RAG-H: A RAG-Hardened SOC Framework for Trustworthy Threat Intelligence and AI Defence

Daniela Ceraku
Thales Digital Factory, Paris, France

cedaniela613@gmail.com

Abstract: Retrieval-Augmented Generation (RAG) is now the dominant approach for enhancing Large Language Models in Cyber Threat Intelligence (CTI), yet it introduces a new attack surface: knowledge base poisoning and prompt injection. Existing defences address isolated stages of the RAG pipeline, offer no end-to-end integrity guarantees, and are evaluated on answer accuracy rather than Security Operations Center (SOC) mission impact. This paper introduces RAG-H, a multi-layer defence framework that secures the full RAG pipeline for SOC use. Ingestion enforces provenance checks and source-reputation scoring so that only trustworthy intelligence enters the knowledge base, retrieval combines semantic relevance with corpus-level credibility and analyst feedback to filter poisoned content, generation applies context sanitisation and self-verification to surface low-confidence outputs for analyst review. We evaluate RAG-H on a CTI corpus of 300 documents, 90 of which are poisoned, and measure which pipeline stages are most vulnerable, which layered controls most reduce the Poison Impact Rate without unacceptable latency, and how the defences affect incident response. Results show that layered trust, consistency, and governance controls significantly mitigate poisoning effects. The framework establishes a reproducible methodology for securing RAG systems in SOCs and introduces evaluation metrics that connect technical robustness to operational impact. An open-source proof-of-concept accompanies the work as a baseline for future research.

Keywords: Retrieval-augmented generation (RAG), Security operations center (SOC), Cyber threat intelligence (CTI), Incident response (IR)

1. Introduction

Security Operations Centres (SOCs) operate under persistent time pressure, analysts must ingest rapidly evolving Cyber Threat Intelligence (CTI), correlate signals from heterogeneous telemetry, and decide on response actions within tight operational windows. To reduce this cognitive load, vendors are integrating Large Language Models (LLMs) into triage and incident response workflows, and recent industry surveys report broad but cautious adoption across SOC teams (SANS, 2025). Empirical studies of human-AI collaboration in SOCs confirm that LLM assistance can accelerate investigation but also show that unreliable outputs erode analyst trust and increase verification overhead (Singh et al., 2025). The central failure mode is well-documented, LLMs hallucinate, fail to distinguish authoritative intelligence from unverified claims, and amplify errors when exposed to manipulated context. Retrieval-Augmented Generation (RAG) has therefore become the dominant approach to grounding LLM outputs in external CTI sources and has been explicitly adapted to adversarial technique annotation in cyber threat reports (Lekssays et al., 2025).

Unlike static knowledge bases, CTI corpora are continuously updated from sources of uneven trustworthiness, including vendor reports, open-source feeds, blogs, and internal investigations. This ingestion pipeline exposes RAG systems to a new attack surface: *knowledge base poisoning*. Zou et al. (2025) show in PoisonedRAG that injecting only a handful of adversarial documents into a corpus is sufficient to steer LLM answers towards attacker-chosen outputs, especially when the injected content is optimized for semantic similarity to common queries. Chen et al. (2024) extend this result to agentic settings, demonstrating that once poisoned content enters long-term memory or shared knowledge bases, the degradation persists across sessions. Adversaries can also embed hidden instructions directly within CTI text, turning retrieved evidence into a prompt-injection vehicle that bypasses generation-time safeguards (RoyChowdhury et al., 2024). In a SOC context, these attacks do not merely produce a wrong answer, they can misdirect triage, contaminate incident reports, and propagate into downstream SOAR (Security Orchestration, Automation and Response) playbooks.

