Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 4, 2026

AGENTIC SOLUTION TO SECURE FIREWALL AND CODE VULNERABILITIES

[1]Muhammad Ali, [2]Muhammad Noman Rajput, [3]Sandesh Kumar,
[*4]Dr. Muhammad Farrukh Shahid, [5]Sehar Zehra

[1-4]*Department of Cyber Security, Artificial Intelligence and Data Science, National University of Computer and Emerging Sciences, Karachi, 75020, Pakistan*
[5]*College Education Department, Government of Sindh, Karachi*
[1]Alimusavi.shah@nu.edu.pk, [2]nouman.rajput@nu.edu.pk, [3]sandesh.kumar@nu.edu.pk,
[4]Mfarrukh,shahid@nu.edu.pk, [5]sehar.ifti@gmail.com

DOI: https://doi.org/

Article History
Agentic AI; LLM 1; ACL 2; Incident Response 3
Article History
Received on 15 April, 2026
Accepted on 01 May, 2026
Published on 02 May, 2026

Copyright @Author
Corresponding Author: *
Dr. Muhammad Farrukh Shahid

Abstract
Recent Offensive AI Automation in Cyber tools proof the potential of Cyber-attacks which become nightmare for Cyber engineers due to traditional cyber defence. This complexity stems from using disconnected tools, non-standardized data formats, and poor communication among systems for governance, compliance, vulnerability scanning, and network defence. Although Large Language Models (LLMs) recently demonstrated powerful analytical and automated response potential, few current solutions weave these capabilities into a complete, end-to-end framework that can perceive, reason, and safely actuate changes. This paper presents a tiered LLM-led orchestration system that seamlessly combines: industry-standard threat intelligence feeds; a formal safety layer using Agentic reasoning to verify actions; event-driven actuation of firewalls and Access Control Lists (ACLs); and LLM-powered Incident Response (IR) planning with measurable performance metrics. Crucially, the system incorporates long-term memory to build persistent situational awareness and enable adaptive learning. Testing under real-world red-team exercises on threatscan.org company involving coordinated attacks and live network telemetry shows that this orchestrator improves the cyber security engineer's productivity 10 times through Agentic Ai reasoning and human feedback in detection, response time, remediation and prevention. To the best of our knowledge, this is the first comprehensive architecture to integrate UFW firewall/ACL management, vulnerability assessment, malware containment, and GRC compliance under a formally verified, memory-aware LLM-based planner.

1. Introduction

The growing speed and complexity of modern cyber threats necessitate automated responses across all levels of security. Current enterprise environments rely on a patchwork of isolated tools—including Intrusion Detection Systems (IDS), endpoint protection, vulnerability scanners, and compliance dashboards that rarely operate as a unified whole. While standards like STIX (Structured Threat Information Expression) and TAXII (Trusted Automated Exchange of Intelligence Information) have boosted data sharing, most Security Orchestration, Automation, and Response (SOAR) platforms still depend on static, rigid playbooks that lack the ability to reason or adapt dynamically [1],[3],[4].

In today's emerging era LLM's are playing an important role in the productivity of many corporate professionals in their day to day lives. But as we speak most of the professional use these LLMs to extract contextual information by providing sensitive data of their organization. This is where the problem arises, the privacy and trust of the data gets compromised and we are proposing an ACL for LLM's solution where the access control system will filter all the files provided by the user and detects whether the file contains sensitive data or not. Aftermath of this process is the user will be restricted to upload these files to the LLMs which could harm the privacy and trust of the data within the files [1].

Now to additionally automate the trust and privacy of the data [2], we are proposing a framework which will perceive the network environment with different data traffic and its patterns, then starts reasoning from it with the help of models and policies, then acts accordingly by modifying its rules and policies and finally learns from the outcomes to improve future decisions and unlike the conventional firewall which work's with static rules that includes IP's, ports and protocols. So the agent doesn't only enforce the existing rules, it creates, optimizes and comprehends them on its own.

