WCRG Global Research
WORLD COMET RESEARCH GROUP

International Journal of Applied Data Science & Modern Computing
Vol. 9, No. 3, 2026
Doi: 10.67228/30713498/IJADSMC-V9I3P101
PP. 01-14

Original Article

Securing Retrieval-Augmented Generation Systems: Threat Modeling, Data Poisoning, Prompt Injection, and Access Control for Enterprise AI

Santosh Kumar Jadala

Cyber Security & Business Analysis Specialist, Independent Researcher, USA

Received: 06-04-2026 Revised: 06-25-2026 Accepted: 07-01-2026 Published: 07-04-2026

ABSTRACT

Retrieval-Augmented Generation (RAG) has emerged as an important architecture for enhancing large language models with external enterprise knowledge, enabling more accurate, contextual, and domain-specific responses. However, integrating retrieval mechanisms with generative models introduces a distinct security attack surface involving untrusted data sources, vector databases, retrieval pipelines, prompts, and access-control mechanisms. This study examines the security risks affecting enterprise RAG systems, with particular attention to data poisoning, prompt injection, unauthorized knowledge retrieval, information leakage, and inadequate access control. It develops a threat-oriented security framework that maps adversarial activities across the RAG lifecycle, from data ingestion and embedding generation to retrieval, prompt construction, and response generation. The proposed approach combines secure data provenance, poisoning detection, prompt isolation, retrieval-level authorization, role-based and attribute-based access controls, Zero Trust principles, continuous monitoring, and auditability. The study further emphasizes defense-in-depth as a necessary strategy for protecting sensitive enterprise knowledge while preserving the operational benefits of RAG. By integrating established cybersecurity principles with emerging large language model threats, the research provides a structured foundation for securing enterprise RAG deployments. The study concludes that effective RAG security requires coordinated controls across data, retrieval, generation, identity, and governance layers.

KEYWORDS

Retrieval-Augmented Generation, RAG Security, Prompt Injection, Data Poisoning, Access Control, Enterprise AI, Zero Trust, Large Language Models.

ISSN: 3071-3498
PP: 01-14.

1. INTRODUCTION

Retrieval-Augmented Generation (RAG) has emerged as an important architecture for improving the reliability and contextual relevance of large language models by combining generative capabilities with information retrieved from external knowledge sources. Rather than depending exclusively on knowledge encoded during model training, RAG systems retrieve relevant documents or passages from external repositories and incorporate them into the model context before generating a response. Lewis et al. (2020) demonstrated that combining parametric language models with non-parametric retrieval mechanisms can substantially improve performance on knowledge-intensive natural language processing tasks. Dense retrieval approaches have further improved the ability of systems to identify semantically relevant information from large document collections (Karpukhin et al., 2020). These capabilities make RAG particularly attractive to enterprises seeking to connect generative AI systems with internal policies, technical documentation, customer records, financial information, operational knowledge, and other proprietary data without continually retraining foundation models.

Despite these advantages, enterprise RAG introduces a broader and more complex security attack surface than conventional language-model applications. A typical RAG pipeline contains several interconnected components, including data ingestion mechanisms, document stores, embedding models, vector databases, retrieval engines, prompt construction processes, generative models, and user-facing applications. Compromise at any of these stages can influence the information retrieved or the responses generated. Attackers may inject malicious documents into trusted knowledge repositories, manipulate retrieval results, embed adversarial instructions within external content, obtain sensitive information through unauthorized queries, or exploit weaknesses in authentication and authorization controls. Indirect prompt injection is especially concerning because malicious instructions contained within retrieved documents may influence the behavior of an LLM without being explicitly supplied by the legitimate user (Greshake et al., 2023). More broadly, language models have demonstrated vulnerabilities to prompt manipulation, data leakage, backdoors, and adversarial inputs (Perez et al., 2022; Zhao et al., 2023).

The central research problem addressed in this study is that enterprise adoption of RAG is progressing faster than the development of integrated security approaches capable of protecting the complete retrieval-generation lifecycle. Existing cybersecurity mechanisms often secure databases, applications, identities, and networks separately, while emerging LLM security research frequently focuses on individual attacks such as prompt injection or model poisoning. This fragmentation leaves enterprises without a unified method for understanding how threats propagate across data ingestion, retrieval, prompt construction, generation, and access-control layers.

A significant research gap therefore exists in integrating traditional enterprise security principles with emerging RAG-specific threats. Studies on data poisoning provide useful defenses against corrupted training or reference data (Steinhardt et al., 2017; Zeng et al., 2023), while research on privacy attacks demonstrates that language models can expose sensitive information under certain

ISSN: 3071-3498
PP: 01-14.

conditions (Carlini et al., 2021; Lukas et al., 2023). Similarly, established role-based and attribute-based access-control models provide mechanisms for restricting resource access (Sandhu et al., 1996; Servos & Osborn, 2017). However, these research streams have rarely been combined into a comprehensive security model specifically designed for enterprise RAG architectures.