1.1 General RAG Pipeline

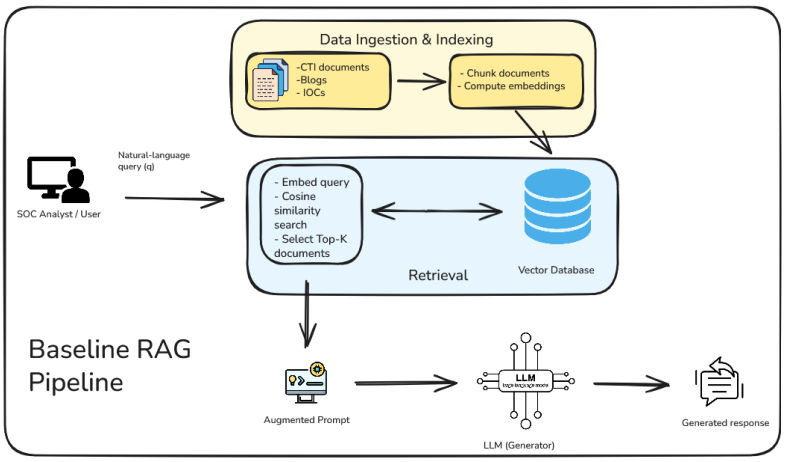

Figure 1: Baseline RAG Pipeline

Figure 1 summarises the three stages of a baseline RAG pipeline as adopted in most CTI assistants. During ingestion, heterogeneous CTI documents are chunked, embedded, and indexed in a vector store. At query time, retrieval ranks documents by semantic similarity to the analyst's query and returns the top-k passages, optionally re-ranked. These passages are concatenated with the original query to form the augmented prompt consumed by the generation stage. Two properties of this pipeline matter for security. First, every stage is a transitive trust point: a failure in ingestion propagates unchanged through retrieval and generation. Second, the pipeline treats all retrieved documents as equally authoritative, which is a reasonable assumption for clean corpora but a dangerous one for CTI, where authorship ranges from MITRE ATT&CK to unvetted blog posts. RAG-H targets both properties.

1.2 Prior Work

Prior work on RAG robustness clusters into three directions. **Attack characterisation**. Zou et al. (2025) formalise knowledge-corruption attacks on RAG and show that a small number of adversarial documents is sufficient to steer generation, Chen et al. (2024) extend this threat model to agents operating over persistent memory; and RoyChowdhury et al. (2024) demonstrate confused-deputy risks in RAG-based enterprise assistants. These works establish that poisoning is practical and cheap, but they stop at the attack side and do not prescribe defences. **Retrieval-stage defences**. Jia et al. (2025) propose RAGRank, a PageRank-style authority score tailored to CTI pipelines, Qian et al. (2025) introduce ClaimTrust, which propagates trust over claim graphs extracted from retrieved documents, Zhou et al. (2025) present TrustRAG, combining clustering with self-assessment to filter suspicious passages and Xiang et al. (2024) derive certifiable robustness bounds under an isolate-then-aggregate retrieval strategy (RobustRAG). These defences are the most mature part of the literature, yet each operates on a single stage and assumes the knowledge base itself is already curated. **CTI-specific RAG**. Lekssays et al. (2025) apply RAG to adversarial technique annotation (TechniqueRAG), showing that domain-adapted retrieval substantially improves CTI tagging quality, however, they do not address adversarial robustness. Complementary to this, empirical work on analyst-AI collaboration in SOCs (Singh et al., 2025) and the SANS 2025 industry survey (SANS, 2025) report that analysts reject AI assistance when they cannot audit why a given evidence chunk was surfaced. Taken together, the literature provides strong single-stage defences and a clear understanding of attacks, but leaves three gaps unaddressed: (i) no integrated defence spans ingestion, retrieval, and generation under a single trust model, (ii) evaluations optimise answer accuracy rather than SOC operational metrics such as Poison Impact Rate, analyst time, or escalation correctness, and (iii) no prior framework exposes trust signals in a form that analysts can audit during incident response. RAG-H targets these three gaps.

1.3 Contributions

To address these gaps, we introduce RAG-H, a hardened RAG framework designed for SOC use. Rather than securing individual components in isolation, RAG-H treats the RAG system as an end-to-end decision pipeline with a single trust model that spans ingestion, retrieval, generation, and operational handover. This paper makes the following contributions:

C1. A multi-layer defence architecture in which ingestion enforces provenance and schema checks with source-reputation scoring, retrieval combines semantic similarity with corpus-level credibility derived from a support graph and analyst feedback, and generation applies context sanitisation and self-verification.

C2. A SOC-aligned evaluation methodology centred on the Poison Impact Rate, latency, and analyst-oversight metrics, rather than answer accuracy in isolation.

C3. An empirical study on a CTI corpus of 300 documents (90 poisoned) that quantifies which pipeline stages are most vulnerable to adversarial interference and which layered controls most reduce the Poison Impact Rate without unacceptable latency.

C4. An open-source proof-of-concept implementation providing a reproducible baseline for future research on securing RAG systems in SOC environments.

These contributions are structured around three research questions:

RQ1. *Which stages of the RAG pipeline are most vulnerable to knowledge-base poisoning and prompt injection in a CTI setting?*

RQ2. *Which combination of ingestion-, retrieval-, and generation-layer controls most reduces the Poison Impact Rate without introducing unacceptable latency or false-positive escalations?*

RQ3. *How do the trust signals produced by RAG-H influence incident response, and are they usable as an audit trail for analyst oversight?*