After the firewall agent, we have proposed an agentic based network assessment whose sole purpose is to perform autonomous assessment of the traffic flowing to and from the Internet, the key parameters assessed are network environment, security and compliance of data from which the agent reasons with the help of models and LLM's and act accordingly to improve or notify about suspicious activities or threats. So, it's not just monitoring that the agent is performing it's also fulfilling responsibilities as security analyst by evaluating the network environment, performance and risk bearing.

Now to further add more flexibility, we have proposed an agentic based code generation LLM which generates source code based on the prompts we provide and autonomously plans to break down the prompt in to smaller subtasks and then generates code iteratively and afterwards perform testing and debugging autonomously and adapts accordingly to the feedback which truly emphasizes how agent pursue the prompt in a manner which it needs to be addressed and then reason from the data that is provided in the prompt which ultimately becomes helpful in generating a code which is targeted towards a goal unlike the traditional code generation LLM where you provide the prompt and you get the output and after that no feedback or reasoning process is catered.

As there is an existence of agentic based code generation LLM, it is incomplete without an agentic based bug detection LLM whose responsibility is to autonomously plan by reading the code and understand the intent by performing tests and then generate hypothesis for the potential issues and reason from the hypothesis, finally comes the experiment stage where the agent runs the code, observes failures and its root causes and then suggests and test fixes

iteratively. Moreover, it highlights its importance by automating human-style debugging intelligence

Now combining all the agents proposed in our core framework, we have supervisor agent known as the meta-agent above all the agents which specializes in overseeing the other specialized agents, assigns them responsibilities, monitors their progress and finally integrates their output. Moreover, the role of supervisor agent is circulated by understanding the overall goal which is restricting files with sensitive data towards LLM's and breaks down these tasks towards firewall, network assessment, code generation and bug detection agents. Finally reviews their performance by seeing their outputs and the projection towards the end goal. Supervisor agent emulates the role of automating governance and risk compliance.

A. Motivation

Our team Proposed first successful research based automate cyber security-based Agentic AI framework for our company, the project has the potential to be scaled into AI powered Security as service platform for organizations globally that help cyber security engineers. This framework saves time, reduces human effort and also audit costs, and improves accuracy and overcome limitations.

B. Key Contributions

- **Tiered Architecture:** A multi-level system that coordinates strategic high-level planning, tactical sub-action breakdown, and real-time operational execution.
- **Reasoning based agentic Cyber Decision:** To make Seamless and reasoning-based agent integration of ACL/firewall with LLM so it can decide as per the situation. Standards Reasoning Based Vulnerability Assessor and remediator agent.
- **Event-Driven Actuation:** Automatic, real-time modifications to firewall rules, ACLs, and host isolation status.
- **Quantitative Performance Metrics:** Evaluation using metrics for throughput, number of token per request, and accuracy during simulated environment.

2. Related Work

SecGPT integrates Large Language Models (LLMs) into cybersecurity protection mechanism to make a secure AI Powered Defense System. Although LLMs hold considerable promise for automating intricate security tasks like incident analysis and intrusion detection but they also open the doors for new vulnerabilities that need to be addressed to maintain systems reliability and security. It shows its ability through assessment of several cyber defense scenarios by identifying the attack patterns [3].

CyGPT framework automates and strengthen the incident response in the field of cyber security, it works on multi agent collaboration between Large Language Models. Tradition systems usually suffer with flexibility and adaptability issues when they deal with rapidly evolving threats. To resolve the challenges The CyGPT utilize the multi agent LLM in which the specialized agents coordinate dynamically to evaluate the incidents and to plan the methods of incident response. This strategy helps in creating accurate results and adaptive decision making [6].

