unified management.

Problem Addressed

- Context-Aware Monitoring: Tracks app-specific network activities.
- Centralized Policy Enforcement: Unified firewall rule management.
- AI-Powered Anomaly Detection: Proactive threat alerts.
- Unified Dashboard: Simplifies security management.

Innovation & Uniqueness :

1. **App-wise Control :** Har app ke liye network access manage kar sakte ho.
2. **AI/ML Detection:** Smart AI se threats jaldi pakad mein aate hain.
3. **Central Dashboard:** Ek jagah se sab kuch monitor karo.
4. **Detailed Logs:** App ki network activity ka full record milta hai.
5. **Context-Aware:** Activity ka time, source, aur purpose samjha jaata hai.
6. **Cross-Platform:** Windows support aur Linux ke liye adaptable.
7. **Stronger Security:** User ko full control aur protection milta hai.

Agar aur chhota karna ho to bhi bata dena!

TECHNICAL APPROACH

Methodology

1. **Agent Installation:** Deploy Python-based firewall on endpoints.
2. **Real-Time Monitoring:** Track app network activity (IP, domains, protocols).
3. **Policy Enforcement:** Apply centralized firewall rules per endpoint.
4. **Data Transmission:** Securely send logs to Flask backend via REST APIs.
5. **AI/ML Anomaly Detection:** Identify threats using AI models.
6. **Dashboard Visualization:** Display security insights dynamically.
7. **Alert & Response**: Notify security teams for action.

Tech Stack

1. **Frontend:** HTML5 (structure), CSS3 (styling), JavaScript (dynamic updates).
2. **Backend:** Python (Flask) for REST API, security, and real-time processing.
3. **AI/ML:** Anomaly detection & alert system.
4. **Database**: SQLite/MongoDB for policy & log storage (optional).

FEASIBILITY AND VIABILITY

Feasibility Analysis

- **Technical Feasibility:** Uses proven tech (Python, Flask, AI/ML) with seamless integration.
- **Operational:** Lightweight agent deployment; centralized policy management.
- **Economic:** Open-source, cost-effective, reduces cybersecurity expenses.
- **Scalability:** Supports multiple endpoints with future upgrade flexibility.
- **Risk Management:** Addresses performance and security risks with mitigation strategies.

Challenges & Mitigation

- **Performance Overhead** → Optimized, lightweight agent design.
- **False Positives** → Continuous AI model refinement.
- **Security Risks** → Strong encryption & authentication for the console.
- **Data Privacy** → Compliance with GDPR, ISO 27001.
- **Integration Complexity** → Flexible, modular policy management.

IMPACT:

•**Enhanced Security Posture:** Empowers cybersecurity teams with precise control and proactive defense against network threats.
•**Improved Threat Response:** Enables faster identification, response, and mitigation of abnormal application behaviors through AI-driven alerts.
•**Reduced Management Complexity:** Streamlines firewall policy management across endpoints through a unified central dashboard, enhancing productivity.
•**Better Resource Utilization:** Optimized monitoring ensures minimal system impact, preserving endpoint performance and stability.

BENEFITS:

- **Social Benefits** : Increased cybersecurity awareness and proactive protection of user privacy and sensitive data.
- **Economic Benefits :** Significant cost savings by preventing security breaches, reducing downtime, and lowering incident-response expenditures.
- **Operational Benefits :** Centralized management reducing manual efforts, enabling efficient allocation of cybersecurity personnel resources.
- **Environmental Benefits** : Efficient software design reduces unnecessary computing load, lowering overall energy consumption.

TECHNOLOGIES USED-

- **Flask (REST API):** Lightweight Python backend framework.
- **Isolation Forest (Anomaly Detection)**: ML-based outlier detection.
- **Endpoint Security:** Application-layer firewall protection.
- **Centralized Network Management**: Efficient, policy-driven control.
- **Real-time Anomaly Detection**: AI-driven cybersecurity monitoring.
- **Web Technologies**: HTML (structure), CSS (style), JS (interactivity).

THANK YOU

DEVASHISH
B-Tech/CSE
8th SEM

AMAN KUMAR
B-Tech/CSE
4th SEM

ABHISHEK
B-Tech/CSE
4th SEM

GAURAV SEHGAL
B-Tech/CSE
4th SEM