ISSN: 2634 - 8853

Journal of Engineering and Applied Sciences Technology

SCIENTIFIC Research and Community

Review Article

Open Access

Real-Time Cyber Threat Detection Using Big Data Analytics: A Scalable Framework for Immediate Threat Response

Naveen Edapurath Vijayan

Sr Data Engineering Manger, Amazon Seattle, WA 98765, USA.

ABSTRACT

The increasing complexity of cyber threats across industries such as human resources (HR), financial services, and government agencies necessitates the development of real time security solutions. Traditional security measures such as rule based intrusion detection systems (IDS) and batch processing analytics are inadequate for detecting dynamic and sophisticated cyber threats, including insider attacks, financial fraud, and government infrastructure intrusions. This paper presents a scalable real time cyber threat detection framework that integrates big data analytics, machine learning, and automated incident response mechanisms to detect and mitigate threats in real time. The proposed system consists of secure data ingestion pipelines, distributed real time processing using Apache Kafka and Spark Streaming, and an ensemble machine learning based anomaly detection model. A cross industry use case highlights the framework's ability to detect insider threats in HR systems, fraudulent financial transactions, and cyber espionage in government networks. The system is implemented on AWS cloud infrastructure, and experimental results show that it achieves 98.6% precision for HR insider threat detection, 94.2% accuracy in financial fraud prevention, and a sub-2-second detection latency for public sector security alerts. This paper also discusses the scalability of the system, highlighting its ability to process over 50,000 security events per second while maintaining real-time performance. The results indicate that big data-driven cybersecurity solutions can significantly improve threat detection, response times, and overall security posture across different industries.

*Corresponding author

Naveen Edapurath Vijayan, Sr Data Engineering Manger, Amazon Seattle, WA 98765, USA.

Received: December 06, 2024; **Accepted:** December 13, 2024; **Published:** December 20, 2024

Keywords: Cybersecurity, Big Data Analytics, Real Time Threat Detection, Anomaly Detection, Machine Learning, Insider Threats, Fraud Detection, Public Sector Security

Introduction

Cybersecurity threats are evolving at an unprecedented rate, targeting sensitive data across enterprise HR systems, financial institutions, and government agencies. Cybercriminals are leveraging sophisticated attack vectors such as zero day exploits, AI-driven malware, and social engineering tactics, making it challenging for traditional security frameworks to keep pace. Existing security approaches often rely on rule-based systems that match attack signatures against predefined patterns. However, these systems fail to detect previously unknown threats, leaving organizations vulnerable to data breaches, fraud, and service disruptions.

Moreover, the exponential growth in real time security log data necessitates scalable solutions capable of ingesting, processing, and analyzing millions of log entries per second. Security teams need real time analytics to detect anomalies and respond before an attack escalates. Delayed threat detection can lead to severe financial and reputational consequences, particularly in industries dealing with sensitive customer data, government records, and proprietary corporate information.

This paper introduces a real-time cyber threat detection framework that leverages big data analytics, distributed machine learning models, and automation driven security responses to provide proactive security monitoring. Unlike traditional security approaches that rely on batch processing, this framework enables continuous streaming analytics using Apache Kafka and Spark Streaming to handle high volume security data in real time. Additionally, the proposed framework integrates AI-driven anomaly detection models, improving accuracy in detecting insider threats, financial fraud, and unauthorized government system access.

Proposed Framework

The proposed real time cyber threat detection framework is designed to handle vast amounts of security data generated by HR systems, financial transactions, and government infrastructures. It enables continuous monitoring, anomaly detection, and automated incident response, ensuring that organizations can proactively mitigate security threats in real time. The framework is built on five critical components that work together to deliver an end to end solution for identifying and neutralizing cyber threats.

These components include Secure Data Ingestion and Streaming, Preprocessing and Feature Engineering, Machine Learning-Based Threat Detection, Real Time Analytics and Visualization, and Automated Incident Response and Mitigation. Each of these components plays a fundamental role in processing, analyzing, and responding to cyber threats efficiently.

Secure Data Ingestion and Streaming

Security data is generated from multiple sources, including network traffic logs, authentication records, financial transactions, and HR system logs. This data arrives in various formats, making it critical to establish a scalable and secure data ingestion pipeline. The ingestion layer serves as the entry point for security logs, ensuring that they are collected, transmitted, and stored in a highly available, low latency environment.

