Endpoint Command Execution Monitoring and Alerting Tool Machine Learning-Based Network Anomaly Detector

Kalaivani M[1], Muhammad Ashiq H[2], Dr.T. Ramaprabha[3]
[1,2]*III B.Sc Digital and Cyber Forensic Science, Department of Digital and Cyber Forensic Science, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India.*
[3]*Associate Professor, Department of Digital and Cyber Forensic Science, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India.*

***Abstract*—The increasing sophistication of cyber threats has made traditional signature-based security mechanisms insufficient for protecting modern enterprise environments. Attackers frequently exploit endpoint systems through unauthorized command execution, privilege escalation, and lateral movement, while simultaneously launching network-based attacks such as data exfiltration, distributed denial-of-service (DDoS), and advanced persistent threats (APTs). To address these challenges, this paper proposes an integrated Endpoint Command Execution Monitoring and Alerting Tool combined with a Machine Learning-Based Network Anomaly Detector. The endpoint monitoring component continuously tracks command-line activities, user privileges, process hierarchies, and execution patterns to detect suspicious behavior in real time. Simultaneously, the network anomaly detection module analyzes traffic flow characteristics using machine learning algorithms to identify abnormal patterns indicative of cyberattacks. The system employs supervised and unsupervised learning techniques, including Random Forest, Support Vector Machine (SVM), Isolation Forest, and LSTM models for sequential pattern recognition. Feature engineering incorporates command frequency analysis, abnormal execution timing, packet size variations, protocol usage anomalies, and authentication irregularities. Experimental evaluation demonstrates high detection accuracy, low false positive rates, and near real-time alert generation. The integrated framework enhances threat visibility, improves zero-day attack detection, and reduces the operational burden on security teams. The proposed solution offers a scalable, adaptive, and intelligent cybersecurity defense mechanism suitable for enterprise networks, government institutions, and critical infrastructure environments.**

***Index Terms*—Endpoint Security, Network Anomaly Detection, Machine Learning, Cybersecurity, Intrusion Detection System (IDS).**

I. INTRODUCTION

With the rapid expansion of enterprise networks and digital infrastructures, cybersecurity threats have become more sophisticated and persistent. Attackers increasingly exploit endpoint systems through unauthorized command execution, privilege escalation, and lateral movement techniques. At the same time, network-based attacks such as Distributed Denial of Service (DDoS), data exfiltration, and advanced persistent threats (APTs) continue to challenge traditional defense mechanisms.

Conventional security tools such as signature-based antivirus systems and static firewalls are often ineffective against zero-day exploits and stealthy malicious activities. To address these limitations, this article presents an integrated approach combining an Endpoint Command Execution Monitoring and Alerting Tool with a Machine Learning-Based Network Anomaly Detector. Together, these systems provide real-time monitoring, intelligent threat detection, and automated alerting to strengthen organizational cybersecurity posture.

II. SYSTEM OVERVIEW AND ARCHITECTURE

The proposed solution integrates endpoint-level visibility with network-wide intelligence to create a comprehensive cybersecurity defense framework. By

combining command execution monitoring at the host level with machine learning-driven network anomaly detection, the system provides layered security capable of identifying both insider threats and external attacks in real time.

2.1 Endpoint Command Execution Monitoring Tool

The Endpoint Command Execution Monitoring Tool operates directly on individual systems (workstations, servers, and critical infrastructure devices). Its primary objective is to detect suspicious command-level activities that may indicate compromise, misuse, or insider threats.

Core Functionalities

1. Real-Time Command Tracking The tool continuously monitors command-line interpreters (e.g., PowerShell, CMD, Bash) and records all executed commands. This ensures visibility into system-level operations that attackers commonly exploit after gaining access.

2. Detection of Suspicious Administrative Commands The system identifies potentially dangerous commands such as:

- Privilege escalation attempts
- User account modifications
- Firewall rule changes
- File deletion or encryption commands
- Unauthorized script execution

Behavioral baselines are established for normal user activity, and deviations from these baselines trigger alerts.

