IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15573

MCP-Based Context-Aware System Monitoring and Threat Detection Agent

G. Priyadharshini, M.E.[1], Balaji A[2], Vishnu S[3], Mohamed Noufal M[4]

Assistant Professor, Department of Computer Science and Engineering (Cyber Security),

Dhanalakshmi Srinivasan College of Engineering and Technology, India[1]

Student, Department of Computer Science and Engineering (Cyber Security),

Dhanalakshmi Srinivasan College of Engineering and Technology, India[2-4]

Abstract: This paper presents an MCP-Based Context-Aware System Monitoring and Threat Detection Agent, an intelligent, real-time cybersecurity monitoring platform leveraging the Model Context Protocol (MCP) to deliver context-aware threat detection and automated response capabilities. Traditional Security Information and Event Management (SIEM) systems rely on static rule-based engines that produce false-positive rates of 25–45%, suffer from alert fatigue, and fail to detect multi-stage Advanced Persistent Threats (APTs). The proposed system integrates a FastAPI backend, PostgreSQL 16 storage, WireGuard VPN encryption, and a Bootstrap 5 web dashboard to provide unified, real-time visibility across network traffic, system logs, and behavioral metrics. The MCP AI agent maintains a rolling context window over incoming security events, enabling temporal correlation, multi-stage attack detection, lateral movement identification, and significant reduction of false positives through composite threat scoring. Validation results demonstrate a 67% reduction in false positives, sub-3-second automated mitigation response, throughput exceeding 6,200 concurrent events per second, an AUC of 0.94 on the ROC curve, and 60–75% reduction in operational costs versus commercial SIEM solutions. All three functional modules—Data Collection & Traffic Monitoring, Threat Analysis & Context Awareness, and Alerting & Secure Notification—have been implemented and validated in a prototype environment over a 30-day test period.

Keywords: Model Context Protocol (MCP), Cybersecurity, Real-Time Threat Detection, Context-Aware AI, SIEM, FastAPI, WireGuard VPN, Anomaly Detection, Automated Incident Response.

I. INTRODUCTION

Modern IT infrastructures generate enormous volumes of security events every second. A typical enterprise environment produces tens of millions of log entries daily across servers, network devices, endpoints, and cloud services. Traditional security monitoring systems rely on static rule-based engines that lack contextual awareness, resulting in false-positive rates of 25–45%, alert fatigue among analysts, and dangerously delayed threat response [1].

Sophisticated multi-stage attacks such as Advanced Persistent Threats (APTs) unfold over days or weeks, with individual steps—reconnaissance, initial compromise, lateral movement, privilege escalation, data exfiltration—each appearing benign in isolation. Only by maintaining contextual memory of events across time can these attack sequences be recognized and interdicted [2], [3].

Industry data underscores the severity: the average Mean Time To Detect (MTTD) a breach is 197 days, and Mean Time To Respond (MTTR) adds another 69 days. According to IBM's Cost of a Data Breach Report 2024, the average breach cost reached USD 4.88 million—a 10% year-on-year increase. Commercial SIEM solutions addressing these gaps cost USD 50,000–500,000 annually in licensing alone [4].

The Model Context Protocol (MCP), introduced by Anthropic, provides a standardized interface for AI agents to interact with external data sources and tools while maintaining evolving contextual memory. In cybersecurity monitoring, MCP enables the AI agent to maintain a coherent, time-evolving understanding of the security state of the monitored environment—something stateless rule engines cannot achieve. By preserving contextual memory across thousands of security events, the MCP agent identifies complex multi-stage attack patterns, detects lateral movement, and distinguishes genuine threats from benign anomalies with far greater accuracy [5].

This paper proposes and validates an MCP-based threat detection system integrating FastAPI, PostgreSQL, WireGuard VPN, and a Bootstrap 5 dashboard. The remainder of the paper is organized as follows: Section II reviews related

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15573

literature; Section III presents the problem description; Section IV describes the system architecture and design; Section V details the methodology; Section VI presents experimental results; and Section VII concludes with future work directions.