Data Sources and Collection Mechanisms

The framework supports a wide range of security log sources, including:

1. **Network Traffic Logs:** These include firewall logs, VPN logs, proxy logs, and intrusion detection system (IDS) alerts. These logs provide insights into network anomalies, unauthorized access attempts, and potential intrusions.
2. **HR System Logs:** Employee authentication logs, user access records, and modification history in HR systems. These logs are essential in identifying insider threats such as unauthorized access to confidential employee data.
3. **Financial Transaction Logs:** Banking transactions, credit card purchases, wire transfers, and fraud detection logs. By monitoring financial transactions, the framework can detect anomalies in payment behaviors.
4. **Government Security Alerts:** Public sector agencies generate security logs related to biometric authentication, IP tracking, and unauthorized access attempts in government networks. These logs help identify espionage attempts and national security threats.

To ensure seamless data collection, the framework utilizes Apache Kafka, a distributed event streaming platform capable of handling millions of log events per second. Kafka acts as an intermediary between log producers (e.g., firewalls, HR systems, financial databases) and the real-time analytics engine. The ingestion pipeline is further secured using TLS encryption, ensuring that security logs cannot be tampered with during transmission.

Preprocessing and Feature Engineering

Once security logs are ingested, they need to be processed, cleaned, and structured before they can be analyzed. Raw security logs contain duplicate entries, missing fields, and unstructured text, making preprocessing an essential step in ensuring data consistency and integrity. Additionally, feature engineering extracts valuable insights from raw logs, making them suitable for machine learning analysis.

Log Normalization and Standardization

Security logs originate from different systems, each using unique data formats. The framework normalizes logs by converting them into a unified schema such as JSON or Avro. This ensures that logs from different sources can be processed consistently. Normalization includes:

- **Timestamp Synchronization:** Aligning logs from different sources based on timestamps to ensure accurate event sequencing.
- **IP Address Resolution:** Mapping raw IP addresses to geolocation data for identifying suspicious activity.
- **User Identity Linking:** Matching user login attempts across multiple systems to detect account misuse.

Feature Extraction and Anomaly Tagging

To facilitate machine learning based anomaly detection, security logs must be transformed into structured feature sets. Feature engineering involves extracting key attributes that indicate potential threats. Some of the most relevant features include:

- **Failed Login Attempts:** A higher frequency of failed logins may indicate brute-force attacks.
- **Transaction Variance:** Sudden increases in transaction amounts can be indicative of fraudulent financial activity.
- **Data Access Patterns:** Employees downloading large amounts of confidential HR data within a short period may indicate an insider threat.
- **Network Traffic Spikes:** Unusual surges in inbound or outbound data traffic could signal data exfiltration attempts.

Once these features are extracted, anomaly scores are assigned to each record, labeling potential security threats based on their deviation from normal behavior.

Machine Learning Based Threat Detection

At the core of the proposed framework is a machine learning-driven anomaly detection engine. The framework employs a hybrid approach, combining supervised and unsupervised machine learning models to detect cyber threats. This ensures that both known attack patterns and previously unseen threats can be identified with high accuracy.

Supervised Learning for Threat Classification

Supervised learning models are trained on labeled security datasets containing previously recorded cyberattacks. These models are capable of identifying known attack patterns and classifying threats based on predefined categories.

The framework uses Random Forest, Gradient Boosting, and LSTMs (Long Short-Term Memory networks) for supervised classification. These models excel at detecting insider threats, financial fraud, and malware activity. Training data is obtained from well-known cybersecurity datasets such as NSL-KDD, CICIDS, and enterprise security logs.

Unsupervised Learning for Anomaly Detection

Unsupervised learning techniques are employed to detect zero-day attacks and previously unknown threats. These models do not require labeled data; instead, they identify anomalies by learning normal behavioral patterns and flagging deviations. The framework utilizes:

- **Autoencoders:** Neural networks that learn compressed representations of normal network behavior and identify deviations.
- **Isolation Forests:** Anomaly detection models that isolate outliers from normal events.
- **k-Means Clustering:** Grouping security logs into clusters to detect unusual behavior.
- By combining supervised and unsupervised learning, the framework significantly reduces false positives while ensuring that emerging threats are detected in real time.

