IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Autonomous AI Agent Framework for Cyber Threat Intelligence

[1] Jayaprakash Y M, Lecturer, Department of Computer Science and Engineering, Government Polytechnic, Nagamangala, Mandya.

ABSTRACT: The increasing complexity, speed, and volume of modern cyber attacks such as APTs, ransomware, and zero-day exploits have made traditional, human-dependent Cyber Threat Intelligence (CTI) systems insufficient for real-time detection and response. To address these challenges, this project proposes an Autonomous AI Agent Framework for Cyber Threat Intelligence that leverages Machine Learning, Natural Language Processing, agentic reasoning, and tools such as Python, Fas tAPI, Wireshark/Pyshark, Elastics earch, Kibana, Lang Chain, and the Open AI GPT-4 API. The system autonomously collects, analyzes, correlates, and interprets network traffic and threat intelligence feeds to identify anomalies, extract Indicators of Compromise, classify threats, and generate actionable reports.

KEYWORDS: Cyber Threat Intelligence, Autonomous AI Agent, Threat Detection, Anomaly Detection, Agentic AI, Real-Time Monitoring.

I. INTRODUCTION

Cyber threats today are increasing in complexity, frequency, and speed, making traditional, manual cybersecurity methods insufficient for timely detection and response. Attacks such as ransomware, zero-day exploits, and Advanced Persistent Threats(APTs) exploit system vulnerabilities faster than human analyst scan react, while the massive volume of threat data makes manual analysis slow and error-prone. Cyber Threat Intelligence (CTI) helps organizations understand attacker behavior and anticipate threats, but conventional CTI systems struggle to operate at real-time scale. To overcome these challenges, Artificial Intelligence—especiallyautonomousAIagents—offersapowerfulsolutionby enabling automated data collection, real-time threat analysis, adaptive learning, and intelligent decision-making. Cyber Threat Intelligence (CTI) plays an essential role in identifying attacker behaviors, understanding emerging threats, and strengthening security decision-making, but manual CTI workflows are limited by human fatigue, slow analysis, and the inability to scale with modern data demands. To address these challenges, Artificial Intelligence—especially autonomous, agent-driven AI models—offers transformative capabilities by enabling proactive threat detection, intelligent pattern recognition, and automated incident reporting

II. LITERATURESURVEY

Existing research shows that AI significantly improves threat detection accuracy, automates analysis, and enhancesreal-time response capabilities compared to traditional manual methods. These studies form the foundation for developing an autonomous CTI framework that is more scalable, adaptive, and efficient in handling evolving cyber threats.

- Maasaoui, Z., Bekri, A., Merzouki, M., Battou, A., & Abane, A. (2024) proposed a scalable network-security monitoring framework that integrates ELK Stack with traffic monitoring tools. Their system applied machine learning on network flow data to classify traffic as benign or malicious, achieving high detection accuracy while maintaining real-time visualization dashboards.
- Robbani,F.D.,Haryatmi,E.,Riyadi,T.A.,Supono,R.A.,Kurniawan,A.B.,&Rosdiana.(2025)implementedaSnort-basedintrusiondetectionsystemwithElasticsearchandKibanavisualization.Theirsolutiondetectedvarious

attacks like SSH brute-force and ping floods while providing real-time dashboards, demonstrating effective network monitoring and alert visualization.

- Likitha, R., Tarun, N., Pallavi, N., & Vidhey, V. G. (2024) integrated Suricata IDS with ELK-based SIEM for improved threat detection. Using pattern-recognition algorithms, their system aggregated and analyzed IDS logs to reduce false positives and increase detection accuracy for both signature-based and anomalous attacks.
- Rajalakshmi, R., Akash, K., Keshavan, T., Rohit, K., & Vinoth Kumar, M. (2025) designed a Security Onion stack using Suricata, Zeek, and Wireshark integrated with ELK. Their platform supported packet capture, real-time intrusion detection, and log visualization, enabling comprehensive forensic and SOC operations.
- Regi, S., & Gurpreet, K. (2024) demonstrated the use of ELK Stack as an open-source SIEM for SMEs. Their system collected and analyzed multi-source logs in real time, providing cost-efficient, scalable monitoring and enhanced threat detection capabilities.
- Davies, T., Eiza, M. H., Shone, N., & Lyon, R. (2025) proposed a collaborative IDS architecture with multiple Snort nodes feeding a centralized SIEM. Aggregation and correlation of alerts enabled the detection of distributed attackswith higher accuracy and reduced false positives compared to single-node IDS deployments.
- Farhan, B. I., & Jasim, A. D. (2024) compared Snort and Suricata IDS for network security monitoring. Their study evaluated detection rate, latency, and resource utilization, providing guidance for selecting suitable IDS depending on network environments and threat models.
- Rahmawati, T., Karna, N., Shin, S. Y., & Putra, M. A. P. (2025) implemented an Intrusion Prevention System (IPS) integrated with ELK for real-time web attack detection. Their model effectively detected threats and provided alert dashboards while operating efficiently in resource-limited setups.
- (Journal of Information Systems Engineering and Management, 2025) presented a network security infrastructure combining Suricata, SysLog server, Elasticsearch, and Kibana. Their system allowed real-time monitoring, packet inspection, and centralized visualization, suitable for enterprise-scale security operations.