2. Methodology

2.1 RAG Hardened Framework

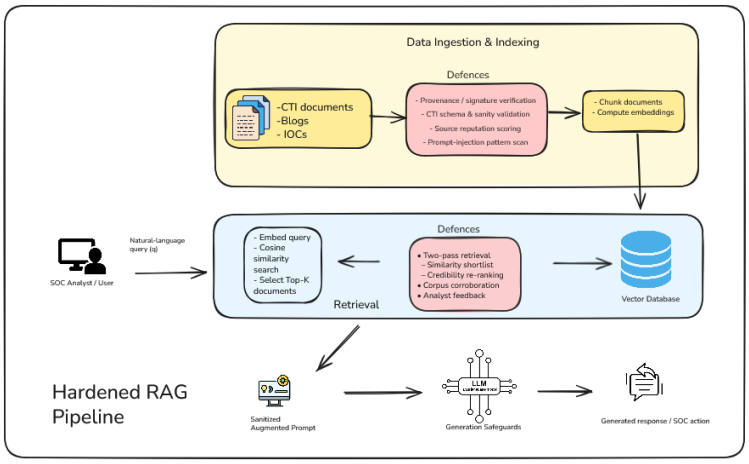

Figure 2: RAG Hardened Pipeline

2.1.1 Phase 1: Ingestion

Prior work shows how poisoning affects retrieval and generation in RAG systems but offers limited protection at ingestion time, before documents enter the knowledge base. This is especially risky in CTI, where reports are continuously ingested from sources with varying trust levels. To mitigate persistent poisoning, we introduce a hardened ingestion process that evaluates each document on entry and assigns a **trust score**. Documents may include text, metadata (e.g., source, timestamp), optional Indicators of Compromise (IOCs), and sometimes cryptographic signatures. During ingestion, the system verifies signatures, checks compliance with CTI schema and basic validity (required fields, timestamps, IOC formats, and absence of prompt-injection instructions), and evaluates source reputation (e.g., trusted providers like MITRE ATT&CK or government feeds). These signals are combined into a single trust score reflecting document reliability at ingestion. Low-trust documents are rejected, medium-trust ones are admitted but labelled as *low confidence*, and high-trust documents are ingested normally. The trust score is stored in metadata and later guides retrieval and generation, enabling the system to weigh document reliability rather than treating all content equally.

2.1.2 Phase 2: Retrieval

Retrieval is the most fragile step in a RAG pipeline under knowledge base poisoning. The reason is simple: whatever the retriever selects becomes the evidence the generator relies on. If the retrieved context is malicious, the generator is effectively forced to "reason" from compromised material and may produce confident but wrong CTI conclusions. RAG-H therefore treats retrieval as an evidence selection problem where relevance is required but not sufficient. The retriever should return documents that match the query and look trustworthy based on how well they are supported by the corpus and by operational signals available in SOC workflows.

Standard RAG retrieval is driven by embedding similarity. The query and each document are embedded into vectors, and the retriever ranks documents by cosine similarity:

$$s(q,d) = \cos(e_q, e_d) = \frac{e_q^{\top} e_d}{\| e_q \| \; \| e_d \|}.$$

A similarity-only retriever returns:

$$R_{\text{sim}} = \text{TopK}_{K_{\text{sim}}} \; s(q,d).$$

This works well when the KB is clean, because high similarity usually correlates with usefulness. Under poisoning, however, similarity becomes an easy target as an attacker can insert text intentionally written to match common query patterns (e.g., "is this IOC malicious?", "attribution of campaign X", "verdict for domain Y"). These poisoned documents can become "nearest neighbours" to many queries, showing up repeatedly even if they are unsupported or misleading. To make retrieval harder to manipulate, RAG-H adds a corpus-level credibility signal that does not rely on a single query-document similarity score. The main idea is to check whether a document is supported by other documents in the knowledge base. We represent the KB as a support graph where each document is a node, and a directed edge indicates that one document supports or corroborates another. Support can be inferred from citations, shared and confirmed IOCs, consistent claims across reports, overlapping technical indicators, or explicit references. Each document is then assigned a global credibility score based on how much support it receives from the rest of the corpus, with support from already credible documents counting more than support from weak or isolated ones. Poorly connected documents, or those supported only by a small, suspicious cluster, receive low credibility. This makes poisoning more difficult, because injected documents often appear isolated, they may match common queries, but they lack strong, independent corroboration from trusted parts of the corpus.

Adapting credibility using SOC analyst feedback

In SOC environments, analysts continually validate evidence during investigations. RAG-H leverages this feedback to keep credibility scores current and resistant to source drift. Each document is linked to a source (e.g., vendor feeds, agency reports, internal intel, or open web), and the system tracks whether documents from each source are confirmed or rejected over time.

From this, it maintains a dynamic reputation score, consistently validated sources gain credibility, while unreliable ones lose it. This reputation biases document scoring, making evidence from recently validated sources more likely to be trusted.