Real Time Analytics and Visualization

Security teams require an intuitive dashboard to monitor threats, investigate incidents, and take action in real time. The framework integrates Elasticsearch and Kibana, enabling security analysts to interact with threat intelligence data through a web based interface.

Key Dashboard Features

- **Live Threat Heatmaps:** Displays the geographic locations of detected threats.
- **Anomaly Alerts:** Generates real-time alerts for suspicious activity.
- **Drill-Down Investigation:** Allows analysts to explore specific security events in detail.

The dashboard ensures that security personnel have full visibility into ongoing cyber threats and can take immediate action to prevent security breaches.

Automated Incident Response and Mitigation

To minimize security risks, the framework incorporates automated incident response mechanisms. When a threat is detected, the system takes immediate action based on predefined security policies.

Automated Security Actions

- **Insider Threat Detection in HR Systems:** If an employee attempts to access unauthorized files, their account is immediately locked, and an alert is sent to HR administrators.
- **Financial Fraud Prevention:** If a fraudulent transaction is detected, it is placed on hold, and additional verification is requested.
- **Government Security Breaches:** If an intrusion attempt is identified, the system blocks the attacker's IP address and updates firewall rules dynamically.

The Security Orchestration, Automation, and Response (SOAR) system ensures that detected threats are neutralized in real time without requiring human intervention.

Implementation of the Real Time Cyber Threat Detection Framework

System Architecture and Deployment

The real time cyber threat detection framework is implemented using a distributed, cloud native architecture designed to handle high throughput security data streams, real time processing, machine learning inference, and automated incident response. The system is deployed on Amazon Web Services (AWS) to ensure scalability, fault tolerance, and efficient resource utilization.

Core Components and Deployment Environment

The framework consists of multiple interdependent components, each playing a critical role in detecting and mitigating cyber threats in real-time. The key components include:

- **Apache Kafka:** Serves as the message broker for ingesting security logs from various sources such as HR systems, financial transactions, and network firewalls. Kafka ensures reliable, distributed, and fault tolerant log streaming.
- **Apache Spark Streaming:** Performs real-time data transformations, feature engineering, and anomaly detection. Spark Streaming is chosen for its ability to process millions of security log entries per second with low latency.
- **TensorFlow and Scikit learn:** Machine learning models are implemented using TensorFlow for deep learning-based threat detection and Scikit learn for traditional ensemble models such as Random Forest and Isolation Forest.
- **Amazon S3 and DynamoDB:** Act as storage solutions for historical security logs and metadata. S3 stores raw logs, while DynamoDB maintains indexed information for real time queries.
- **Elasticsearch and Kibana:** Provide real-time security analytics and visualization, allowing security analysts to monitor ongoing threats via an interactive dashboard.
- **AWS Lambda:** Hosts lightweight machine learning inference models that analyze incoming data streams in real time. This serverless approach optimizes costs while maintaining high availability.
- **Security Orchestration, Automation, and Response (SOAR) System:** Integrated with AWS Lambda to trigger automated responses such as account suspension, transaction blocking, or firewall rule updates upon detecting suspicious activity.

Cloud Deployment Model

The implementation is deployed in a multiregional AWS environment to ensure disaster recovery and high availability. Kafka and Spark clusters are provisioned using Amazon EMR, while machine learning models are deployed as Sage Maker endpoints for real-time inference. To enhance security, IAM roles and KMS encryption are implemented to restrict access to sensitive security logs.

Data Ingestion and Processing Pipeline

The data ingestion pipeline is responsible for continuously collecting, formatting, and forwarding security events to the real time analytics layer. The pipeline operates in a multistage streaming architecture, ensuring that all logs are processed as soon as they arrive.

Real Time Data Sources

The framework ingests data from multiple sources, each requiring specialized processing techniques. The major data sources include:

- **HR System Logs:** Captures login attempts, file access requests, and administrative actions in HR applications.
- **Financial Transactions:** Includes banking transactions, online purchases, and wire transfers that must be monitored for fraudulent activity.
- **Network and Firewall Logs:** Aggregates IDS alerts, VPN access records, and traffic patterns to identify suspicious activity.
- **Government Security Logs:** Includes biometric authentication logs, IP address tracking, and unauthorized access attempts to classified government networks.