Accordingly, this study aims to identify and classify major threats affecting enterprise RAG systems, develop a structured threat model across the RAG pipeline, examine data poisoning and prompt injection risks, evaluate access-control requirements, and propose a defense-in-depth security framework. The research asks: What are the principal attack surfaces within enterprise RAG architectures? How can data poisoning and prompt injection compromise retrieval and generation processes? How should access control be enforced across enterprise knowledge repositories and retrieval operations? What combination of preventive, detective, and governance controls can provide effective protection?

The study contributes by integrating RAG security, AI threat modeling, access control, Zero Trust principles, data integrity, and continuous monitoring within a unified framework. Consistent with the NIST AI Risk Management Framework, effective enterprise AI security requires risk identification, governance, measurement, and continuous management rather than isolated technical safeguards (NIST, 2023). The proposed approach therefore provides both a theoretical foundation and a practical security architecture for organizations seeking to deploy RAG while protecting confidentiality, integrity, availability, accountability, and trust.

2. LITERATURE REVIEW AND SECURITY FOUNDATIONS

2.1. Architecture and Operation of RAG Systems

Retrieval-Augmented Generation (RAG) combines information retrieval with generative language modeling to improve the factual grounding and domain relevance of generated responses. Instead of relying entirely on knowledge stored in model parameters, a RAG system retrieves relevant external information and supplies it to the language model as contextual evidence before response generation. Lewis et al. (2020) established this architecture by combining a parametric sequence-to-sequence model with a non-parametric retrieval component, demonstrating improved performance on knowledge-intensive natural language processing tasks. Dense Passage Retrieval further strengthened this approach by using learned vector representations to retrieve semantically relevant passages from large collections (Karpukhin et al., 2020).

In an enterprise setting, the typical RAG pipeline begins with document ingestion, followed by preprocessing, text segmentation, embedding generation, and storage within a vector database. When a user submits a query, the system generates a corresponding query embedding, identifies similar document chunks, constructs an augmented prompt containing the retrieved information, and sends the prompt to a large language model for response generation. Retrieval-augmented language model pre-training has shown that access to external knowledge can enhance language-model performance without requiring all factual information to remain embedded within model parameters (Guu et al.,

ISSN: 3071-3498
PP: 01-14.

2020). Izacard and Grave (2021) similarly demonstrate the effectiveness of combining passage retrieval with generative models for knowledge-intensive question answering.

However, each additional RAG component creates another potential security boundary. Enterprise RAG must therefore protect not only the language model, but also the ingestion pipeline, embedding processes, vector database, retrieval mechanisms, prompts, identities, and generated outputs.

2.2. Enterprise AI Security and Trust Requirements

Enterprise RAG applications frequently access confidential resources such as internal policies, intellectual property, employee information, customer records, financial documents, and technical knowledge. Security must therefore preserve confidentiality, integrity, availability, authenticity, and accountability across the complete retrieval-generation lifecycle.

The NIST Artificial Intelligence Risk Management Framework emphasizes that trustworthy AI requires continuous governance, risk identification, measurement, and management throughout the AI lifecycle (National Institute of Standards and Technology [NIST], 2023). For RAG systems, trust additionally depends on whether retrieved evidence originates from authorized and verifiable sources. A technically accurate language model cannot be considered trustworthy if attackers can alter the external knowledge on which its responses depend.

Privacy is another important concern. Research demonstrates that language models can unintentionally expose memorized or personally identifiable information under certain attack conditions (Carlini et al., 2021; Lukas et al., 2023). Connecting language models to enterprise repositories increases the importance of controlling what information can be retrieved and which users are permitted to access it.

2.3. Threat Modeling for LLM and RAG Pipelines

Threat modeling provides a systematic method for identifying valuable assets, trust boundaries, adversaries, attack paths, and potential consequences. For enterprise RAG, critical assets include source documents, embeddings, vector indexes, credentials, prompts, retrieved context, generated outputs, and model interfaces. Threats may originate from external attackers, malicious insiders, compromised data sources, unauthorized users, or manipulated third-party content. Weidinger et al. (2022) demonstrate that language-model risks are multidimensional and include information hazards, discrimination, malicious use, misinformation, and security concerns. RAG expands these risks because the model is dynamically influenced by external information at inference time.

A useful RAG threat model must therefore evaluate attacks across several layers: data ingestion, storage, retrieval, prompt construction, generation, and user access. This layered approach makes it possible to understand how a compromise at one stage can propagate through subsequent stages.

ISSN: 3071-3498
PP: 01-14.

2.4. Data Poisoning and Knowledge-Base Manipulation

Data poisoning represents a major integrity threat to RAG systems. Unlike conventional model poisoning, an attacker may not need to modify model parameters directly. If malicious content is inserted into a searchable enterprise knowledge base, the retrieval mechanism may later select it as trusted context.