II. LITERATURE REVIEW

He et al. [1] proposed a Double-Layer Detection framework for internal threats in enterprise systems using deep learning, combining behavioral profiling with anomaly scoring via LSTM-based sequence modeling, achieving high detection accuracy while minimizing false positives. This work directly informs the composite threat scoring formula adopted in the proposed MCP agent.

Haile et al. [2] presented a real-time automated classification framework combining Cyber Threat Intelligence (CTI) with Latent Dirichlet Allocation (LDA) and NLP to detect zero-day threats not covered by static signatures. The CTI integration layer in the proposed system is directly influenced by this work.

Kim et al. [3] introduced a Cyber Threat Inference system using chain-rule and correlation analysis on threat class sequences, enabling proactive defense by predicting imminent attack stages. The sequence-aware context multiplier C(W) in the proposed scoring formula is inspired by this approach.

Crothers et al. [6] surveyed threat models and detection methods for machine-generated content, categorizing generation methods and detection strategies. The proposed system incorporates NLP-based content analysis in the MCP agent's payload inspection module for detecting AI-generated phishing artifacts.

Wang et al. [7] demonstrated AI-based real-time threat detection across heterogeneous smart city IoT infrastructure, validating asynchronous event processing architectures as the key enabler of high-throughput monitoring. The proposed system adopts FastAPI's async event loop model from this work.

He et al. [8] proposed methods for detecting unknown threats in smart contracts using semantic analysis and taint tracking. Their unsupervised anomaly detection approach—comparing behavioral fingerprints against a known-good baseline rather than a catalogue of known threats—is adopted in the anomaly scorer.

Alzaabi and Mehmood [9] reviewed 12 years of ML-based insider threat detection, finding that hybrid models combining supervised and unsupervised components consistently outperform single-approach systems, and that temporal and behavioral feature engineering has greater impact than model architecture alone.

Liu et al. [10] demonstrated federated learning for cybersecurity threat intelligence sharing, enabling collaborative model improvement without sharing sensitive event data. This work informs the federated learning extension identified in the future work section.

Hassan et al. [11] introduced WATSON, a graph-based system for reconstructing attack provenance from system audit logs using dependency graph analysis. The graph-based kill chain visualization in future work is inspired by this research, with the current relational PostgreSQL schema designed to support a future Neo4j graph layer.

Existing literature demonstrates substantial progress in individual aspects of threat detection, but no existing solution provides a unified, context-aware, open-source platform combining MCP-based temporal correlation, automated response, and enterprise-grade scalability with minimal deployment cost.

III. PROBLEM DESCRIPTION

A. Existing System

The cybersecurity monitoring landscape is dominated by traditional SIEM solutions and rule-based IDS/IPS systems. These collect logs and network traffic and apply predefined rules to generate alerts. While effective for known attack signatures, they are fundamentally limited in contextual awareness and cross-source correlation. In centralized monitoring setups, the high volume of alerts—often tens of thousands per day—results in alert fatigue. Industry studies report that analysts spend 50–70% of their time investigating false positives. Key disadvantages include: high false-positive rates (25–45%); stateless single-event analysis; inability to detect zero-day attacks; slow manual remediation extending MTTD to ~197 days; prohibitive licensing costs of USD 50K–500K+ per year; and centralized architectures creating single points of failure.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15573

B. Proposed System

The proposed MCP-Based Context-Aware Threat Detection Agent addresses each limitation through an intelligent, distributed, and automated platform. The system replaces static rule-based detection with a dynamic MCP AI agent that understands event context, correlates data across sources, and makes intelligent threat assessments in real time. Key advantages include: 67% reduction in false positives; sub-3-second automated mitigation; 6,200+ events/sec throughput; WireGuard encryption at near-zero overhead; unified dashboard visibility; and 62% reduction in total operational cost through open-source technology.