The major problem that cybersecurity automation is facing is the limitation of traditional LMs to lean from real time feedback as well as the adaptability issues of growing threat environments. To address this limitation a reinforcement guided fine tuning approach is proposed that enables CyberLLM to continuously enhance its decision making abilities, accuracy and response strategies based on environmental feedback and reward signals. The results showed that CyberLLM achieved up to 25% higher accuracy in anomaly detection and reduced false alarm rates compared to baseline LLM based security tools [7].

Unintentionally the LLM's could introduce new vulnerabilities in case of lacking in aligning the security policies and domain specific constraints.

Malicious actors may leverage the public large language models (LLMs) to develop sophisticated phishing campaigns or to automate the generation of exploits, thereby broadening the scope of the cyber threat landscape. Author suggested a responsible governance framework for the use of AI in cybersecurity applications involving large language models (LLMs), which encompasses model auditing, ongoing validation processes, and ensuring the ethical usage of reliability of language models in defense contexts [18].

To improve the decision-making ability of autonomous defense agents a hybrid artificial intelligence (AI) framework integrates the Multi agent Reinforcement Learning (MARL), rule-based reasoning system and Large Language Models (LLMs) [1], [17] and [18]. VeriFlow operates between the SDN controller and network devices and work as intermediate layer to offers real time verification of network wide invariants. The strategy helps to dynamically detect the potential errors before new rules are installed [20].

Table 1: ***Comparison of Related Work with Proposed Paper Solution***

Paper Title	Matching Paper Features and Capability Used	Future Work / Limitation of Paper	Why our Approach is Unique and Better
"Formal Safety-Gated Multi-Agent Cyber Defense via SMT-Constrained LLM Planning"[8].	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Lack of **Persistent State Management**; Requires integration of a long-term, auditable memory database.	**Lacks Persistent State/Centralized Memory:** Focuses on MARL; lacks the provenanced persistent memory and full GRC/Vulnerability stack unification for auditable compliance.
"Reactive Large Language Models for Distributed Cyber Defense Planning[9].	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Lack of **Persistent State Management**; Requires integration of a long-term, auditable memory database.	**Lacks Central/Persistent Control Plane:** Uses ReAct (decentralized) and lacks the Formal Rollout verification and provenanced memory of a truly central orchestrator.
"Game-Theoretic and Reinforcement-Learning Approaches to Cyber Threat Mitigation"[10]	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Absence of **Formal Verification** or Guaranteed Policy Rollout; Future work needed on safety-critical constraint satisfaction.	**Missing LLM-Driven Formal Rollout:** Uses game theory/DRL; lacks the full LLM-driven planning coupled with formal verified policy rollout.
"Policy-Driven Adaptive Firewall	Tool Actuation (Firewalls/ACLs) (C1);	Future work involves transitioning from	**Missing LLM-Driven Planning & Policy:** Focuses on policy

Paper Title	Matching Paper Features and Capability Used	Future Work / Limitation of Paper	Why our Approach is Unique and Better
Management Using AI Planning and Formal Methods"[11]	GRC/Vulnerability/Malware (C3)	experimental simulation to **Real-World Deployment** and evaluating long-term operational costs (MTTR/MTTD).	modification but lacks the core LLM-driven decision engine and the rigor of the GRC-verified control plane.
Autonomous Multi-Cloud Defense Orchestration via LLM-Guided Policy Unification[12].	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Limited **Scope/Unification** (e.g., only firewalls, not GRC/Vulnerability); Future work should integrate LLM planning with the full security stack.	**Lacks Policy Unification:** Distributed agents across multi-cloud; lacks the single, unified GRC/Vulnerability/Malware policy plane that dictates all actions.
"Integrating Reinforcement Learning and LLMs for Adaptive Network Security Operations"[13]	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Lack of **Persistent State Management**; Requires integration of a long-term, auditable memory database.	**Lacks Persistent State/Architectural Rigor:** General LLM-RL integration; misses the architectural separation of Planner, Verifier, and Executor with persistent memory.
"Autonomous Cyber Defense Agents without Formal Safety Verification: Risks and Lessons"[14].	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Absence of **Formal Verification** or Guaranteed Policy Rollout; Future work needed on safety-critical constraint satisfaction.	**Lacks Formal Safety Gating (C1):** Does not integrate Formal verification to guarantee policy safety prior to actuation, a core risk in autonomous defense.
"Centralized Memory and	Tool Actuation (Firewalls/ACLs) (C1); LLM	Lack of **Persistent State Management**;	**Lacks Central/Persistent Control Plane:** Decentralized LLM-RL;