3. Privilege Escalation Monitoring Unauthorized elevation of privileges is a common attack vector. The tool tracks:

- Changes in user roles
- Sudden switch to administrative accounts
- Use of system-level execution commands

4. Abnormal Execution Pattern Alerts The monitoring agent analyzes contextual indicators such as:

- Execution outside business hours
- High-frequency repetitive commands
- Suspicious command chaining
- Unusual parent-child process relationships

If anomalies are detected, alerts are generated and forwarded to the security dashboard.

Data Collection and Transmission

The monitoring agent collects detailed command metadata, including:

- User ID and system ID
- Timestamp
- Process ID and parent process ID
- Command arguments
- Execution duration
- System resource utilization

To ensure integrity and confidentiality, logs are encrypted and securely transmitted to a centralized analysis server for correlation with network-level events.

2.2 Machine Learning-Based Network Anomaly Detector

While endpoint monitoring focuses on host-level activities, the Network Anomaly Detector provides visibility into traffic patterns across the entire network.

Core Functionalities

1. Packet Inspection and Flow Monitoring The system captures traffic metadata using network sensors or packet capture tools. Instead of deep packet content inspection (which may raise privacy concerns), it primarily analyzes traffic flow characteristics.

2. Feature Extraction Key network features include:

- Packet size distribution
- Protocol usage frequency
- Flow duration
- Byte transmission rate
- Connection frequency
- Failed login attempts
- Source-destination communication patterns

These features help identify suspicious behaviors such as DDoS attempts, port scanning, brute-force attacks, and data exfiltration.

3. Machine Learning-Based Anomaly Detection The system applies supervised and unsupervised algorithms to classify traffic as normal or malicious.

- Supervised models detect known attack signatures based on labeled datasets.
- Unsupervised models identify zero-day or unknown threats by detecting deviations from established normal traffic patterns.

4. Risk Scoring and Alerting Each detected anomaly is assigned a risk score based on severity, confidence level, and potential impact. High-risk events trigger:

- Immediate alerts
- Automated IP blocking
- Endpoint isolation
- Incident report generation

2.3 Multi-Layered Architecture

The overall architecture is designed as a scalable, modular, and layered system:

1. Data Collection Layer

- Endpoint monitoring agents
- Network traffic sensors
- Log aggregation services

This layer gathers raw command and traffic data in real time.

2. Preprocessing and Feature Engineering Layer

- Data normalization
- Noise filtering
- Feature extraction
- Handling missing or corrupted logs

This layer transforms raw data into structured features suitable for machine learning models.

3. Machine Learning Detection Engine

- Model training and validation
- Real-time inference engine
- Behavioral baseline modeling
- Anomaly classification and prediction

This is the intelligence core of the system.

4. Alerting and Response Module

- Security dashboard visualization
- Email/SMS alerts
- SIEM integration
- Automated containment actions

The response module ensures rapid mitigation of detected threats to minimize damage.

Integrated Security Advantage

By correlating endpoint command behavior with network traffic anomalies, the system achieves enhanced detection accuracy. For example, if an unusual administrative command is executed and followed by abnormal outbound traffic, the system identifies a high-confidence security incident.

This dual-layered, machine learning-driven architecture provides proactive defense, improved visibility, and faster incident response, making it suitable for enterprise networks, financial institutions, and critical infrastructure environments.

III. METHODOLOGY AND MACHINE LEARNING APPROACH

The proposed system adopts a hybrid machine learning methodology that integrates both supervised and unsupervised learning techniques to detect endpoint-level threats and network anomalies effectively. This approach ensures the detection of known attack patterns as well as previously unseen (zero-day) threats.

3.1 Data Collection

Accurate detection depends heavily on high-quality and diverse datasets. The system collects data from multiple sources to build a comprehensive threat detection model.

1. Endpoint Command Execution Logs

Endpoint agents continuously record:

- Executed commands and arguments
- User identity and privilege level
- Process hierarchy (parent-child relationships)
- Execution timestamps
- Resource consumption metrics

These logs help distinguish between normal administrative activity and suspicious behavior.