Research has established that machine-learning systems can be influenced through strategically poisoned datasets (Steinhardt et al., 2017; Suciu et al., 2018). More recent work demonstrates that language models remain vulnerable to poisoning during instruction tuning and other forms of malicious data manipulation (Wan et al., 2023). Zeng et al. (2023) proposed Meta-Sift as a method for identifying clean subsets within poisoned datasets, illustrating the need for mechanisms capable of distinguishing legitimate from adversarial information.

In RAG environments, knowledge-base poisoning may involve inserting fabricated documents, modifying legitimate files, manipulating metadata, or creating adversarial passages optimized for retrieval. Attackers could therefore influence model responses without directly compromising the underlying LLM.

2.5. Prompt Injection and Indirect Prompt Injection

Prompt injection exploits the difficulty language models have in consistently distinguishing trusted system instructions from untrusted natural-language content. An attacker may provide instructions designed to override system behavior, reveal confidential information, or manipulate the generated response.

Schulhoff et al. (2023) demonstrated widespread vulnerabilities to prompt hacking techniques across language models. More critically for RAG systems, Greshake et al. (2023) showed that indirect prompt injection can compromise LLM-integrated applications through malicious instructions embedded in external resources. In a RAG pipeline, a poisoned document could contain hidden or explicit instructions such as asking the model to ignore previous policies, disclose confidential context, or perform unintended actions.

This creates a fundamental security challenge because retrieved documents serve simultaneously as information and model input. Consequently, RAG systems require separation between instructions and retrieved content, input sanitization, content provenance, contextual filtering, and output validation.

2.6. Access Control, Authentication, and Data Confidentiality

Access control is essential because successful authentication alone should not grant unrestricted access to all enterprise knowledge. Role-Based Access Control assigns permissions according to organizational roles and remains widely applicable to structured enterprise environments (Sandhu et al., 1996). Attribute-Based Access Control provides finer-grained authorization based on factors such

ISSN: 3071-3498
PP: 01-14.

as user identity, department, resource classification, context, and environmental conditions (Servos & Osborn, 2017).

For RAG systems, authorization should be enforced during retrieval rather than only at the application interface. A user should retrieve only document chunks that the user is authorized to access. Zero Trust Architecture strengthens this principle by rejecting implicit trust and requiring continuous verification of subjects, resources, and access requests (Rose et al., 2020). NIST's cloud-native Zero Trust guidance further supports fine-grained policy enforcement across distributed environments (Chandramouli & Butcher, 2023).

2.7. Limitations of Existing RAG Security Approaches

Existing research provides strong individual solutions for poisoning, prompt attacks, privacy, access control, and AI risk management, but these areas remain largely fragmented. Traditional access-control research does not directly address adversarial natural-language instructions, while prompt-injection research often focuses on model behavior without fully considering enterprise identity and data-governance requirements. Similarly, poisoning defenses developed for model training do not completely address dynamically changing RAG knowledge repositories.

The principal limitation is therefore the absence of an integrated defense model spanning ingestion, retrieval, prompt construction, model generation, authorization, monitoring, and governance. Secure enterprise RAG requires defense-in-depth rather than isolated controls. This limitation motivates the development of a comprehensive threat model and security architecture in the following section.

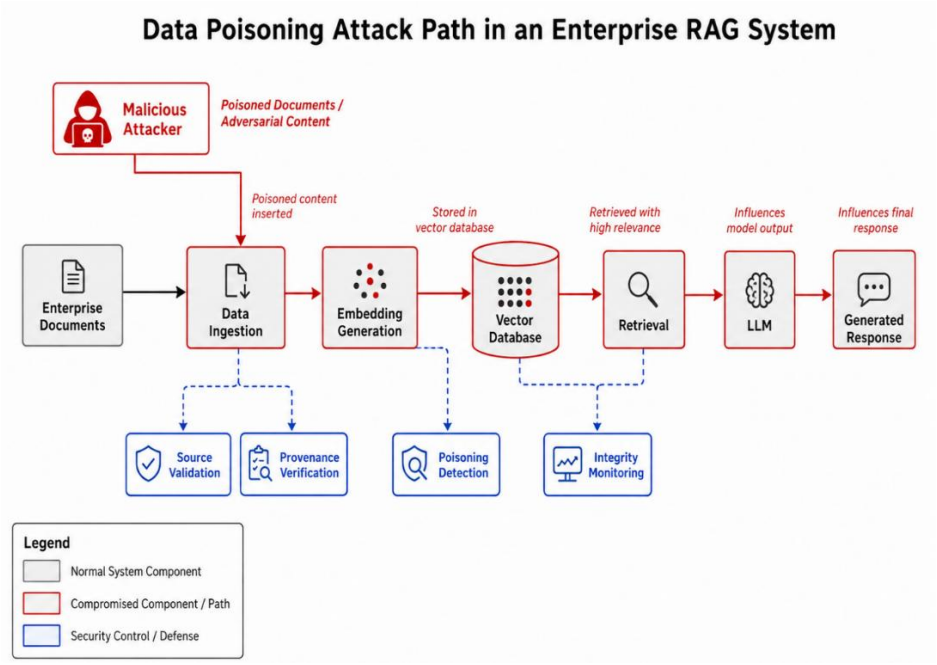

Fig 1 - Data Poisoning and Knowledge-Base Manipulation Pathway in an Enterprise RAG Architecture

ISSN: 3071-3498
PP: 01-14.