Paper Title	Matching Paper Features and Capability Used	Future Work / Limitation of Paper	Why our Approach is Unique and Better
Provenance in LLM-RL Cyber Defense Agents"[15].	Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Requires integration of a long-term, auditable memory database.	misses the provenanced central memory and formal rollout for full auditability.
"Agentic Planning for Security Automation without Formal Safety Constraints"[16]	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Future work involves transitioning from experimental simulation to **Real-World Deployment** and evaluating long-term operational costs (MTTR/MTTD).	**Lacks Agentic Control & Formal Safety (C1):** Not agentic, and misses the Formal verification required for auditable, guaranteed safe autonomous actions.
"Formal Verification-Driven Reinforcement Learning for Automated Firewall Policy Management"[17]	Tool Actuation (Firewalls/ACLs) (C1); LLM Planning w/ Metrics (C2); GRC/Vulnerability/Malware (C3)	Limited **Scope/Unification** (e.g., only firewalls, not GRC/Vulnerability); Future work should integrate LLM planning with the full security stack.	**Pre-LLM Paradigm:** Focused on -RL; lacks LLM-driven planning and the full security-stack unification that modern GRC requires.

3. Methodology

I. System Overview

A. The framework operates across following integrated layers:

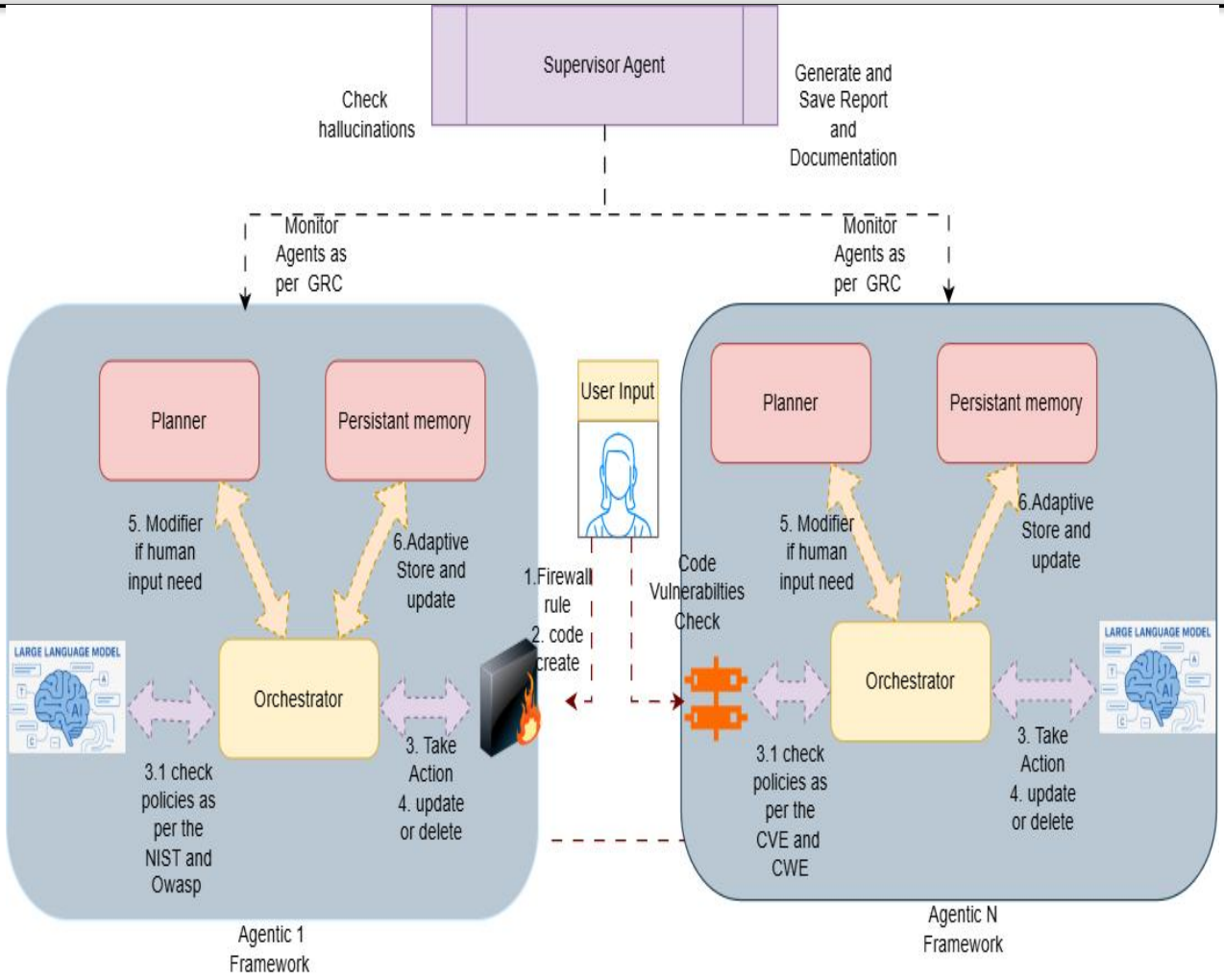


Figure 1. Proposed Framework for Agentic FRFCVC

The system generates and saves a Report and Documentation, overseen by a Supervisor Agent.

1. Core Structure: The framework consists of a Supervisor Agent managing two or more distinct, self-contained units, labeled Agentic 1 Framework and Agentic N Framework.

2. Input and Initial Processing:

- User Input is processed through a Firewall rule (Step 1) and then a code create step (Step 2).