2. Network Traffic Datasets

Network-level data is collected from:

- Live traffic monitoring tools
- NetFlow records
- Packet capture utilities
- Public intrusion detection datasets (for training and benchmarking)

This data includes metadata such as source/destination IP addresses, port numbers, protocols, packet sizes, and connection durations.

3. Normal and Malicious Behavior Samples

To ensure effective learning:

- Normal behavior data is collected over a baseline period.
- Malicious samples include known attack scenarios such as brute-force attempts, privilege escalation, malware communication, port scanning, and lateral movement.

Combining real-world logs with benchmark datasets improves generalization and detection capability.

3.2 Data Preprocessing

Raw logs and traffic records often contain noise, redundant entries, or missing values. Therefore, preprocessing is performed before model training.

Steps include:

- Data cleaning and removal of corrupted logs
- Handling missing values using imputation techniques

- Encoding categorical features (e.g., protocol types, user roles)
- Normalization or scaling of numerical features
- Timestamp standardization
- Log aggregation to reduce redundancy

For sequence-based models like LSTM, time-series windowing techniques are applied to structure sequential data.

3.3 Feature Engineering

Feature engineering plays a critical role in improving model accuracy and reducing false positives.

A. Endpoint Features

1. Frequency of Command Execution Repeated execution of certain commands within short intervals may indicate automated attack scripts.
2. Unusual Command Sequences Abnormal command chaining (e.g., privilege escalation followed by file encryption) may suggest malicious intent.
3. Privilege Escalation AttemptsDetection of sudden changes from normal user privileges to administrative rights.
4. Execution During Non-Business Hours Commands executed outside predefined operational hours may indicate compromised accounts.
5. Process Tree Anomalies Suspicious parent-child process relationships (e.g., office application launching system shell).

B. Network Features

1. Source and Destination IP Patterns Communication with unknown or blacklisted IP addresses.
2. Abnormal Packet Sizes Extremely large or unusually small packet distributions may indicate data exfiltration or scanning.
3. Flow Duration Anomalies Long persistent connections or rapid short bursts may signal command-and-control (C2) activity.
4. Failed Authentication Attempts Repeated login failures may indicate brute-force attacks.
5. Protocol Usage Irregularities Unusual protocol activity compared to baseline traffic.

These engineered features are transformed into structured vectors used by machine learning models.

3.4 Algorithms Used

The system employs a combination of classification and anomaly detection models:

1. Logistic Regression

Used as a baseline model due to its simplicity and interpretability. It provides probability-based classification and helps understand feature influence.

2. Random Forest

An ensemble learning method that handles high-dimensional data effectively. It improves accuracy and reduces overfitting by combining multiple decision trees.

3. Support Vector Machine (SVM)

Effective for high-dimensional feature spaces and suitable for detecting subtle anomalies in structured datasets.

4. Isolation Forest

An unsupervised anomaly detection algorithm that isolates abnormal data points by randomly partitioning data. Particularly useful for detecting zero-day attacks where labeled data may not be available.

5. Long Short-Term Memory (LSTM)

A deep learning model designed for sequential and time-series data. It captures temporal dependencies in:

- Command execution sequences
- Network traffic flows over time

LSTM improves detection of stealthy attacks that unfold gradually.

3.5 Model Training and Validation

To ensure robustness and avoid overfitting, the dataset is divided as follows:

- 70% Training Set – Used to train machine learning models
- 15% Validation Set – Used for hyperparameter tuning and performance optimization
- 15% Testing Set – Used for final evaluation

Additionally, cross-validation techniques may be applied to improve generalization.

Performance is evaluated using:

- Accuracy
- Precision and Recall
- F1-Score
- ROC-AUC
- False Positive and False Negative Rates

3.6 Hybrid Detection Strategy

The final system combines outputs from multiple models using ensemble or weighted scoring techniques. Alerts are generated only when:

- Multiple models confirm suspicious behavior, or
- Risk scores exceed predefined thresholds.