III. PROPOSED SYSTEM

The proposed system introduces an autonomous AI agent framework for cyber threat intelligence (CTI) that integrates real-time network monitoring, threat detection, and visualization. Network traffic is captured using Wireshark/PyShark, while system logs are ingested through Logstash, creating a consolidated dataset for analysis. Collected data is pre-processed using Python, which filters noise and extracts relevant features for detection.The processed logs and network traffic are then stored, indexed, and visualized using Elastics earch and Kibana, enabling real-time dashboards for monitoring malicious activity and analyzing network events. The system provides an automated, interactive platform for security monitoring, allowing users to view detailed visualizations, detect anomalies, and track potential threats efficiently. Its modular architecture ensures scalability and ease of future enhancement, while the implemented components already deliver an effective foundation for real-time network monitoring and threat visualization.

IV. METHODOLOGY

A.System Architecture

The proposed system is meticulously designed to automate and streamline the end-to-end process of legal document analysis and reasoning, focusing on adaptability, modularity, and transparency. The architecture employs modern AI techniques, multi-agent or chest ration, and dynamic model selection to simulate intelligent legal analysis. The workflow is distributed across three tightly integrated layers:

1. Data Collection Layer

The first stage of the methodology is Data Collection, where the system gathers all essential raw information required for threat intelligence processing. This includes capturing real-time network packets using tools like Wireshark or Pyshark, collecting system logs from servers, firewalls, routers, and other network devices, and integrating external threat intelligence feeds that provide Indicators of Compromise (IoCs) such as malicious IP addresses, domains, file hashes, and malware signatures. By combining internal and external data sources, the system builds a large, diverse dataset that reflects real-world cyber activities, making it capable of detecting both common and advanced threats.This step is crucial because the accuracy and effectiveness of the entire CTI pipeline depend on the quality and richness of the data collected at the beginning.

2. **Data Preprocessing**

Once the data is collected, it undergoes a detailed **Data Preprocessing** phase to ensure it is clean, relevant, and ready for analysis. Raw cyber security data is often noisy, unstructured, and inconsistent, so preprocessing involves removing irrelevant entries, filtering out duplicate logs or packets, correcting errors, handling missing values, and converting raw traffic into structured formats like JSON or CSV. Python-based scripts extract important features such as source and destination IPs, timestamps, ports, protocols, request types, and abnormal behaviors. This transformation not only standardizes the dataset but also enhances its quality, enabling AI models to detect hidden patterns more accurately. Effective preprocessing significantly boosts the performance of the threat detection system. A dynamic routing mechanism assigns tasks to the most suitable model based on task complexity, latency tolerance, and model confidence.

3. **Model Integration and Agent Design**

After preparing the data, the next major step is **Model Integration and Agent Design**. Here, the cleaned data is connected with AI models that perform the intelligent analysis. Tools like the Open AI GPT API, Lang Chain, and Fast API are integrated to build an autonomous CTI agent capable of reasoning, decision-making, and multi-step analysis. This stage defines the logical behavior of the agent—how it interprets input data, how it classifies threat types, how it correlates events across logs, and what actions it should take when suspicious activity is detected. By designing an agent with agentic reasoning capabilities, the system becomes capable of interacting with data dynamically, generating contextual explanations, identifying patterns, and working independently without constant human involvement. This integration forms the brain of the proposed system.

4. **Threat Detection and Analysis**

The fourth stage, **Threat Detection and Analysis**, is where the system actively evaluates processed data to identify cyber threats in real time. AI models examine patterns, anomalies, and deviations from normal behaviour to detect Indicators of Compromise such as unusual login attempts, unexpected traffic spikes, malicious payloads, or suspicious communication with known blacklisted IPs. NLP-based analysis helps interpret logs semantically, allowing the system to detect hidden or subtle signs of attacks. The threat analysis engine categorizes the threat, determines its severity, correlates it with known attack techniques, and provides early alerts. This stage transforms raw network and log data into meaningful intelligence that can prevent or reduce the impact of cyber attacks.