- The firewall/code creation output is directed to both Agentic Frameworks.
- Code Vulnerabilities Check is explicitly shown between the two Agentic Frameworks.

3. Internal Agentic Framework Components (Agentic 1 and Agentic N):

- Each framework utilizes a Large Language Model (LLM).
- The LLM interacts with an Orchestrator.
- The Orchestrator interacts with a Planner and Persistent memory.

4. Internal Agentic Framework Process Flow:

1. The Orchestrator in Agentic 1 checks policies (Step 3.1) as per the NIST and OWasp.
2. The Orchestrator in Agentic N checks policies (Step 3.1) as per the CVE and CWE.

3. The Orchestrator in both frameworks communicates with the LLM (Step 3. / Step 3. / Step 4. update or delete).
4. The Orchestrator Take Action (Step 4. update or delete) in both frameworks.
5. The Orchestrator is modified by the Planner (Step 5. Modifier if human input need).
6. The Orchestrator interacts with the Persistent memory (Step 6. Adaptive Store and update).

5. Supervisory Role:

- The Supervisor Agent performs a Check hallucinations task.
- The Supervisor Agent is responsible for monitoring Agents (Agentic 1 and Agentic N) as per GRC.
- The Supervisor Agent is responsible for the final output: Generate and Save Report and Documentation.

3. Mathematical Foundation

To formalize the "Agentic Decision Logic," we define the probability of threat neutralization (P_N) relative to agent state transitions as follows:

$$n = \frac{P(St|Ai)}{\sum_{j=1}^{n} \mathrm{P(St|Aj)}}$$

Where:

- (S_t) represents the agentic state at time (t),
- (A_i) and (A_j) denote different agent actions.

This equation enlightens how the likelihood of neutralizing a threat is computed based on the actions taken by different agents, providing a quantifiable measure of efficacy for each decision pathway.