Two-pass retrieval rule

RAG-H uses a two-pass retrieval strategy because credibility alone cannot ensure relevance. In the first pass, the system retrieves a larger shortlist using cosine similarity to ensure that candidate documents are on topic. In the second pass, it re-ranks only this shortlist using the corpus-level credibility score and selects the final k documents based on the RAGRank study. This design avoids two common failures. First, credible but irrelevant documents are never retrieved, because all final results must come from the similarity shortlist. Second, poisoned documents that rank highly by similarity are suppressed if they lack corroboration and have low credibility.

2.1.3 Phase 3: Generation

The generation phase is a critical phase because the language model directly consumes retrieved documents as input. If these documents have been poisoned or manipulated, the model may follow hidden instructions or reason from false evidence. In the CTI, this can lead to prompt injection, incorrect attribution of attacks, or

hallucinations. To reduce these risks, we introduce generation safeguards that strictly control how retrieved context is interpreted and limit the model to using documents as evidence rather than as instructions.

Context Sanitization

Before any retrieved document reaches the language model, it undergoes a deterministic sanitization step to remove prompt injection artifacts embedded in otherwise legitimate CTI reports. Adversarial CTI documents often combine credible threat information with hidden instructions designed to influence model behaviour. Generic filters are insufficient because CTI content naturally includes technical commands and alerts that may appear suspicious but are benign. To address this, sanitization targets known poisoning patterns from adversarial CTI narratives. Each document is scanned line by line using fixed rules, and lines matching injection patterns are removed. These include explicit markers (e.g., "INSTRUCTION:", "SYSTEM PROMPT:"), override phrases (e.g., "IGNORE PREVIOUS"), and coercive elements like repeated exclamation marks or attention-grabbing warnings. The system also removes subtle directives identified during analysis of poisoned data, such as unusual phrasing or formatting linked to successful attacks. This process is fully deterministic and independent of the language model, preventing adaptive attacks where the model might be influenced during filtering. Only the offending lines are deleted, preserving the surrounding factual content. As a result, documents remain valid evidence but cannot function as instruction channels. By enforcing sanitization at generation time, CTI documents are treated strictly as passive data, ensuring the model reasons only over clean, non-manipulative inputs.

2.1.4 Phase 4: Operation

The operation phase controls how AI generated outputs are turned into real actions inside the SOC. While many RAG-based security assistants stop after generating an answer, RAG-H extends the pipeline to operational execution and ensures that automation never bypasses analyst judgment when there is uncertainty. For this reason, RAG-H enforces a strict human-in-the-loop gate before any automated action is executed. The system aggregates trust signals from all earlier phases into a single global trust score $T \in [0,1]$. A trust threshold τ defines when automation is allowed. In addition, the system tracks whether semantic contradictions were detected during retrieval or generation, represented by a binary flag Δ. If the trust score is below the threshold, or if any contradiction is detected, automated execution is blocked and the case is escalated to an analyst. Unlike traditional SOAR guardrails that rely only on static rules, this gate is AI aware and directly uses uncertainty signals produced by the RAG pipeline.

To control execution in a structured way, RAG-H maps each AI generated output to a predefined action class. The action class determines how the system handles execution and whether human approval is required.

Table 1: Action Classes and Execution Policies

Action Type	Description	Execution Policy
Informational	Descriptive outputs such as threat summaries or IOC explanations	Automatically logged
Advisory	Recommendations or suggested mitigations	Analyst approval required
Critical Action	High-impact actions: blocking an IP, isolating a host, etc.	Analyst approval + evidence threshold

When analyst validation is required, the system generates a structured decision package with the proposed action, trust score, detected contradictions, and supporting CTI evidence. A short explanation shows why the evidence supports the recommendation, enabling quick, informed decisions.

The operation phase defines execution control only, quantitative evaluation is covered in the Results section. A core contribution of RAG-H is selective human oversight, triggered by trust and consistency signals rather than applied uniformly. This ensures automation runs only with strong evidence, is halted under uncertainty, and remains transparent to analysts.

3. Experiments and Evaluation Results

3.1 Experimental Setup

The experimental setup was designed to approximate a realistic RAG workflow under controlled conditions. The corpus includes 200–500 CTI documents from MITRE and CISA, combining trusted reports with deliberately poisoned samples containing misleading claims and embedded instructions. A set of 100 analyst-style SOC queries was automatically generated to reflect common investigation tasks.

All experiments use the GPT-4o-mini model via the OpenAI API for generation. Retrieval relies on a vector index built with FAISS. Adversarial content is generated using custom scripts simulating both knowledge base poisoning and instruction injection attacks. The proof of concept is available here: https://github.com/DanielaCe18/RAG-H.git

3.2 Experiments and Evaluation Results

This section reports the empirical evaluation of the proposed RAG-Hardened pipeline compared to the baseline RAG system. All experiments are conducted using the same automatically generated query set, derived from the underlying CTI database, and evaluated across two corpora: clean (no poisoned documents) and poisoned (containing adversarially injected documents). For each query, both systems retrieve the top-k = 5 most related documents and generate an answer.