5. **System feedback and Learning**

In **System Feedback and Learning**, the CTI agent improves itself continuously through feedback loops. When the system raises alerts, analysts may mark them as true positives, false positives, or low-risk anomalies. This feedback is fed back into the AI models, helping them refine their understanding of what constitutes a real threat. The system also learns from historical threat data and adapts to emerging threats using updated intelligence feeds. Over time, this self-learning capability reduces false alarms, increases prediction accuracy, and enhances the agent’s ability to detect new or evolving attack techniques. This makes the system more robust, reliable, and adaptive to future cyber landscapes.

6. **Visualization and Reporting**

The final stage is **Visualization and Reporting**, where all processed and analyzed results are presented in a clear, visual, and actionable manner. Tools like Elastics earch and Kibana generate dashboards that show real-time metrics, threat trends, attack timelines, log patterns, risk scores, and network activity graphs. These visual representations help analysts quickly understand the security status of the system. Additionally, the AI agent automatically generates detailed threat intelligence reports that summarize detected threats, explain their impact, list IoCs, and provide recommendations for mitigation. This ensures that both technical experts and non-technical stakeholders can interpret the findings easily and respond effectively.

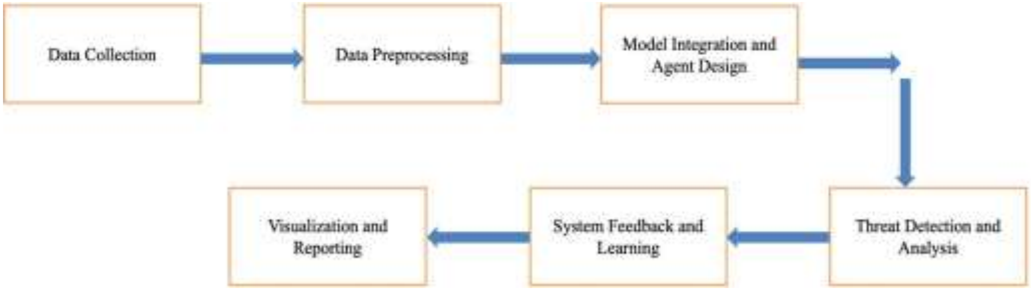

Data Collection
Data Preprocessing
Model Integration and Agent Design
Threat Detection and Analysis
System Feedback and Learning
Visualization and Reporting

Fig.1MethodologyforAutonomousAI-BasedCyberThreatIntelligence

V. RESULTS

The results of the system demonstrate that the Autonomous AI Agent Framework effectively monitors network traffic, analyzes logs in real time, and identifies potential threats with improved accuracy and reduced manual intervention. The dashboards generated through Kibana clearly visualize search rates, indexing performance, latency metrics, and recent log activities, allowing easy interpretation of system behavior. The Elastic Cloud deployment interface confirms stable operation of all backend services, while the container monitoring dashboard shows efficient resource usage and proper functioning of Logstash, Elastic search, and Kibana services. Overall, the results validate that the integrated AI-driven CTI system successfully automates threat detection, supports real-time monitoring, and provides actionable insights through user-friendly visual interfaces.

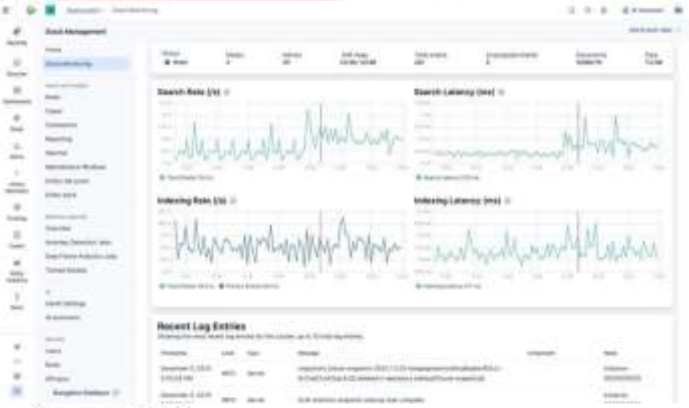
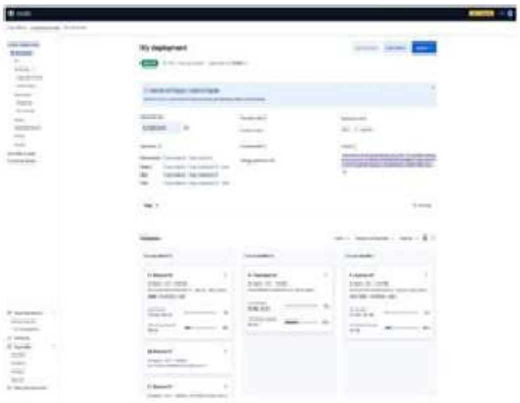


Fig.1.StackMonitoring–Elasticsearch& KibanaOverviewInterface.