Data poisoning and knowledge-base manipulation pathway in an enterprise RAG system, illustrating how malicious content can enter the retrieval pipeline and influence generated responses.

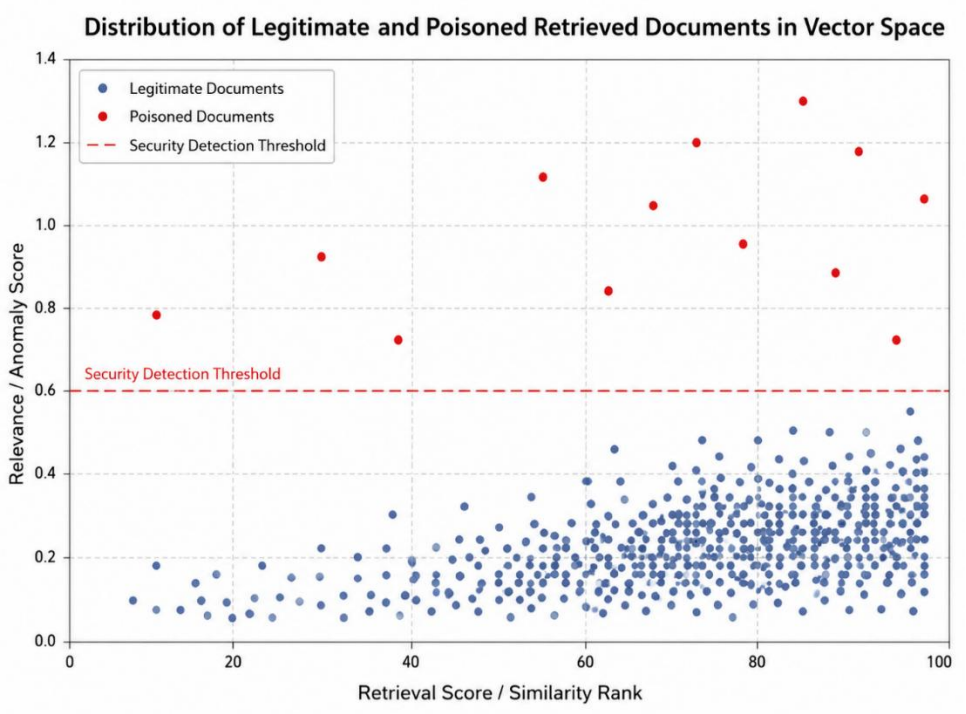

Fig 2 - Conceptual Distribution of Legitimate and Poisoned Documents Based On Retrieval Similarity and Anomaly Detection Thresholds

Conceptual distribution of legitimate and poisoned documents in vector space, showing how anomaly detection thresholds can support identification of suspicious retrieved content.

3. THREAT MODEL AND RAG ATTACK SURFACE

An enterprise Retrieval-Augmented Generation system consists of several interconnected components that collectively expand the security boundary beyond the underlying large language model. Core assets include enterprise documents, source repositories, embedding models, vector databases, retrieval indexes, authentication credentials, access-control policies, user prompts, retrieved context, model outputs, system instructions, and audit records. These assets vary in sensitivity and therefore require different levels of protection. In particular, enterprise knowledge repositories may contain intellectual property, financial information, customer records, internal policies, employee data, and other confidential information. Because RAG systems dynamically retrieve such information at inference time, the integrity and confidentiality of the retrieval layer become as important as the security of the generative model itself (Lewis et al., 2020; NIST, 2023).

The threat model considers several adversary profiles. External attackers may attempt to manipulate public or externally connected information sources, exploit exposed APIs, or construct malicious queries. Insiders may possess legitimate credentials but intentionally misuse authorized

ISSN: 3071-3498
PP: 01-14.

access to retrieve sensitive information. Compromised third-party content providers represent another threat because malicious content may enter the RAG pipeline without directly breaching enterprise infrastructure. Adversarial goals can include disclosure of confidential data, manipulation of generated responses, disruption of retrieval accuracy, unauthorized access, persistence through poisoned knowledge, and circumvention of organizational security controls. Threat modeling must therefore consider both direct system compromise and indirect manipulation of the information consumed by the model.