This multi-model strategy significantly reduces false positives while maintaining high detection sensitivity. The hybrid methodology combining structured feature engineering, supervised classification, unsupervised anomaly detection, and deep learning-based sequence analysis ensures comprehensive protection. By correlating endpoint behavior with network anomalies, the system achieves higher detection accuracy, improved zero-day identification, and enhanced security resilience across enterprise environments.

IV. IMPLEMENTATION AND DEPLOYMENT

The proposed system is designed to be modular, scalable, and enterprise-ready. Its implementation leverages widely adopted open-source technologies to ensure flexibility, cost-effectiveness, and ease of integration with existing infrastructure.

4.1 Technology Stack

Programming Language: Python

Python is chosen due to its extensive ecosystem for machine learning, cybersecurity automation, and backend development. It enables rapid development, clean code structure, and seamless integration with monitoring tools and APIs.

Libraries and Frameworks

1. Scikit-learn Used for implementing traditional machine learning algorithms such as:

- Logistic Regression
- Random Forest
- Support Vector Machine
- Isolation Forest

It provides efficient model training, hyperparameter tuning, and evaluation tools.

2. TensorFlow Utilized for deep learning models such as LSTM networks. It enables:

- Sequential traffic pattern learning
- Command execution sequence modeling
- GPU-accelerated training for large datasets

3. NumPy and Pandas These libraries handle:

- Data preprocessing
- Feature engineering
- Log aggregation
- Statistical analysis

They ensure structured transformation of raw logs into model-ready datasets.

Web Framework: Flask or Django

A web-based interface is developed for:

- Real-time dashboard visualization
- Alert management
- Incident tracking
- User authentication and access control

Flask is lightweight and suitable for microservices architecture. Django is preferred when enterprise-level authentication, scalability, and built-in security features are required.

The dashboard provides:

- Threat severity levels
- Historical event logs
- Endpoint status overview
- Network anomaly heatmaps

Database System: PostgreSQL or MongoDB

The system uses a database layer to store logs, alerts, and model outputs.

PostgreSQL (Relational Database):

- Structured storage of endpoint and network logs
- ACID compliance
- Suitable for analytical queries

MongoDB (NoSQL Database):

- Flexible storage of unstructured logs
- Scalable document-based architecture
- Efficient handling of large volumes of JSON-formatted event data

The database ensures:

- Log retention for forensic analysis
- Fast querying for incident investigations
- Audit trail maintenance

Packet Capture Tools: Scapy / Wireshark APIs

Scapy:

- Python-based packet manipulation tool
- Captures and analyzes live network packets
- Extracts traffic metadata for feature engineering

Wireshark APIs:

- Integration for deeper traffic inspection
- Flow-based analysis
- Network protocol identification

These tools focus on metadata extraction to minimize performance impact and privacy concerns.

4.2 Endpoint Monitoring Agent

The endpoint monitoring component runs as a lightweight background service on:

- Windows systems (as a system service)
- Linux systems (as a daemon process)

Agent Characteristics:

- Minimal CPU and memory consumption
- Encrypted log transmission (TLS/SSL)
- Automatic restart mechanism
- Tamper detection and integrity validation

The agent continuously:

- Monitors command-line activity
- Tracks process creation events
- Observes privilege escalation
- Logs system modifications

Data is periodically sent to the centralized server for correlation with network events.

4.3 Network Monitoring Sensors

Network sensors are deployed at:

- Gateway routers
- Core switches
- Firewall monitoring points

These sensors:

- Capture traffic flow metadata
- Extract statistical features
- Forward summarized logs to the analysis server

To avoid bandwidth overhead:

- Full packet payloads are not stored
- Only flow-based metadata is collected
- Sampling techniques may be applied in high-traffic environments

This ensures minimal impact on network performance.