Answer Accuracy

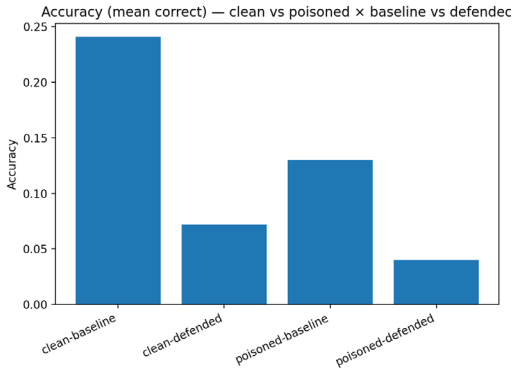

Figure 3: Accuracy (mean correct): clean vs poisoned × baseline vs defended

In this experiment, we measure answer correctness under a strict, evidence-based evaluation criteria. An answer is marked as correct only if it (i) contains the expected identifier (MITRE ATT&CK technique ID or CVE - Common Vulnerabilities and Exposures entry) and (ii) explicitly references supporting evidence from the retrieved corpus. This definition intentionally penalizes speculative or hallucinated responses, even if the surface content appears plausible. On the clean corpus, the baseline system achieves higher accuracy. This result reflects the baseline's behaviour: it always attempts to produce a complete answer, regardless of evidence quality. When the corpus is clean, this strategy often succeeds. The defended pipeline shows lower accuracy on the clean corpus. This decrease **is not due to incorrect reasoning**, but rather to intentional abstention. When retrieved documents do not provide sufficiently strong or consistent evidence, the defended system refuses to answer or escalates the query for analyst review. Such refusals are counted as incorrect under the evaluation metric, even though they represent **safer behaviour**. On the poisoned corpus, accuracy decreases for both systems. However, the defended pipeline exhibits a sharper drop. This reflects the system's sensitivity to conflicting or untrusted evidence: instead of producing potentially misleading answers, it opts to withhold judgment. The baseline system, by contrast, continues to generate answers even when the supporting context is adversarial.

Poison Exposure at Retrieval Time

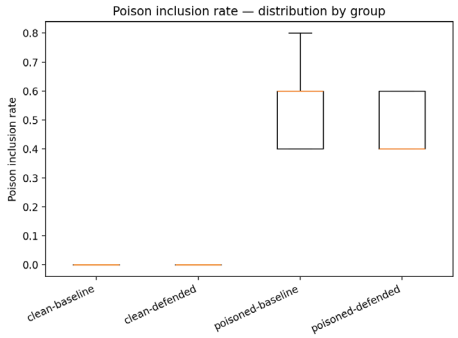

Figure 4: Poison inclusion rate: distribution by group

This experiment measures poison inclusion rate, defined as the fraction of retrieved documents (out of the top-k) that are marked as poisoned. This metric captures retrieval time exposure to adversarial content and is computed before any generation or filtering logic is applied. On the clean corpus, both baseline and defended systems show a poison inclusion rate of 0.0 for all queries. This confirms that the ingestion and indexing pipeline **successfully removed poisoned** documents prior to retrieval.

On the poisoned corpus, the baseline system retrieves a large proportion of poisoned documents. The median poison inclusion rate is approximately 0.6, meaning that more than half of the retrieved context is adversarial in typical cases.

The defended pipeline reduces this exposure, lowering the median inclusion rate to approximately 0.4. While poisoned documents are still retrieved, their relative presence is reduced. This demonstrates that the defence does not rely on aggressive document removal alone, but instead partially limits poison dominance while preserving retrieval diversity.

End to end Latency

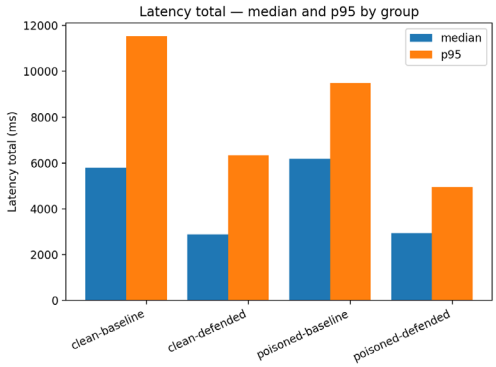

Figure 5: Latency total: median and p95 by group

This experiment measures end-to-end latency, from query submission to final answer or refusal, reporting both median and p95 to capture tail behaviour. Across clean and poisoned corpora, the defended pipeline consistently shows lower latency than the baseline.