IV. SYSTEM ARCHITECTURE AND DESIGN

The overall architecture follows a layered, distributed design with five primary layers: Data Collection, Processing and Analysis, MCP AI Agent, Storage, and Presentation. Each layer communicates with adjacent layers through well-defined REST APIs, ensuring modularity and independent scalability. All inter-layer communication traverses WireGuard-encrypted channels where components span network boundaries.

A. MCP Protocol — Technical Background

The Model Context Protocol (MCP) is an open standard defining how AI language models interact with external data sources, tools, and environments via a JSON-RPC 2.0 interface over stdio, HTTP/SSE, or WebSocket transports. The MCP server in this system exposes cybersecurity-specific tools—event query, threat signature lookup, behavioral baseline retrieval, and response execution—to the AI agent client. Key properties exploited include: stateful context window (temporal correlation across tool invocations); tool composition (chaining queries and responses in a single reasoning pass); structured output (machine-parseable threat assessments directly persisted to PostgreSQL); and transport agnosticism (HTTP/SSE deployment alongside FastAPI).

B. Module Architecture

Module 1 — Data Collection & Traffic Monitoring: Employs Scapy-based packet sniffing, inotify-based log watching, and psutil metric collection deployed as lightweight background agents on monitored hosts. Raw data is serialized into canonical JSON event schema and forwarded via WireGuard tunnel to the FastAPI ingestion endpoint. Supports configurable BPF capture filters, rate limiting, and event deduplication.

Module 2 — Threat Analysis & Context Awareness: The MCP AI agent maintains per-entity rolling context windows (default: 500 events or 10 minutes). Analysis proceeds in two phases: fast rule-based signature lookup against 150+ known patterns, then ML-based anomaly scoring for non-matching events. The composite threat score is: $S = (\alpha \cdot S\_sig + \beta \cdot S\_anom) \times C(W)$, where $C(W)$ is a sequence-aware context multiplier that amplifies scores for events extending known attack progressions. Scores above configurable threshold θ trigger threat escalation.

Module 3 — Alerting & Secure Notification: Receives confirmed threat records and executes automated responses (IP block via iptables, process termination via SIGKILL, host isolation via route table manipulation). Dispatches multi-channel notifications: WebSocket push to dashboard, SMTP email, and configurable webhook (Slack, PagerDuty, JIRA). All actions are logged to the PostgreSQL AuditLog table.

C. Data Flow and Database Design

The Level 0 Context Diagram shows four external entities: System Administrator (configures policies), Monitored Hosts (generate data streams), Security Analyst (receives alerts), and MCP Threat Intelligence Feed (provides current threat indicators). The Level 1 DFD decomposes the system into seven sub-processes: P1 Data Ingestion, P2 Event Normalization, P3 Context-Aware Analysis (MCP), P4 Threat Scoring, P5 Alert Generation, P6 Automated Response, and P7 Dashboard Update. PostgreSQL 16 serves as the central data repository with five core tables: Events, ThreatRecords, Alerts, Responses, and AuditLog.

V. METHODOLOGY

A. MCP Agent Algorithm

Algorithm 1: MCP Threat Detection Agent
Input: Event stream $E = \{e_1, e_2, \ldots, e_n\}$
Output: Threat records T with automated responses R

1. Initialize context window $W \leftarrow \emptyset$ per monitored entity
2. For each incoming event e_i:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15573

3. Normalize e_i → canonical schema (source, type, payload, timestamp)
4. Append e_i to entity context window W; evict if |W| > 500 or age > 10 min
5. S_sig ← SignatureLookup(e_i) // fast rule-based match
6. If S_sig = 0: S_anom ← AnomalyScore(e_i, W) // ML baseline deviation
7. C(W) ← ContextMultiplier(W) // amplify if sequence matches APT pattern
8. S_composite ← (α·S_sig + β·S_anom) × C(W)
9. If S_composite > θ:
10. Create ThreatRecord(e_i, S_composite, W)
11. Execute AutoResponse(ThreatRecord) // iptables/SIGKILL/route
12. Dispatch MultiChannelAlert(ThreatRecord)
13. Log to AuditLog(ThreatRecord, response, timestamp)
14. End if
15. End for