  Kafka serves as the central streaming hub, enabling low latency log ingestion and message queuing. Each data source has a dedicated Kafka topic partition, allowing parallel processing of different log types without causing bottlenecks.

Streaming Data Transformation and Preprocessing

Once logs are ingested, they undergo real time transformations in Apache Spark Streaming to extract meaningful security features. This includes:

- **Timestamp Alignment:** Synchronizing log entries across multiple systems to create an accurate event timeline.
- **Log Normalization:** Standardizing data into a structured format such as JSON or Avro.
- **Noise Reduction:** Removing redundant log entries and filtering out low risk activity to reduce processing overhead.
- **Feature Engineering:** Extracting security-relevant attributes such as login frequency, transaction amounts, data transfer rates, and failed authentication attempts.

  The preprocessed logs are temporarily stored in Amazon S3 before being forwarded to the machine learning layer for anomaly detection.

Machine Learning Model Training and Deployment

The machine learning based anomaly detection system is the core of the framework, responsible for identifying insider threats, fraudulent transactions, and government network intrusions. The hybrid machine learning approach consists of both supervised and unsupervised learning models deployed in a distributed environment.

Model Training and Optimization

Supervised Learning Models:

- **Random Forest:** Used for financial fraud detection, trained on labeled transaction datasets.

- **Gradient Boosting Machines (XGBoost):** Trained to classify high-risk HR system activity.
- **LSTM Neural Networks:** Detects sequence based insider threat behaviours, such as repeated unauthorized data access.

Unsupervised Learning Models

- **Autoencoders:** Identify zero day threats by learning normal behavior patterns and flagging deviations.
- **Isolation Forests:** Detect outliers in security logs, used for identifying unknown fraud attempts.
  All models are trained using Amazon Sage Maker, ensuring GPU acceleration for deep learning models and distributed training for large datasets.

Model Deployment and Real-Time Inference

Once trained, the models are deployed using AWS Lambda and Sage Maker endpoints for real time inference. Security logs are streamed into the models, and predictions are made within milliseconds, ensuring rapid detection of cyber threats. The model inference pipeline follows these steps:

- **Kafka Streams Forward Logs to ML Models:** Real time security logs are forwarded to the deployed machine learning models via Kafka topics.
- **Feature Vectorization:** Incoming logs are transformed into feature vectors before being analyzed by the models.
- **Threat Score Computation:** Each event is assigned a risk score, indicating the likelihood of a security breach.
- **Decision Making:** High-risk threats are immediately escalated, triggering automated incident response actions.

Real Time Analytics and Security Dashboard

The framework integrates Elasticsearch and Kibana to provide a real-time security analytics dashboard. Security analysts can visualize threat intelligence and investigate suspicious activity as it happens.

Dashboard Features

- **Live Threat Heatmaps:** Visual representation of attack origins.
- **Anomaly Alerts:** Flagging of high risk transactions and network intrusions.
- **Drill-Down Investigation:** Detailed log inspection for forensic analysis.
  The dashboard is designed for real time querying, allowing security teams to respond to threats without delays.

Automated Incident Response and Security Actions

To mitigate security risks, the framework includes a rule-based and AI-driven automated incident response mechanism. When an attack is detected, the system automatically executes predefined security policies to contain and neutralize threats.

Response Automation Scenarios

- **HR Insider Threats:** If an employee accesses unauthorized HR files, their account is locked, and HR administrators are alerted.
- **Financial Fraud Detection:** If a fraudulent transaction is detected, it is automatically placed on hold for manual verification.
- **Government Cyber Intrusions:** If unauthorized access to a government network is detected, the source IP is blocked, and firewall rules are updated dynamically.
  The Security Orchestration, Automation, and Response (SOAR) system ensures fast, automated incident mitigation, reducing the workload on security teams while preventing attack escalation.

Performance Evaluation and Benchmarks

The framework was tested using three datasets corresponding to HR security logs, financial transactions, and government network access logs. The evaluation focused on detection accuracy, processing speed, and system scalability.

Performance Metrics

Metric	HR In-sider Threats	Financial Fraud	Government Cyber In-trusions
Precision	98.6%	94.2%	96.5%
Recall	97.8%	93.4%	95.1%
Detection Latency	1.8 sec	1.4 sec	1.9 sec
Throughput	50,000 logs/sec	45,000 logs/sec	48,000 logs/sec

The system demonstrated high precision and recall, achieving real time performance with low latency and high throughput.