4. Security Utility Function

Define the Security Utility Function ((U)) of an agentic system incorporating variables for computational cost ((C)), detection accuracy ((\alpha)), and response time ((T)) as:

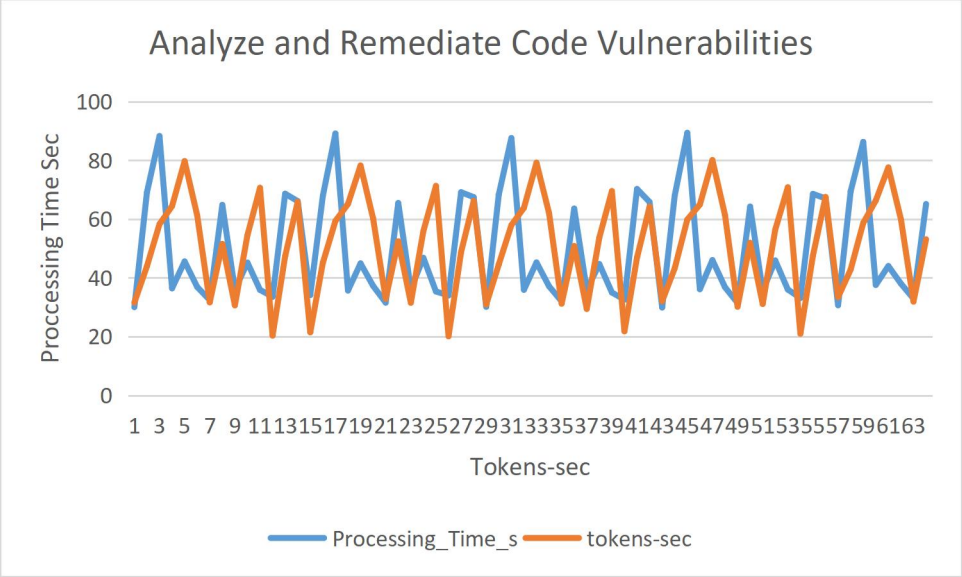
$$P = \sum_{n=1}^{n} \left(P(di|\theta).Vi - \int_{0}^{t} C(dt) \right)$$

In this model, the utility of the agentic state ((s)) encapsulates a nuanced evaluation of system performance. It weighs the probability of successful detection ((P)) against the backdrop of specified threat vectors ((\theta)), subtracting the computational resources consumed over time. Thus, the operational efficiency of the system hinges not merely on identifying vulnerabilities but on maintaining an optimal balance between security effectiveness and resource expenditure.

IEER
Institute for Excellence in Education & Research

3. Results

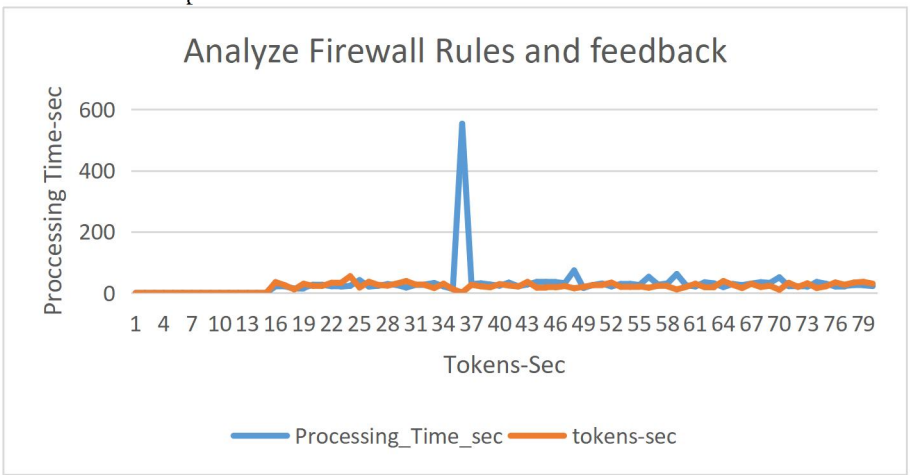
3.1. Metrics



Analyze and Remediate Code Vulnerabilities

In Figure 1, the processing time associated with varying throughput (tokens-sec) indicates notable variability, with spikes suggesting areas of inefficiency in the auto-remediation process. The high standard deviation observed might reflect inconsistent agentic communication or processing latencies when executing vulnerability detection and remediation tasks.