B. Threat Scoring Formula

The composite threat score $S = (\alpha \cdot S\_sig + \beta \cdot S\_anom) \times C(W)$ combines rule-based signature confidence $S\_sig \in [0,1]$ and ML-derived anomaly score $S\_anom \in [0,1]$ weighted by $\alpha = 0.6$ and $\beta = 0.4$ respectively. The context multiplier $C(W) \in [1.0, 2.5]$ is computed from the current entity's event sequence: $C(W) = 1 + \Sigma_i P(class_i \mid class_{i-1})$ over the observed threat class chain. Events matching known APT kill-chain progressions receive multipliers approaching 2.5, ensuring multi-stage attacks generate high composite scores even when individual events appear benign.

C. Deployment Configuration

The system is containerized using Docker Compose with four services: fastapi-backend (Python 3.11, uvicorn), mcp-agent (MCP server + AI agent process), postgres-db (PostgreSQL 16 with persistent volume), and dashboard (Nginx serving Bootstrap 5 static assets with WebSocket proxy). WireGuard is deployed as a kernel module on each monitored host, with the server-side endpoint hosted alongside the FastAPI backend. All inter-service communication uses TLS 1.3 internally; WireGuard provides the transport-layer encryption for agent-to-server monitoring traffic.

VI. RESULTS AND VALIDATION

A. Functional Test Results

All functional test cases were executed against the deployed prototype over a 30-day test period. Event ingestion, MCP context window management, signature lookup, anomaly scoring, composite scoring, alert dispatch, automated response execution, and audit logging all passed with 100% functional correctness. The system correctly identified all 47 injected test attack scenarios, including 12 multi-stage APT simulations.

B. Performance Metrics

Table I presents the achieved performance metrics against defined targets. The system achieved 6,247 events/sec throughput (target: 6,000+), 2.3-second automated mitigation latency (target: <3 sec), and 67% false-positive reduction (target: ≥60%). The ROC AUC of 0.94 confirms strong classifier discrimination between genuine threats and benign anomalies.

TABLE I. Performance Metrics – Targets vs. Achieved

Metric	Target	Achieved
Event throughput	≥ 6,000 events/sec	6,247 events/sec
Automated mitigation latency	< 3 seconds	2.3 seconds
False-positive reduction	≥ 60%	67%
ROC AUC	≥ 0.90	0.94
MCP inference latency	< 2 seconds	1.7 seconds
Operational cost reduction	≥ 60%	62–75%
WireGuard CPU overhead	< 5%	< 3%

C. Stress Testing

Stress testing used Locust to inject concurrent event streams from 1 to 10,000 events/sec in 1,000-event increments. The system maintained sub-50ms queue latency up to 6,200 events/sec. Beyond this threshold, latency degraded gracefully to 180ms at 8,000 events/sec and 410ms at 10,000 events/sec, with no data loss at any tested load level. Horizontal scaling by adding a second FastAPI instance extended the throughput ceiling to 11,800 events/sec.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15573

D. Penetration Testing

Penetration testing using OWASP ZAP and manual SQL injection probing found zero critical or high-severity vulnerabilities. The parameterized query implementation successfully blocked all 23 SQL injection attempts. WireGuard key rotation every 24 hours prevented replay attack scenarios. Role-based access control correctly denied all 15 privilege escalation test attempts.

E. Comparison with Existing Systems

Table II compares the proposed system against representative commercial and open-source SIEM alternatives across key dimensions. The proposed system achieves superior false-positive reduction (67% vs. 15–30% for rule-based SIEM) and context-aware detection while eliminating annual licensing costs of USD 50,000–500,000.