The retrieval and vector database layers create distinctive attack surfaces. Vector databases store numerical representations of enterprise information that enable semantic retrieval. Although embeddings are different from original documents, unauthorized access to the retrieval infrastructure can still expose sensitive relationships, metadata, or underlying content. Attackers may also manipulate document embeddings, similarity rankings, metadata, or retrieval filters to increase the likelihood that malicious information is returned. Because dense retrieval systems rank documents according to semantic similarity, deliberately constructed content may be positioned to dominate retrieval for targeted queries (Karpukhin et al., 2020).

Data poisoning represents a particularly significant integrity threat. An attacker who can insert, alter, or influence documents in an enterprise knowledge base may indirectly manipulate subsequent model responses. Poisoning may involve fabricated documents, subtle changes to legitimate content, malicious metadata, backdoor triggers, or adversarial passages designed to obtain high retrieval relevance. Research has demonstrated that machine-learning and language-model systems remain vulnerable to poisoning and backdoor attacks (Steinhardt et al., 2017; Wan et al., 2023; Zhao et al., 2023). In a RAG context, this risk is amplified because knowledge repositories may be continuously updated, allowing malicious content to affect responses without retraining the underlying model.

Prompt injection further extends the attack surface. Direct prompt injection occurs when users deliberately construct instructions intended to override system policies. Indirect prompt injection occurs when malicious instructions are embedded within documents later retrieved as context. Greshake et al. (2023) demonstrated that such external content can influence LLM-integrated applications, creating risks including instruction hijacking, sensitive-data disclosure, and unintended actions. Retrieval manipulation and prompt injection may also operate together, where attackers first increase the retrieval probability of a malicious document and then use embedded instructions to influence generation.

Unauthorized knowledge access and information leakage represent confidentiality threats. Authentication determines who a user is, but authorization must additionally determine which documents, chunks, or knowledge domains that user may retrieve. RBAC and ABAC provide established mechanisms for assigning access according to roles and contextual attributes (Sandhu et al., 1996; Servos & Osborn, 2017). Zero Trust principles further require continuous verification and

ISSN: 3071-3498
PP: 01-14.

least-privilege access rather than implicit trust based solely on network position or initial authentication (Rose et al., 2020).

Table 1: Threat Classification and Risk Assessment for Enterprise RAG Systems

Threat	Primary Target	Attack Method	Potential Impact	Risk Level	Key Controls
Knowledge-base poisoning	Enterprise data repository	Malicious document insertion or modification	Manipulated retrieval and inaccurate responses	High	Provenance validation, integrity checking
Retrieval manipulation	Vector database/retriever	Embedding or ranking manipulation	Malicious content prioritized	High	Retrieval monitoring, anomaly detection
Direct prompt injection	Prompt interface	Adversarial user instructions	Policy bypass and unintended output	High	Prompt filtering, instruction separation
Indirect prompt injection	Retrieved documents	Hidden malicious instructions	LLM behavior manipulation	Critical	Content isolation, sanitization, validation
Unauthorized retrieval	Knowledge repository	Privilege abuse or weak authorization	Confidential information exposure	Critical	RBAC, ABAC, least privilege
Information leakage	LLM output/context	Extraction or inference attacks	Disclosure of sensitive enterprise data	Critical	Output filtering, access controls, monitoring
Credential misuse	Authentication layer	Stolen or abused credentials	Unauthorized system and data access	High	MFA, continuous authentication, Zero Trust

The proposed risk assessment framework classifies threats according to likelihood, exploitability, affected asset, and potential organizational impact. Risks involving sensitive-data disclosure or persistent manipulation of retrieval results should receive the highest priority because they can compromise both confidentiality and decision integrity. This threat-oriented analysis provides the foundation for developing a defense-in-depth enterprise RAG security framework that integrates preventive, detective, and corrective controls across the complete retrieval-generation lifecycle.

4. SECURE ENTERPRISE RAG FRAMEWORK AND MITIGATION STRATEGIES

The proposed Secure Enterprise Retrieval-Augmented Generation framework adopts a defense-in-depth approach in which security controls are distributed across the complete RAG lifecycle rather than concentrated at the language-model interface. This approach recognizes that enterprise RAG security depends on the combined protection of source data, ingestion pipelines, embeddings, vector databases, retrieval mechanisms, prompts, access policies, model outputs, and operational

ISSN: 3071-3498
PP: 01-14.

monitoring. Consistent with the NIST Artificial Intelligence Risk Management Framework, effective protection requires governance, continuous risk assessment, measurement, and security controls that operate throughout the AI system lifecycle (NIST, 2023). The framework therefore combines preventive, detective, and corrective mechanisms to reduce the likelihood that compromise at a single layer propagates throughout the RAG pipeline.