Although defences are often expected to add overhead, the difference comes from generation behaviour. The baseline tends to produce long, verbose answers (especially with misleading context) leading to high variance and heavy-tailed latency. In contrast, the defended pipeline frequently stops early when confidence thresholds are not met, returning short refusal or escalation outputs. This reduces overall latency, particularly at the p95 level, and results in more predictable performance.

Retrieval Latency

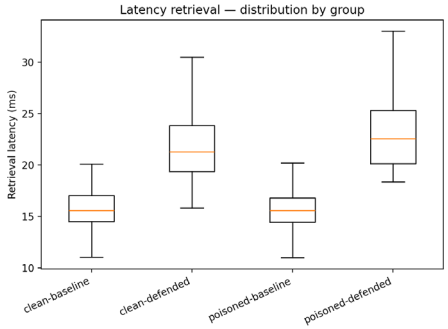

Figure 6: Latency retrieval: distribution by group

To isolate the cost of defensive mechanisms at retrieval time, we separately measure retrieval latency. The defended pipeline introduces a small but consistent increase in retrieval latency compared to the baseline. This overhead corresponds to additional operations such as trust scoring, metadata inspection, and provenance analysis. Importantly, this overhead remains low (on the order of milliseconds) and shows limited variance. Retrieval latency distributions are similar across clean and poisoned corpora, indicating that defensive retrieval costs are stable and predictable.

Generation Latency

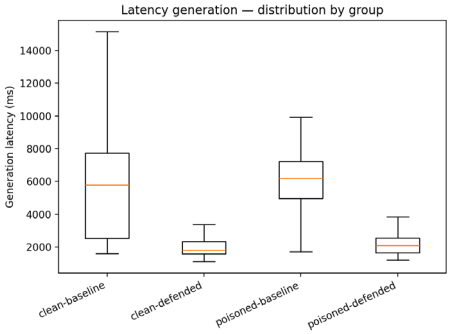

Figure 7: Latency generation: distribution by group

This experiment focuses on generation-stage latency, which dominates total response time in most RAG systems. Baseline generation latency exhibits high variance and extreme outliers, particularly on the poisoned corpus. These cases correspond to long speculative generations triggered by adversarial context. The defended pipeline significantly reduces both median and tail generation latency. By refusing to generate when confidence is insufficient, the system avoids expensive over-generation. This behaviour explains the improved total latency observed in Figure 5.

Operational Decision Behaviour

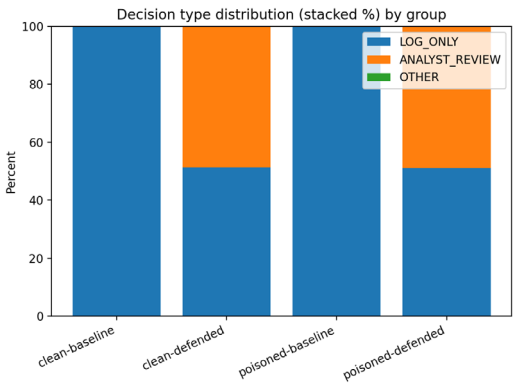

Figure 8: Decision type distribution (stacked %) by group

Beyond answer content, we analyse operational decisions emitted by the system. Each query results in a decision type indicating how the output should be handled. The baseline system always produces LOG_ONLY decisions, regardless of corpus condition. This reflects an implicit assumption that all answers are safe to consume. The defended pipeline produces a mixture of decisions. On both corpora, a substantial fraction of queries are escalated to ANALYST_REVIEW, particularly under poisoned conditions. Queries with sufficient confidence are handled automatically. This behaviour demonstrates that the defended pipeline explicitly models uncertainty and adapts its decisions to evidence quality.

Operational Action Behavior

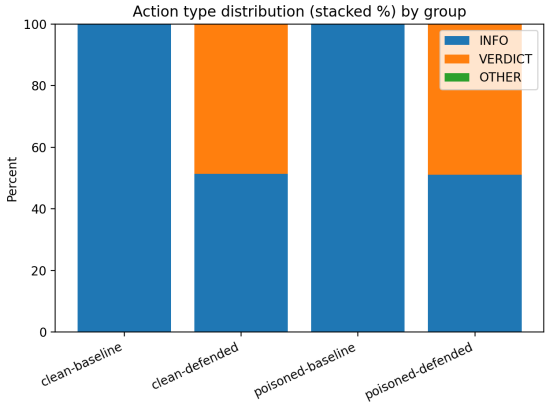

Figure 9: Action type distribution (stacked %) by group

We further examine the action types produced by the system. The baseline system always emits informational actions (INFO), serving as a passive advisory tool. The defended pipeline emits both informational actions and explicit verdicts (VERDICT). This indicates that the system is capable of producing actionable outputs when evidence is strong, while deferring ambiguous cases to human analysts.