TABLE II. System Comparison – Proposed vs. Existing

Feature	Rule-Based SIEM	ML-Based IDS	Proposed System
Context-aware detection	No	Partial	Yes (MCP rolling window)
False-positive reduction	15–25%	30–45%	67%
Automated response	No	No	Yes (<3 sec)
Multi-stage APT detection	No	Partial	Yes
Annual licensing cost	USD 50K–500K	USD 10K–100K	USD 0 (open-source)
Throughput	~3,000 ev/sec	~4,000 ev/sec	6,247 ev/sec
Encrypted monitoring traffic	Optional	Rarely	Yes (WireGuard)

VII. CONCLUSION AND FUTURE WORK

This paper presented an MCP-Based Context-Aware System Monitoring and Threat Detection Agent that fundamentally advances the state of the art in intelligent cybersecurity monitoring. By embedding the Model Context Protocol into the monitoring pipeline, the proposed system achieves context-aware detection accuracy and automated response speed that far exceed traditional rule-based SIEM approaches, while eliminating prohibitive commercial licensing costs.

The system's key contribution is the integration of MCP's stateful context window with composite threat scoring, enabling recognition of multi-stage APT campaigns invisible to stateless classifiers. Experimental validation over a 30-day prototype deployment demonstrates 67% false-positive reduction, sub-3-second automated mitigation, 6,247 events/sec throughput, and 0.94 ROC AUC—all meeting or exceeding defined targets. The open-source technology stack (FastAPI, PostgreSQL, WireGuard) delivers enterprise-grade security at 60–75% lower total cost of ownership than commercial alternatives.

Future work will extend the system in five directions: (1) federated learning for privacy-preserving cross-organization threat model sharing; (2) graph-based kill-chain visualization using Neo4j for attack provenance reconstruction; (3) cloud-native workload monitoring integrating eBPF-based kernel-level tracing; (4) IoT device support via lightweight MQTT-based agents; and (5) LLM-generated phishing detection at the payload inspection layer.

REFERENCES

[1]. Z. He et al., "Double-Layer Detection Framework for Internal Threats Using Deep Learning," IEEE Transactions on Information Forensics and Security, 2024.
[2]. Y. Haile et al., "Real-Time Automated Cyber Threat Classification Using CTI, LDA, and NLP," IEEE Open Journal of the Computer Society, 2025.
[3]. J. Kim et al., "Cyber Threat Inference Using Threat Class Sequences and Chain-Rule Correlation," IEEE Access, 2025.
[4]. IBM Security, "Cost of a Data Breach Report 2024," IBM Corporation, 2024.
[5]. Anthropic, "Model Context Protocol: Specification and Developer Guide," Anthropic, 2024. [Online]. Available: https://modelcontextprotocol.io
[6]. E. Crothers et al., "Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods," IEEE Access, 2023.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15573

[7]. X. Wang et al., "AI-Based Real-Time Cyber Threat Detection for Smart City IoT Infrastructure," IEEE Transactions on Consumer Electronics, 2025.

[8]. Z. He, Y. Ding, C. Chan, and M. Guizani, "Unknown Threat Detection in Smart Contracts Using Semantic Analysis and Taint Tracking," IEEE Internet of Things Journal, 2024.

[9]. M. Alzaabi and A. Mehmood, "A Survey of Machine Learning-Based Insider Threat Detection: Twelve Years of Research," IEEE Access, 2024.

[10]. Y. Liu et al., "Federated Learning for Cybersecurity Threat Intelligence Sharing," IEEE Transactions on Network Science and Engineering, 2023.

[11]. W. Hassan et al., "WATSON: Abstracting Behaviors from Audit Logs via Provenance Graphs," Proc. NDSS Symposium, 2020.

[12]. S. Vaarandi et al., "A Systematic Literature Review of Cyber Security Monitoring in Maritime," IEEE Access, vol. 13, pp. 85307–85329, 2025.

[13]. F. Wei et al., "Security Risk Assessment System for Power Monitoring Based on Zero Trust Architecture," Proc. 4th Int. Conf. CEI, 2024.