Security begins at data ingestion. Documents entering enterprise knowledge repositories should undergo source authentication, integrity verification, content inspection, metadata validation, and provenance recording before embedding generation. Trusted-source allowlists, document hashing, digital signatures, version histories, and provenance metadata can help establish whether retrieved information originated from an approved source. Such controls are particularly important because poisoned content can influence downstream model behavior without modifying the underlying model itself. Research on data poisoning demonstrates that carefully manipulated data can significantly affect machine-learning behavior, reinforcing the importance of validating information before it becomes available to retrieval systems (Steinhardt et al., 2017; Wan et al., 2023).

Poisoning detection should continue after ingestion through retrieval-integrity controls. Organizations can monitor unusual embedding distributions, abnormal similarity scores, sudden changes in document retrieval frequency, duplicate adversarial passages, and unexpected metadata modifications. Suspicious documents should be quarantined for human or automated review before they are supplied to the language model. Techniques for identifying cleaner subsets within potentially poisoned datasets provide useful foundations for this type of integrity monitoring (Zeng et al., 2023). Retrieval logs should additionally record which documents influenced individual responses, enabling investigators to trace problematic outputs to their underlying evidence.

Prompt injection requires controls at both user-input and retrieved-context layers. Direct prompts should be inspected for attempts to override system instructions, request unauthorized information, or manipulate application behavior. More importantly, retrieved documents should be treated as untrusted data rather than authoritative instructions. Greshake et al. (2023) demonstrate that indirect prompt injection can exploit external content consumed by LLM-integrated applications. The framework therefore recommends separating system instructions, user commands, and retrieved evidence; sanitizing suspicious content; restricting model tool permissions; and validating generated outputs before sensitive actions are executed.

Access control represents another critical layer. Role-Based Access Control can restrict information according to organizational responsibilities, while Attribute-Based Access Control provides more granular decisions based on user attributes, document sensitivity, department, purpose, location, or other contextual factors (Sandhu et al., 1996; Servos & Osborn, 2017). Authorization should be applied directly during retrieval so that unauthorized document chunks never enter the model context. This approach supports least privilege and aligns with Zero Trust principles that require continuous verification rather than implicit trust (Rose et al., 2020; Chandramouli & Butcher, 2023).

WCRG Global Research
ISSN: 3071-3498
PP: 01-14.

Continuous monitoring, logging, and auditability complete the framework. Logs should capture authentication events, retrieval queries, document identifiers, policy decisions, detected prompt attacks, model responses, administrative modifications, and security overrides. An incident-response process should support rapid containment of poisoned documents, credential revocation, vector-index rebuilding, forensic analysis, and recovery. Framework effectiveness can be evaluated through metrics such as poisoning detection accuracy, prompt-injection detection rate, false-positive rate, unauthorized retrieval prevention rate, sensitive-data leakage rate, retrieval integrity, and incident-detection time**.** Validation should combine controlled adversarial testing, red-team exercises, and simulated enterprise attack scenarios (Perez et al., 2022). Overall, the framework establishes that secure enterprise RAG requires coordinated protection across data, retrieval, identity, prompt, model, and governance layers rather than dependence on any single defensive mechanism.

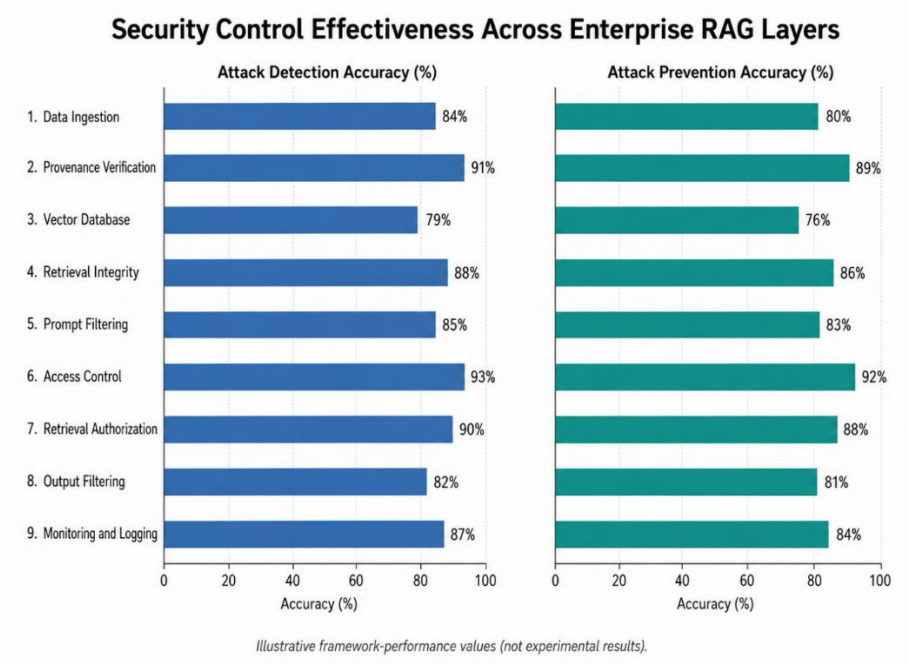

Fig 3 - Illustrative Comparison of Attack Detection and Prevention Effectiveness across the Security Layers of the Proposed Enterprise RAG Framework