Conclusion

The proposed real-time cyber threat detection framework effectively addresses the growing challenges of cybersecurity across HR, financial, and government sectors. By integrating big data analytics, distributed machine learning models, and automated security response mechanisms, the system provides a scalable and efficient solution for detecting and mitigating threats in real time. The secure data ingestion layer, powered by Apache Kafka, ensures high-throughput log collection from diverse sources, while Apache Spark Streaming enables low latency data transformation and feature extraction. The hybrid machine learning-based anomaly detection engine combines supervised and unsupervised learning techniques to accurately classify threats while reducing false positives. Furthermore, the real time analytics dashboard, built using Elasticsearch and Kibana, provides an intuitive interface for security teams to monitor threats, analyze incidents, and respond proactively.

Experimental evaluations demonstrate that the framework is highly accurate, with 98.6% precision in HR insider threat detection, 94.2% accuracy in financial fraud detection, and a sub-2-second detection latency for government security threats. The system is capable of processing over 50,000 security events per second, proving its scalability in high-volume environments. By integrating automated incident response mechanisms, the framework ensures that security threats are mitigated without manual intervention, reducing the mean time to respond (MTTR) and minimizing the potential impact of cyberattacks [1-17].

Future enhancements to the framework will focus on adaptive AI-driven threat detection, leveraging reinforcement learning and generative AI to continuously evolve its understanding of cyber threats. Additionally, blockchain based security logging will be explored to enhance data integrity and forensic analysis capabilities. As organizations face increasingly complex cyber threats, the adoption of real time, AI-driven cybersecurity frameworks will be essential in safeguarding sensitive information and critical infrastructure.

References

1. S Zuech, T M Khoshgoftaar, R Wald (2015) "Intrusion detection and big heterogeneous data: a survey," Journal of Big Data 2: 1-41.
2. M H Bhuyan, D K Bhattacharyya, J K Kalita (2014) "Network anomaly detection: Methods, systems, and tools," IEEE Communications Surveys & Tutorials 16: 303-336.
3. J D Ullrich (2020) "Artificial intelligence for cyber threat detection: Current trends and future outlook," Cybersecurity Journal 8: 45-59.
4. U Kumar, P Mohapatra, S Ram (2020) "Real-time intrusion detection and prevention system in big data," Journal of Big Data 7: 1-20.
5. J Han, M Kamber, J Pei (2011) Data Mining: Concepts and Techniques, Morgan Kaufmann.
6. Javaid, Q Niyaz, W Sun, M Alam (2016) "A deep learning approach for network intrusion detection system," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies 21-26.
7. S Bhatia, M Alam, R Singh (2020) "Big data-driven cyber threat intelligence," Future Generation Computer Systems 108: 567-580.
8. Mukherjee, L T Heberlein, K N Levitt (1994) "Network intrusion detection," IEEE Network 8: 26-41.
9. P Malhotra, L Vig, G Shroff, P Agarwal (2015) "Long short-term memory networks for anomaly detection in time series," Proceedings of the 23rd European Symposium on Artificial Neural Networks 89-94.
10. M Tavallaee, E Bagheri, W Lu, A A Ghorbani (2009) "A detailed analysis of the KDD CUP 99 data set," Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications 53-58.
11. K Scarfone, P Mell (2007) "Guide to intrusion detection and prevention systems (IDPS)," National Institute of Standards and Technology (NIST), Special Publication 800-894.
12. J R Quinlan (1993) C4.5: Programs for Machine Learning, Morgan Kaufmann.
13. Krügel, T Toth, E Kirda (2002) "Service-specific anomaly detection for network intrusion detection," Proceedings of the 2002 ACM Symposium on Applied Computing 201-208.
14. T Chen, C Guestrin (2016) "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 785-794.
15. McAfee, G Brynjolfsson (2012) "Big data: The management revolution," Harvard Business Review 90: 60-68.
16. G M Weiss, F Provost (2003) "The effect of class distribution on classifier learning: An empirical study," Journal of Artificial Intelligence Research 18: 247-286.
17. M Ring, S Wunderlich, D Landes, A Hotho (2019) "A survey of network-based intrusion detection data sets," Computers & Security 86: 47-167.