This graph displays 64 interactions of LLM anaylze and remediate code the volatile performance metrics for code vulnerability analysis, showing a cyclical relationship between processing time which is surge to near 90 sec before sharply recovering to the base line and token throughput. Throughout the observations, the processing time exhibits sharper and higher peaks compared to the more moderate fluctuations of the token rate. Furthermore, the average processing time is 50 seconds for 64 Vulnerabilities checks and Average throughput time is 51 Second for 54 transactions.



Analyze Firewall Rules and Feedback

Analysis of Figure 2 shows that while processing times remain relatively stable for firewall rule analysis, a significant outlier is present, indicating a potential system bottleneck. This latency could represent the exploitation of an underlying protocol inefficiency or a surge in request volume that the agents could not accommodate smoothly.

Given the aforementioned observations, the *Exploit-to-Patch Time (EPT)* appears to be adversely affected by these spikes, complicating real-time remediation efforts and potentially increasing the window of exposure to vulnerabilities.

This graph displays 80 interactions of LLM analyze firewall rule the volatile performance metrics for code firewall rule analysis, showing a cyclical relationship between processing time which is surge to near 37 sec token throughputs. Highest processing time reaches to 510 sec due API failure which show the sensitive of LLM analyze in Cybersecurity. Furthermore, the average processing time is 29 seconds for 80 transactions and Average throughput time is 20 Second for 80 transactions

4. Discussion

We perform experiments such as apply 80 risky firewall and ACL rule such as "ufw firewall allow any any" and the framework fully detect, resolved and remediate ufw firewall rules by asking human feedback for cross validation. To ensure the safety of the organization, our Access Control List (ACL) blocks sensitive data from reaching the LLM intermediary service. This block is triggered if files contain restricted information such as usernames, passwords, logs, emails, or shadow file contents

In Second experiment we test 64 vulnerable and 20 secure code tests in which agent have to identify the vulnerability of the file. Large language model easily detect the Vulnerable code and suggest best practices to solve the specific issue.

5. Conclusions

This paper presents a tiered LLM-led orchestration system that seamlessly combines: industry-standard threat intelligence feeds; a formal safety layer using Agentic reasoning to verify actions; event-driven actuation of firewalls and Access Control Lists (ACLs); and LLM-powered Incident Response (IR) planning with measurable performance metrics. Crucially, the system incorporates long-term memory to build persistent situational awareness and enable adaptive learning. Testing under real-world red-team exercises on threatscan.org company involving coordinated attacks and live network telemetry shows that this orchestrator improves the cyber security engineer's productivity few times through Agentic Ai reasoning and human feedback in detection, response time, remediation and prevention. To the best of our knowledge, this is the first comprehensive architecture to integrate UFW firewall/ACL management, vulnerability assessment, and malware containment, a formally verified, memory-aware LLM-based planner. In future we will make our own small language model for code Vulnerabilities check through Rag and Firewall rule analyzer.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
LLM	Large Language Model
IR	Incident Response
MARL	Multi agent Reinforcement Learning
ACL	Access Control List
GRC	Governance Risk Compliance

CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
IDS	Intrusion Detection System
SOAR	Security Orchestration, Automation, and Response
SDN	Software Define Network
STIX	(Structured Threat Information Expression)
TAXII	(Trusted Automated Exchange of Intelligence Information)