5. DISCUSSION

The findings of this study demonstrate that securing Retrieval-Augmented Generation systems requires a broader security perspective than protecting the underlying large language model alone. RAG architectures introduce interconnected attack surfaces across data ingestion, embedding generation, vector storage, retrieval, prompt construction, model inference, and access control. Consequently, weaknesses within any single component can propagate through the pipeline and undermine the confidentiality, integrity, and reliability of enterprise AI outputs. This observation extends earlier work on RAG architectures by emphasizing that external knowledge integration creates both operational value and additional security dependencies (Lewis et al., 2020).

ISSN: 3071-3498
PP: 01-14.

Data poisoning and malicious document injection represent particularly significant threats because attackers may influence generated responses without modifying model parameters. Existing poisoning research demonstrates that carefully manipulated information can alter machine-learning behavior (Steinhardt et al., 2017; Wan et al., 2023). Within RAG environments, provenance verification, integrity monitoring, anomaly detection, and controlled ingestion therefore become essential safeguards. Prompt injection presents an equally important challenge. Indirect prompt injection is particularly dangerous because malicious instructions embedded in retrieved content can be interpreted by the model as legitimate commands (Greshake et al., 2023).

The proposed defense-in-depth framework addresses these risks by integrating technical and organizational controls. RBAC, ABAC, retrieval-level authorization, and Zero Trust principles help ensure that users retrieve only information they are authorized to access (Sandhu et al., 1996; Servos & Osborn, 2017; Rose et al., 2020). Continuous monitoring and auditability further support detection and investigation of abnormal activities.

The study therefore suggests that enterprise RAG security should be treated as a continuous risk-management process rather than a one-time configuration activity. Consistent with NIST (2023), organizations should continuously identify, measure, monitor, and manage AI-related risks while adapting controls as threat techniques evolve.

6. CONCLUSION

Retrieval-Augmented Generation provides enterprises with an effective means of combining large language models with current, domain-specific, and proprietary knowledge. However, its reliance on external data and retrieval infrastructure creates security risks that conventional language-model protections alone cannot adequately address. This study identified major threats affecting enterprise RAG systems, including knowledge-base poisoning, malicious document injection, retrieval manipulation, direct and indirect prompt injection, unauthorized knowledge access, and sensitive-information leakage.

The proposed defense-in-depth framework integrates secure data ingestion, provenance verification, poisoning detection, retrieval integrity controls, prompt isolation, RBAC, ABAC, Zero Trust, least-privilege authorization, continuous monitoring, logging, and incident response. Collectively, these controls establish security boundaries across the entire retrieval-generation lifecycle.

The central conclusion is that secure RAG deployment requires coordinated protection of data, retrieval, identity, prompts, model interactions, and governance mechanisms**.** No individual security control is sufficient against the diverse threats affecting enterprise RAG architectures. Organizations should therefore adopt continuous threat modeling, layered security controls, adversarial testing, and auditable governance practices to preserve confidentiality, integrity, accountability, and trust while realizing the operational benefits of Retrieval-Augmented Generation.

ISSN: 3071-3498
PP: 01-14.

REFERENCES

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[2] Kunaparaju, C. (2025). AI-Driven Cyber Defense Systems: Strengthening National Security through Intelligent Threat Prediction and Response. Algora, 2(1), 1-30.

[3] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[4] Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. *Proceedings of the 37th International Conference on Machine Learning, 119*, 3929–3938. https://proceedings.mlr.press/v119/guu20a.html

[5] Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 874–880. https://doi.org/10.18653/v1/2021.eacl-main.74

[6] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*. https://doi.org/10.1145/3605764.3623985

[7] Schulhoff, S., Pinto, J., Khan, A., Bouchard, L. F., Si, C., Anati, S., Tagliabue, V., Kost, A., Carnahan, C., & Boyd-Graber, J. (2023). Ignore this title and HackAPrompt: Exposing systemic vulnerabilities of LLMs through a global prompt hacking competition. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 4945–4977. https://doi.org/10.18653/v1/2023.emnlp-main.302

[8] Yip, D. W., Esmradi, A., & Chan, C. F. (2023). A novel evaluation framework for assessing resilience against prompt injection attacks in large language models. *2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 1–5. https://doi.org/10.1109/CSDE59766.2023.10487667

[9] Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2019). Universal adversarial triggers for attacking and analyzing NLP. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2153–2162. https://doi.org/10.18653/v1/D19-1221

[10] Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., & Irving, G. (2022). Red teaming language models with language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3419–3448. https://doi.org/10.18653/v1/2022.emnlp-main.225

[11] Zhao, S., Wen, J., Luu, A., Zhao, J., & Fu, J. (2023). Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12303–12317. https://doi.org/10.18653/v1/2023.emnlp-main.757

