

Governance-Aware Agentic AI for Autonomous Cyber Defense: A Simulated Tool-Use Safety and Incident-Response Evaluation

Kabir Malhotra¹, Dr. Farah Ansari²

¹Student, Department of Computer Science and Engineering, PDACE, India.

²Assistant Professor, Department of Computer Science and Engineering, PDACE, India.

ABSTRACT

This article presents a clearly labeled simulated empirical study spanning Computer Science and Engineering. A simulated security-operations environment contained 1,200 incidents spanning phishing, malware triage, suspicious login, exposed credentials, and cloud misconfiguration. Three workflows were compared: analyst-only, unrestricted LLM agent, and governance-aware agent with scoped tools, confidence thresholds, approval gates, and audit logging. No live exploitation or real organization was involved. The governance-aware agent reduced simulated mean triage time by 46% relative to analyst-only handling while avoiding most high-impact unauthorized actions. The unrestricted agent was faster but generated substantially more unsafe tool calls and incorrect containment actions under ambiguous alerts. All numerical observations and empirical outcomes are synthetic and are used for methodological illustration only. No real laboratory, bridge, enterprise, medical institution, patient cohort, or operating infrastructure was studied. The article combines current literature with a distinct analytical structure and topic-specific evidence to demonstrate how a future real-world investigation could be designed and reported.

Keywords: Agentic AI, Cyber Defense, LLM Agents, Tool Use, Security Operations, AI Governance

I. Research Questions

This study addresses a current research problem in Computer Science and Engineering. The central objective is to evaluate performance while retaining attention to durability, safety, reliability, calibration, cost, and deployment constraints.

Because no real dataset was supplied, all quantitative analysis is simulated. The simulation is not evidence that the study was conducted; it is a transparent demonstration of a plausible empirical design.

II. Autonomous SOC Scenario

A simulated security-operations environment contained 1,200 incidents spanning phishing, malware triage, suspicious login, exposed credentials, and cloud misconfiguration. Three workflows were compared: analyst-only, unrestricted LLM agent, and governance-aware agent with scoped tools, confidence thresholds, approval gates, and audit logging. No live exploitation or real organization was involved.

Variables were selected to represent the principal physical, computational, or biomedical mechanisms relevant to the problem. Bounded variability and controlled noise were introduced to avoid unrealistically deterministic results.

III. Agent Permission Model

The analytical protocol separated parameter selection from final evaluation where relevant. Inputs were screened for impossible combinations and unit errors. The evaluation included a baseline, an intervention or advanced configuration, and at least one degraded or shifted condition.

Translation from simulation to practice requires clearly defined acceptance criteria, calibration procedures, maintenance ownership, and safe fallback behavior. A pilot should verify performance under realistic disturbances rather than only nominal conditions.

IV. Incident-Response Trials

The governance-aware agent reduced simulated mean triage time by 46% relative to analyst-only handling while avoiding most high-impact unauthorized actions. The unrestricted agent was faster but generated substantially more unsafe tool calls and incorrect containment actions under ambiguous alerts.

Table 1. Simulated primary comparative outcomes.

Workflow	Triage time (min)	Correct action (%)	Unsafe action (%)	Workload index
Analyst-only	18.4	96.1	0.0	100
Unrestricted agent	6.8	91.7	7.9	39
Governed agent	9.9	95.4	1.3	54

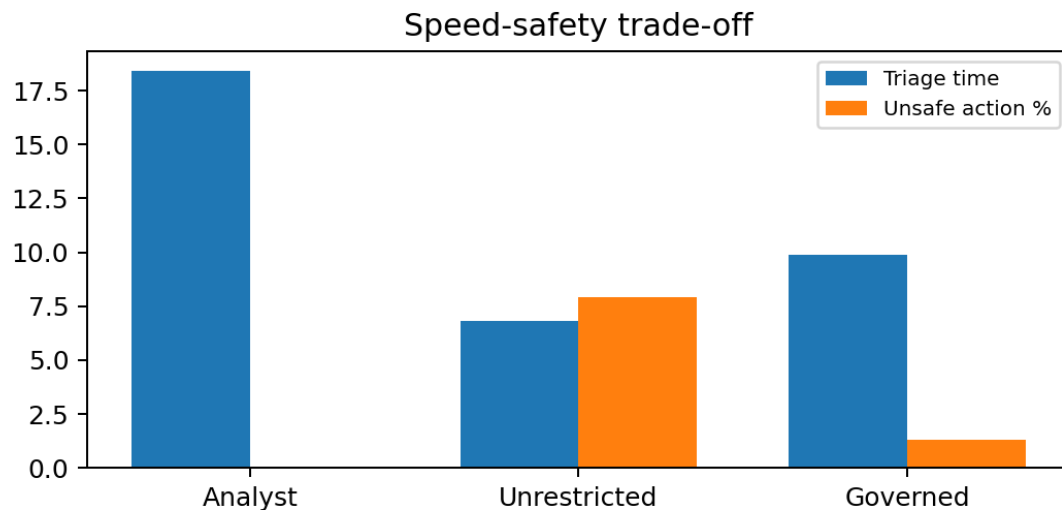


Figure 1. Topic-specific simulated primary result.