Latency-Accuracy Trade-off

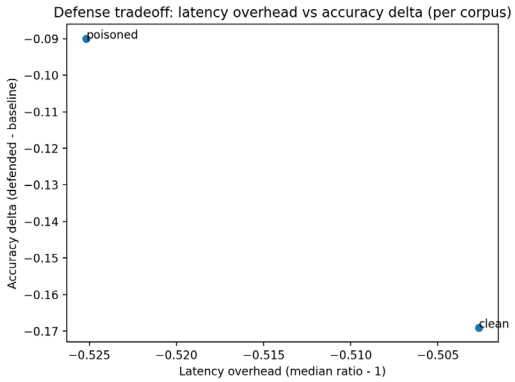

Figure 10: Action type distribution (stacked %) by group

Figure 10 summarizes the trade-off introduced by the defended pipeline by jointly plotting latency overhead and accuracy delta relative to the baseline. For both clean and poisoned corpora, the defended system exhibits lower latency but reduced accuracy under the strict evaluation metric. This reflects a deliberate design choice: the system prioritizes trustworthiness and risk mitigation over answer coverage. In security-critical settings, where incorrect answers can propagate false intelligence or trigger incorrect responses, this trade-off is often preferable.

4. Conclusion

This paper presented RAG-H, a multi-layer defence framework that secures the full RAG pipeline for SOC use by enforcing provenance checks at ingestion, combining semantic relevance with corpus-level credibility at retrieval, and applying context sanitisation and self-verification at generation. We evaluated RAG-H on a CTI

corpus of 300 documents, 90 of which were poisoned, and compared the hardened pipeline against a baseline RAG system under both clean and adversarial conditions.

Returning to our research questions:

RQ1 *- retrieval is the stage most vulnerable to poisoning, because similarity-only ranking treats adversarial documents optimised for query patterns as legitimate evidence.*

RQ2 *- layered controls that combine ingestion-time provenance with retrieval-time credibility signals produce the largest reduction in the Poison Impact Rate, with generation-time verification adding a smaller but non-trivial margin, added latency remains within acceptable operational bounds.*

RQ3 *- RAG-H's trust signals translate into measurable operational behaviour, escalating low-confidence cases to analyst review and producing structured verdicts when evidence is strong, giving analysts an auditable trail of why each output was generated.*

5. Limitations and Future Work

This study is a controlled proof of concept: the corpus is small relative to a production CTI knowledge base, the attack set does not cover end-to-end supply-chain compromise, and analyst feedback was simulated. Three directions are immediate priorities: scaling the evaluation to larger corpora including live telemetry and operational Security Information and Event Management (SIEM) feeds, validating the framework with practising SOC analysts, and extending the threat model to adaptive adversaries that observe and adjust to the defence.

Systems that always return an answer appear more capable on the surface but silently pass poisoned context and hallucinated conclusions downstream. RAG-H trades a measurable loss in answer coverage for a safety property that aligns with how SOCs actually operate, where incorrect confidence is more costly than a delayed decision. Trustworthy AI in security will not be achieved by scaling language models alone, but by pipelines that are auditable by design: systems that know when not to decide and can show analysts why.

Ethics Declaration: This research did not require ethical approval, as it involved no personal data or live operational systems. All experiments used publicly available or generated data.

AI Declaration: AI tools assisted with code generation and grammar but all outputs were reviewed by the author, who retains full responsibility for the research design, analysis, and content.

References

Chen, Z., Xiang, Z., Xiao, C., Song, D., & Li, B. (2024). AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases. [online] arXiv.org. Available at: https://arxiv.org/abs/2407.12784 [Accessed 29 Oct. 2025].

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M. and Wang, H. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. [online] arXiv.org. Available at: https://arxiv.org/abs/2312.10997 [Accessed 12 Oct. 2025].

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T. and Fritz, M. (2023). Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23), pp. 79–90. [online] ACM Digital Library. Available at: https://dl.acm.org/doi/10.1145/3605764.3623985 [Accessed 11 Nov. 2025].

Habibzadeh, A., Rashid, S.F. and Sadrossadat, S.A. (2025). Large Language Models for Security Operations Centers: A Comprehensive Survey. [online] arXiv.org. Available at: https://arxiv.org/abs/2509.10858 [Accessed 10 Oct. 2025].

Husari, G., Al-Shaer, E., Ahmed, M., Chu, B. and Niu, X. (2017). TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources. In Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC 2017), pp. 103–115. [online] ACM Digital Library. Available at: https://dl.acm.org/doi/10.1145/3134600.3134646 [Accessed 4 Dec. 2025].