4.4 Alert Generation Mechanism

When suspicious activity is detected, the system generates alerts through multiple channels:

1. Email Notifications

Security administrators receive real-time alerts with:

- Threat description
- Affected endpoint
- Severity score
- Recommended action

2. Dashboard Visualization

A centralized web dashboard displays:

- Live threat feed
- Graphical traffic analysis
- Endpoint health status
- Risk score trends

3. SIEM Integration

The system can integrate with Security Information and Event Management (SIEM) platforms to:

- Correlate logs with other security tools
- Enable centralized monitoring
- Support compliance requirements

4. SMS or Webhook Alerts

For high-severity incidents:

- Instant SMS notifications
- Webhook integration with incident management tools
- Automated ticket generation

4.5 Automated Response Mechanism

To reduce response time and limit damage, the system supports automated containment actions.

1. Terminating Suspicious Processes

If malicious command execution is detected:

- The system immediately kills the process
- Prevents further execution of harmful scripts

2. Blocking IP Addresses

For network-level threats:

- Firewall rules are dynamically updated
- Malicious IP addresses are blocked

3. Endpoint Isolation

Compromised systems can be:

- Temporarily disconnected from the network
- Moved to a quarantine VLAN
- Restricted to limited access mode

This prevents lateral movement and data exfiltration.

4.6 Deployment Strategy

The system supports flexible deployment models:

- On-Premise Deployment: Suitable for government or high-security institutions
- Cloud-Based Deployment: Scalable architecture using virtual machines or containers
- Containerized Deployment: Using Docker for portability and easy updates

Load balancing and horizontal scaling can be implemented for large enterprise environments.

4.7 Security and Reliability Considerations

- Encrypted communication between agents and server
- Role-based access control (RBAC)

- Regular model retraining pipelines
- Log integrity validation
- Backup and disaster recovery mechanisms

The implementation leverages modern technologies and modular architecture to ensure scalability, reliability, and enterprise readiness. By combining lightweight endpoint agents, efficient network monitoring sensors, machine learning intelligence, and automated response capabilities, the system delivers a comprehensive and adaptive cybersecurity defense solution suitable for large-scale deployment.

V. CONCLUSION

The rapid evolution of cyber threats, including insider attacks, privilege escalation, advanced persistent threats (APTs), and zero-day exploits, demands intelligent and adaptive security solutions beyond traditional signature-based systems. This paper presented an integrated Endpoint Command Execution Monitoring and Alerting Tool combined with a Machine Learning-Based Network Anomaly Detector to provide comprehensive, real-time protection across enterprise environments.

The proposed framework enhances endpoint visibility by continuously monitoring command execution patterns, privilege escalations, and abnormal process behaviors. Simultaneously, the network anomaly detection module leverages supervised and unsupervised machine learning algorithms to identify irregular traffic patterns, data exfiltration attempts, brute-force attacks, and command-and-control communications. The hybrid detection strategy, incorporating models such as Random Forest, Support Vector Machine (SVM), Isolation Forest, and LSTM, significantly improves detection accuracy while minimizing false positives.

By correlating endpoint and network-level events, the system achieves high-confidence threat detection and enables rapid incident response through automated containment mechanisms such as process termination, IP blocking, and endpoint isolation. The modular architecture ensures scalability, flexibility, and seamless integration with SIEM platforms and enterprise infrastructures.

Overall, the proposed solution provides a proactive, scalable, and intelligent cybersecurity defense model suitable for enterprise networks, government institutions, financial sectors, and critical infrastructure systems. Future work may focus on federated learning, AI-driven adaptive threat intelligence integration, and advanced behavioral analytics to further enhance detection capabilities.

REFERENCES

[1] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. IEEE Symp. Security and Privacy, 2010.

[2] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[3] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, 1995.

[4] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation Forest,” in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008.

[5] MITRE Corporation, “MITRE ATT&CK framework,” 2023.

[6] National Institute of Standards and Technology, Guide to Intrusion Detection and Prevention Systems (IDPS), 2020.

[7] Canadian Institute for Cybersecurity, “CICIDS2017 dataset,” 2017.

[8] DARPA, “DARPA intrusion detection evaluation dataset,” 1999.

[9] SANS Institute, Incident Handler’s Handbook, 2022.