[12] Li, L., Song, D., Li, X., Zeng, J., Ma, R., & Qiu, X. (2021). Backdoor attacks on pre-trained models by layerwise weight poisoning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3023–3032. https://doi.org/10.18653/v1/2021.emnlp-main.241

[13] Qi, F., Chen, Y., Zhang, X., Li, M., Liu, Z., & Sun, M. (2021). Mind the style of text! Adversarial and backdoor attacks based on text style transfer. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4569–4580. https://doi.org/10.18653/v1/2021.emnlp-main.374

[14] Gan, L., Li, J., Zhang, T., Li, X., Meng, Y., Wu, F., Yang, Y., Guo, S., & Fan, C. (2022). Triggerless backdoor attack for NLP tasks with clean labels. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2942–2952. https://doi.org/10.18653/v1/2022.naacl-main.214

[15] Zeng, Y., Pan, M., Jahagirdar, H., Jin, M., Lyu, L., & Jia, R. (2023). Meta-Sift: How to sift out a clean subset in the presence of data poisoning? *32nd USENIX Security Symposium (USENIX Security 23)*, 1667–1684. https://www.usenix.org/conference/usenixsecurity23/presentation/zeng

[16] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. *30th USENIX Security Symposium*

ISSN: 3071-3498
PP: 01-14.

(USENIX Security 21), 2633–2650. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[17] National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1

[18] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *Computer, 29*(2), 38–47. https://doi.org/10.1109/2.485845

[19] Servos, D., & Osborn, S. L. (2017). Current research and open problems in attribute-based access control. *ACM Computing Surveys, 49*(4), Article 65, 1–45. https://doi.org/10.1145/3007204

[20] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

[21] Chandramouli, R., & Butcher, Z. (2023). *A zero trust architecture model for access control in cloud-native applications in multi-cloud environments* (NIST Special Publication 800-207A). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207A

[22] Wan, A., Wallace, E., Shen, S., & Klein, D. (2023). Poisoning language models during instruction tuning. *Proceedings of the 40th International Conference on Machine Learning, 202*, 35413–35425. https://proceedings.mlr.press/v202/wan23b.html

[23] Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., & Zanella-Béguelin, S. (2023). Analyzing leakage of personally identifiable information in language models. *2023 IEEE Symposium on Security and Privacy (SP)*, 346–363. https://doi.org/10.1109/SP46215.2023.10179300

[24] Yang, Z., He, X., Li, Z., Backes, M., Humbert, M., Berrang, P., & Zhang, Y. (2023). Data poisoning attacks against multimodal encoders. *Proceedings of the 40th International Conference on Machine Learning, 202*, 39299–39313. https://proceedings.mlr.press/v202/yang23f.html

[25] Zhu, B., Cui, G., Chen, Y., Qin, Y., Yuan, L., Fu, C., Deng, Y., Liu, Z., Sun, M., & Gu, M. (2023). Removing backdoors in pre-trained models by regularized continual pre-training. *Transactions of the Association for Computational Linguistics, 11*, 1608–1623. https://doi.org/10.1162/tacl_a_00622

[26] Wang, W., & Feizi, S. (2023). Temporal robustness against data poisoning. *Advances in Neural Information Processing Systems, 36*. https://proceedings.neurips.cc/paper_files/paper/2023/hash/94bcb01789fccf15afe2764d8fe0f40e-Abstract-Conference.html

[27] Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P. S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, A., Biles, C., Brown, S., Kenton, Z., Hawkins, W., Stepleton, T., Birhane, A., Hendricks, L. A., Rimell, L., Isaac, W., & Gabriel, I. (2022). Taxonomy of risks posed by language models. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 214–229. https://doi.org/10.1145/3531146.3533088

[28] Kunaparaju, C. (2024). Adversarial AI in National Security: Understanding and Countering AI-Generated Cyber Threats. Letters in High Energy Physics.

[29] Elnikety, E., Mehta, A., Vahldiek-Oberwagner, A., Garg, D., & Druschel, P. (2016). Thoth: Comprehensive policy compliance in data retrieval systems. *25th USENIX Security Symposium (USENIX Security 16)*, 637–654. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/elnikety

[30] Steinhardt, J., Koh, P. W., & Liang, P. S. (2017). Certified defenses for data poisoning attacks. *Advances in Neural Information Processing Systems, 30*, 3517–3529. https://proceedings.neurips.cc/paper/2017/hash/9d7311ba459f9e45ed746755a32dcd11-Abstract.html

[31] Suciu, O., Marginean, R., Kaya, Y., Daumé III, H., & Dumitras, T. (2018). When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks. *27th USENIX Security Symposium (USENIX Security 18)*, 1299–1316. https://www.usenix.org/conference/usenixsecurity18/presentation/suciu

[32] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, 3–18. https://doi.org/10.1109/SP.2017.41