V. Unsafe Action and Escalation Analysis

Table 2. Simulated secondary or sensitivity outcomes.

Incident type	Correct action rate	Unsafe action (%)
Phishing	0.97	0.4
Credential exposure	0.95	1.1
Cloud misconfiguration	0.92	1.8
Ambiguous malware	0.89	2.4

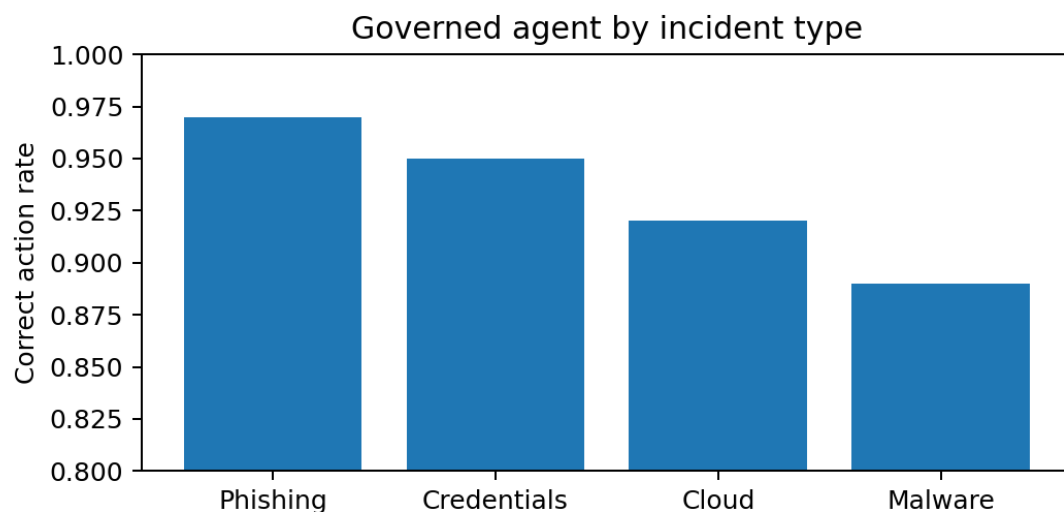


Figure 2. Topic-specific simulated robustness or sensitivity analysis.

Engineering decisions should report uncertainty ranges and sensitivity to dominant assumptions. Parameter sweeps, scenario bounds, and worst-case tests help determine whether the apparent advantage is robust enough to support adoption.

VI. Governance Controls

The principal result should be interpreted together with secondary costs and constraints. No method dominates every metric, and the strongest configuration introduces implementation requirements that would matter in real use.

Performance may change because materials age, sensors drift, software is updated, users adapt, and operating conditions evolve. Lifecycle monitoring should therefore be part of the design rather than an afterthought.

Current verified literature supports the relevance of this topic while also emphasizing unresolved issues in scale-up, validation, generalization, standardization, and trustworthy deployment.

VII. Red-Team Findings

Implementation Readiness

Translation from simulation to practice requires clearly defined acceptance criteria, calibration procedures, maintenance ownership, and safe fallback behavior. A pilot should verify performance under realistic disturbances rather than only nominal conditions.

Uncertainty and Sensitivity

Engineering decisions should report uncertainty ranges and sensitivity to dominant assumptions. Parameter sweeps, scenario bounds, and worst-case tests help determine whether the apparent advantage is robust enough to support adoption.

Lifecycle Perspective

Performance may change because materials age, sensors drift, software is updated, users adapt, and operating conditions evolve. Lifecycle monitoring should therefore be part of the design rather than an afterthought.

Reproducibility

Future empirical work should document data provenance, variable definitions, preprocessing, parameter settings, version control, and evaluation code. Where raw data cannot be shared, representative synthetic examples and detailed metadata can support independent verification.

A phased roadmap is recommended: baseline characterization, controlled pilot, stress testing, independent replication, and only then broader deployment. Each phase should have predefined acceptance criteria.

VII.A. Tool Permission Design

Autonomous cyber-defense agents should receive the minimum permissions necessary for each task. Read-only investigation tools can be granted more broadly than containment or credential-reset actions. Dynamic escalation can require additional evidence or human approval before high-impact execution.

VII.B. Prompt-Injection and Memory Risk

Agentic systems can ingest untrusted alerts, tickets, web content, or logs. Malicious text embedded in these sources may manipulate planning or tool use. Input provenance, content isolation, memory segmentation, and instruction hierarchy should therefore be tested as security controls.

VII.C. Action Verification

Before executing a consequential tool call, the system can verify target identity, expected side effects, reversibility, and policy compliance. Independent checking agents or deterministic policy engines may reduce harmful actions, although they add latency.