Fig.2.PerformanceAnalytics – Search, Indexing &Dashboard



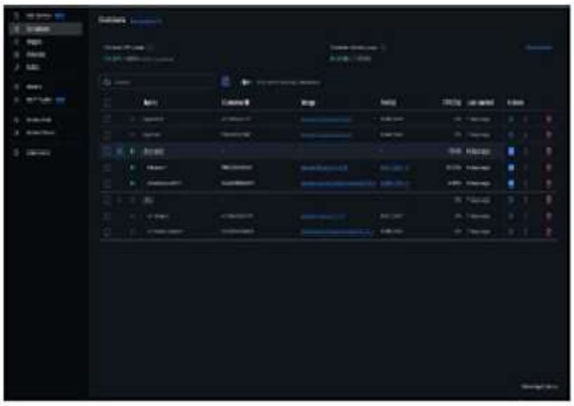


Fig.3.DeploymentManagementConsole

Fig4.ContainerManagementInterface

VI. CONCLUSION AND FUTUREWORK

The Autonomous AI Agent Framework for Cyber Threat Intelligence demonstrates the potential of integrating AI-driven analysis, real-time data monitoring, and automated decision-making to enhance modern cybersecurity. By combining tools such as Wireshark/Pyshark, Elastic search–Kibana, Fast API, and OpenAI-based models, the system effectively detects anomalies, extracts Indicators of Compromise, generates intelligent insights, and reduces manual workload through adaptive learning and automated reporting. The project showcases how autonomous agents can transform traditional CTI processes into a more scalable, proactive, and efficient threat detection ecosystem. Looking ahead, the system can be further strengthened by incorporating SOAR-based automated mitigation, advanced deep learning models, reinforcement learning for adaptive defense, and multilingual threat intelligence processing.Expanding support for behavioral analytics, distributed deployment across cloud environments, and integrating cyberattack simulation platforms will enhance scalability, robustness, and real-world applicability. These future enhancements will make the framework even more autonomous, intelligent, and capable of addressing evolving cybersecurity challenges.

REFERENCES

1. Maasaoui,Z.,Bekri,A.,Merzouki,M., Battou,A.,&Abane,A.(2024).A scalable network-securitymonitoring framework using ELK Stack and machine learning for real-time traffic classification.
2. Robbani,F.D.,Haryatmi,E.,Riyadi,T.A.,Supono,R.A.,Kurniawan,A.B.,&Rosdiana.(2025).Snort-based intrusion detection integrated with Elasticsearch and Kibana for real-time attack visualization.
3. Likitha,R.,Tarun,N.,Pallavi,N.,&Vidhey,V.G.(2024).SuricataIDSintegrationwithELK-basedSIEMfor enhanced threat detection using pattern-recognition algorithms.
4. Rajalakshmi, R., Akash, K., Keshavan, T., Rohit, K., & Vinoth Kumar, M. (2025). Security Onion-based intrusion detection architecture integrating Suricata, Zeek, and Wireshark with ELK for SOC and forensic operations.
5. Regi,S.,&Gurpreet,K.(2024).ELKStackasanopen-sourceSIEMforSMEs:Real-timemulti-sourcelog collection, analysis, and threat detection.
6. Davies, T., Eiza, M. H., Shone, N., & Lyon, R. (2025). Collaborative IDS architectureusing distributed Snort nodes and centralized SIEM correlation for improved attack detection.
7. Farhan,B.I.,&Jasim,A.D.(2024).ComparativestudyofSnortandSuricataintrusiondetectionsystemsfor network security monitoring and performance evaluation.
8. Rahmawati, T., Karna, N., Shin, S. Y., & Putra, M. A. P. (2025). Intrusion Prevention System integrated with ELK for real-time web attack detection in resource-constrained environments.
9. Journal of Information Systems Engineering and Management. (2025). Network security infrastructure integrating Suricata, Syslog server, Elasticsearch, and Kibana for enterprise-scale monitoring and packet inspection.
10. IJNRD.(2025).HybridintrusiondetectionsystemusingSuricata-basedDeepPacketInspectionandML-driven ELK log