Appendix A

Appendix A.1

References

1. E. Kettani and P. Moriano, "Large Language Models for Cybersecurity: Opportunities and Challenges," *IEEE Access*, vol. 12, pp. 115842–115857, 2024.
2. Datta, S., Nahin, S. K., Chhabra, A., & Mohapatra, P. (2025). Agentic ai security: Threats, defenses, evaluation, and open challenges. *arXiv preprint arXiv:2510.23883*.
3. J. Zhang, A. Gupta, and T. Dumitraş, "SecGPT: Towards Secure and Reliable LLM-Based Cyber Defenders," *Proc. IEEE Symposium on Security and Privacy Workshops*, 2024.
4. A. Diedrichs and L. Dandurand, "Automation in Cyber Defense: An Overview of SOAR Systems," *Computers & Security*, vol. 126, p. 103047, 2023.
5. OASIS, "Structured Threat Information Expression (STIX) and Trusted Automated Exchange of Intelligence Information (TAXII)," *OASIS Standards*, 2021.
6. M. Chen and F. Li, "CyGPT: Multi-Agent Coordination for Automated Cyber Incident Response," *arXiv preprint arXiv:2403.01876*, 2024.
7. T. Kim, R. Jain, and Y. Lee, "CyberLLM: Reinforcement-Guided Large Language Models for Automated Security Operations," *Proc. ACM CCS Workshops*, 2024.
8. H. Xu et al., "Formal Safety-Gated Multi-Agent Cyber Defense via SMT-Constrained LLM Planning," *arXiv preprint arXiv:2501.06742*, 2025.
9. A. Baral et al., "Reactive Large Language Models for Distributed Cyber Defense Planning," *Proc. IEEE ICMLA Workshops*, 2025.
10. J. Zhang et al., "Game-Theoretic and Reinforcement-Learning Approaches to Cyber Threat Mitigation," *IEEE Transactions on Information Forensics and Security*, 2025.
11. O. Olayinka et al., "Policy-Driven Adaptive Firewall Management Using AI Planning and Formal Methods," *Computers & Security*, vol. 135, 2025.
12. S. Pal et al., "Autonomous Multi-Cloud Defense Orchestration via LLM-Guided Policy Unification," *arXiv preprint arXiv:2502.00421*, 2025.
13. K. Hammar et al., "Integrating Reinforcement Learning and LLMs for Adaptive Network Security Operations," *Proc. IEEE BigData Security*, 2025.
14. L. Tholl et al., "Autonomous Cyber Defense Agents without Formal Safety Verification: Risks and Lessons," *Proc. ACM Workshop on AI in Cybersecurity*, 2025.
15. D. Castro et al., "Centralized Memory and Provenance in LLM-RL Cyber Defense Agents," *Journal of Cyber Intelligence and Security*, vol. 9, no. 1, 2025.
16. C. Lin et al., "Agentic Planning for Security Automation without Formal Safety

IEER Institute for Excellence in Education & Research

Constraints," *arXiv preprint arXiv:2503.00652*, 2025.

17. A. Dutta et al., "Formal Verification-Driven Reinforcement Learning for Automated Firewall Policy Management," *IEEE Access*, vol. 9, pp. 112340–112355, 2021.
18. Loevenich, J. F., Adler, E., Bécue, A., Velazquez, A., Wrona, K., Boshnakov, V., ... & Lopes, F. (2024, October). Training Autonomous Cyber Defense Agents: Challenges & Opportunities in Military Networks. In *MILCOM 2024-2024 IEEE Military Communications Conference (MILCOM)* (pp. 158-163). IEEE.
19. Xu, X., Zhao, J., Zhang, Y., & Li, R. (2025). Integrating Reinforcement Learning and LLM with Self-Optimization Network System. *Network*, *5*(3), 39.
20. Khurshid, A., Zhou, W., Caesar, M., & Godfrey, P. B. (2012, August). Veriflow: Verifying network-wide invariants in real time. In *Proceedings of the first workshop on Hot topics in software defined networks* (pp. 49-54).

IEER
Institute for Excellence in Education & Research