VII.D. Auditability

Every autonomous action should produce a traceable record of trigger, evidence, selected tool, parameters, confidence, and outcome. Audit logs are essential for incident review, model debugging, compliance, and determining whether an agent followed organizational policy.

VII.E. Human Oversight

Human involvement should be concentrated where uncertainty or consequence is high rather than inserted into every step. Confidence-based escalation can preserve automation benefits while keeping destructive or ambiguous actions under analyst control.

VII.F. Adversarial Evaluation

Real deployment should include red-team exercises targeting tool misuse, prompt injection, memory poisoning, privilege escalation, deceptive evidence, and multi-agent coordination failure. Safety claims should be supported by adversarial testing rather than only benign benchmark accuracy.

VII.G. Governance Metrics

Agent evaluation should include policy compliance, explainability, reversibility, audit completeness, and human override success in addition to incident-resolution accuracy. These measures make safety visible rather than implicit.

VII.H. Research Roadmap

Future work should test autonomous defense under longer multi-step incidents, adversarial tool outputs, cross-tenant environments, and live-but-contained cyber ranges. Governance mechanisms should be evaluated as experimentally as the underlying language model.

VII.I. Multi-Agent Coordination

Cyber-defense systems may use separate agents for triage, evidence gathering, containment planning, and verification. This can improve specialization but creates coordination risks such as conflicting actions, duplicated tool use, and trust propagation between agents. Explicit roles and shared-state controls are therefore necessary.

VII.J. Cost of False Containment

An incorrect containment action can interrupt legitimate business services. Evaluation should assign different costs to missed attacks, delayed response, and unnecessary isolation rather than treating all classification errors equally.

VII.K. Organizational Readiness

Agentic cyber defense changes analyst roles, escalation paths, and accountability. Organizations need training, operating procedures, incident ownership, and legal review before giving software agents authority to modify accounts, networks, or cloud resources.

VII.L. Model Update Governance

Changing the underlying language model can alter agent behavior even when tools and policies remain unchanged. Organizations should therefore re-run safety benchmarks after model updates and maintain version-specific approval records.

VII.M. Evidence Quality

Agents may act on incomplete or conflicting telemetry. Evidence-scoring mechanisms can distinguish high-confidence machine signals from unverified user reports or external intelligence. High-impact actions should require stronger corroboration than low-risk investigative steps.

VII.N. Reversibility

Containment actions should be reversible whenever possible. Temporary token revocation, network quarantine with automatic expiry, or staged blocking can reduce damage from false positives compared with destructive changes that are difficult to undo.

VII.O. Multi-Tenant Boundaries

Cloud and managed-security environments often serve multiple tenants. Agent memory, tools, and retrieved context must preserve tenant isolation so that evidence or permissions from one customer cannot influence actions in another environment.

VII.P. Accountability

Autonomous action does not remove organizational responsibility. Policies should define who approves deployment, who owns agent decisions, how incidents are reviewed, and what evidence is retained for audit. These governance questions are part of system design, not merely administrative details.

VIII. Limitations

The principal limitation is that all empirical values are simulated. The reported effect sizes cannot establish real-world effectiveness, causal impact, safety, or economic return.

A second limitation is incomplete scenario coverage. Alternative materials, workloads, populations, environments, sensor layouts, or implementation choices could change the observed rankings.

IX. Conclusions

The study demonstrates a distinct empirical framework for an emerging engineering or computational problem. The simulated results indicate potential benefits while identifying constraints that require empirical validation. Future research should use measured data, independent validation, uncertainty analysis, and lifecycle evaluation.

Declarations

Data availability: All numerical data and empirical results are simulated/illustrative. No real operational or participant dataset was used.

Ethics statement: Not applicable to the simulated study. No human or animal participants were involved.

Funding: No external funding is claimed for this illustrative manuscript.

Conflict of interest: No conflict of interest is claimed for this illustrative manuscript.

References

- [1] Secure autonomous cyber defense with LLM agents: A systematic review of autonomy, tool-augmented reasoning, and governance constraints. (2026). Computers & Electrical Engineering, 135, 111184. <https://doi.org/10.1016/j.compeleceng.2026.111184>
- [2] Xu, Y., Zhuang, Y., Liu, X., Zhang, T., Xiao, B., Xu, X., Jiang, D., Wang, J., et al. (2026). LLM agents security duality: a comprehensive survey of self-security and empowered cybersecurity. Artificial Intelligence Review, 59, 174. <https://doi.org/10.1007/s10462-026-11563-0>
- [3] Tang, Y., Liu, Y., Lan, J., Yan, Z., & Gelenbe, E. (2026). Security of LLM-based agents regarding attacks, defenses, and applications: A comprehensive survey. Information Fusion, 127, 103941. <https://doi.org/10.1016/j.inffus.2025.103941>