Jalalvand, F., Chhetri, M.B., Nepal, S. and Paris, C. (2025). Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities. ACM Computing Surveys. [online] ACM Digital Library. Available at: https://dl.acm.org/doi/10.1145/3723158 [Accessed 7 Sept. 2025].

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A. and Fung, P. (2023). Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12), Article 248. [online] ACM Digital Library. Available at: https://dl.acm.org/doi/10.1145/3571730 [Accessed 6 Oct. 2025].

Jia, A., Ramesh, A., Shamsi, Z., Zhang, D. and Liu, A. (2025). RAGRank: Using PageRank to Counter Poisoning in CTI LLM Pipelines. [online] arXiv.org. Available at: https://arxiv.org/html/2510.20768v1 [Accessed 23 Nov. 2025].

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W.-T. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural

Language Processing (EMNLP), pp. 6769–6781. [online] ACL Anthology. Available at: https://aclanthology.org/2020.emnlp-main.550/ [Accessed 20 Sept. 2025].

Lekssays, A., Shukla, U., Sencar, H.T. and Parvez, M.R. (2025). TECHNIQUERAG: Retrieval-Augmented Generation for Adversarial Technique Annotation in Cyber Threat Intelligence Text. Findings of ACL 2025, pp. 20913–20926. [online] ACL Anthology. Available at: https://aclanthology.org/2025.findings-acl.1076.pdf [Accessed 15 Nov. 2025].

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S. and Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp. 9459–9474. [online] arXiv.org. Available at: https://arxiv.org/abs/2005.11401 [Accessed 9 Sept. 2025].

Mezzi, E., Massacci, F. and Tuma, K. (2025). Large Language Models Are Unreliable for Cyber Threat Intelligence. In Proceedings of the 20th International Conference on Availability, Reliability and Security (ARES 2025), pp. 343–364. [online] Springer. Available at: https://doi.org/10.1007/978-3-032-00627-1_17 [Accessed 9 Sept. 2025].

Perez, F. and Ribeiro, I. (2022). Ignore Previous Prompt: Attack Techniques For Language Models. ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). [online] arXiv.org. Available at: https://arxiv.org/abs/2211.09527 [Accessed 8 Oct. 2025].

Qian, H., Li, B. and Wang, Q. (2025). ClaimTrust: Propagation Trust Scoring for RAG Systems. [online] arXiv.org. Available at: https://arxiv.org/abs/2503.10702 [Accessed 30 Nov. 2025].

RoyChowdhury, A., Luo, M., Sahu, P., Banerjee, S., & Tiwari, M. (2024). ConfusedPilot: Confused Deputy Risks in RAG-based LLMs. [online] arXiv.org. Available at: https://arxiv.org/abs/2408.04870 [Accessed 4 Dec. 2025].

SANS 2025 AI Survey: Measuring AI Impact on Security - Three Years Later. [online] SANS.org. Available at: https://www.sans.org/white-papers/sans-2025-ai-survey-measuring-ai-impact-security-three-years-later [Accessed 15 Oct. 2025].

Singh, R., Tariq, S., Jalalvand, F., Chhetri, M.B., Nepal, S., Paris, C. and Lochner, M. (2025). LLMs in the SOC: An Empirical Study of Human-AI Collaboration in Security Operations Centres. [online] arXiv.org. Available at: https://arxiv.org/abs/2508.18947 [Accessed 29 Oct. 2025].

Xiang, C., Wu, T., Zhong, Z., Wagner, D., Chen, D. and Mittal, P. (2024). Certifiably Robust RAG against Retrieval Corruption. [online] arXiv.org. Available at: https://arxiv.org/abs/2405.15556 [Accessed 3 Oct. 2025].

Xiang, C., Wu, T., Zhong, Z., Wagner, D., Chen, D., & Mittal, P. (2024). Certifiably Robust RAG against Retrieval Corruption. [online] arXiv.org. Available at: https://arxiv.org/abs/2405.15556 [Accessed 1 Oct. 2025].

Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y. and Wang, H. (2024). Large Language Models for Cyber Security: A Systematic Literature Review. [online] arXiv.org. Available at: https://arxiv.org/abs/2405.04760 [Accessed 21 Oct. 2025].

Zhou, H., Lee, K.-H., Zhan, Z., Chen, Y., Li, Z., Wang, Z., Haddadi, H. and Yilmaz, E. (2025). TrustRAG: Enhancing Robustness and Trustworthiness in Retrieval-Augmented Generation. [online] arXiv.org. Available at: https://arxiv.org/abs/2501.00879 [Accessed 5 Oct. 2025].

Zou, W., Geng, R., Wang, B., & Jia, J. (2025). PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. In Proceedings of the 34th USENIX Security Symposium. [online] USENIX. Available at: https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag [Accessed 10 Oct. 2025].