World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

LLM integration and the zero trust perimeter: A systematic review of trust enforcement failures and defensive architectures

Tendai Nemure *

Department of Graduate Computer Science and Engineering, Yeshiva University.

World Journal of Advanced Research and Reviews, 2026, 29(03), 1344-1353

Publication history: Received on 09 February 2026; revised on 16 March 2026; accepted on 18 March 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.29.3.0679

Abstract

Objective: This systematic review examines how large language model (LLM) integration into enterprise systems exposes implementation-level gaps in Zero Trust Architecture (ZTA), catalogs defensive architectures proposed to close these gaps, and presents a taxonomy mapping ZTA principles to LLM-induced failure modes.

Methods: Following PRISMA guidelines, 68 sources published between 2020 and 2026 were synthesized from six databases, standards bodies (NIST, OWASP, MITRE), and industry analyses, appraised using a three-tier quality framework distinguishing peer-reviewed, standards, and industry evidence.

Results: The review identifies trust enforcement failures across four dimensions: non-deterministic delegation collapsing identity verification; context and memory exploitation through prompt injection and memory poisoning; policy enforcement bypass via agentic automation; and microsegmentation weakening through RAG data aggregation. Defensive architectures range from production-ready guardrails to theoretical proposals for decentralized agent identity and semantic data flow controls. No existing standard comprehensively addresses the LLM–ZTA intersection.

Conclusion: ZTA principles remain sound but their implementation mechanisms require extension to account for probabilistic, context-dependent, and autonomously acting LLM components. Critical open problems include bounded trust models for agentic LLMs, verifiable AI identity, and standardized evaluation metrics for ZTA resilience under AI integration.

Keywords: Zero Trust Architecture; Large Language Models; Agentic AI; Prompt Injection; Microsegmentation; LLM Security

1. Introduction

Enterprise adoption of large language models (LLMs) has accelerated since 2023, with organizations deploying these systems for customer service automation, code generation, document analysis, and increasingly as autonomous agents executing multi-step workflows with real-world side effects [1,2]. In parallel, Zero Trust Architecture (ZTA) has become the prevailing enterprise security paradigm, formalized in NIST Special Publication 800-207 [3] and mandated for U.S. federal agencies under Executive Order 14028. The urgency of robust security architectures is underscored by the evolving threat landscape facing critical infrastructures, where cybersecurity challenges continue to outpace defensive capabilities [52]. ZTA rests on three axioms: never trust, always verify; assume breach; and enforce least privilege.

The convergence of these trends creates a fundamental tension. ZTA is technology-agnostic by design, but its practical implementation, the concrete mechanisms for identity verification, policy enforcement, and microsegmentation, has not

* Corresponding author: Tendai Nemure

kept pace with LLM-based systems. LLMs are probabilistic systems that process untrusted natural-language inputs, maintain context windows accumulating information across trust boundaries, and—when deployed as agents—execute actions with varying autonomy. An LLM agent inheriting user credentials collapses the identity verification chain not because ZTA fails to require verification, but because no deployed mechanism currently binds agent actions to user intent in real time. A RAG pipeline retrieving documents from multiple sensitivity tiers reveals that existing segmentation controls lack the semantic awareness to govern information flows within an LLM's processing context.

Existing literature addresses aspects of this intersection. The BSI/ANSSI joint release [4] proposes zero-trust design principles for LLM-based systems, the Cloud Security Alliance [5] offers guidance on securing enterprise data in LLM environments, and Red Hat [6] outlines zero-trust approaches for agentic AI. Comprehensive surveys catalog attacks against LLMs [7,8] and review ZTA deployment challenges [9]. However, three dimensions remain unaddressed: (a) systematic mapping of how each ZTA principle is affected by LLM integration at the implementation level; (b) comparative evaluation of defensive architectures against ZTA alignment criteria; and (c) a unified taxonomy connecting ZTA failure modes to LLM-specific attack vectors with measurable residual gaps.

This paper contributes: (1) a systematic analysis of LLM-induced ZTA implementation gaps across identity, context, policy enforcement, and data plane dimensions; (2) a structured review of defensive architectures evaluated against ZTA alignment; (3) a taxonomy of trust enforcement failures with measurable closure criteria, cross-principle analysis, and a worked case study; and (4) identification of open research challenges with practical intermediate steps.

2. Review Methodology

This systematic review follows PRISMA guidelines adapted for interdisciplinary cybersecurity research [10], synthesizing literature from January 2020 to March 2026 to encompass ZTA maturation following NIST SP 800-207 and the emergence of LLM-based enterprise systems.

2.1. Search strategy

Literature was identified through structured searches of IEEE Xplore, ACM Digital Library, Springer Link, ScienceDirect, Google Scholar, and arXiv. Two concept groups were combined: ("zero trust" OR "ZTA" OR "microsegmentation" OR "least privilege") AND ("large language model" OR "LLM" OR "prompt injection" OR "agentic AI" OR "RAG security" OR "guardrail"). Targeted searches for standards from NIST, OWASP, MITRE, and CSA supplemented database searches, along with snowball sampling from highly cited works.

2.2. Inclusion and exclusion criteria

Sources were included if they addressed: (a) interaction between LLMs and zero trust principles; (b) LLM security vulnerabilities relevant to trust-controlled environments; (c) defensive architectures for LLMs in zero-trust contexts; or (d) identity, access control, or segmentation for AI agents. Sources were excluded if they addressed general ML security without LLM specificity, discussed ZTA without AI integration, appeared in predatory venues, or presented opinion-based commentary without technical grounding.

2.3. Screening, quality appraisal, and limitations

Database searches returned 312 records; after removing 65 duplicates, 247 unique records remained. Title and abstract screening excluded 134 records. Full-text assessment of 113 sources excluded 45 (19 general ML without LLM specificity, 14 ZTA without AI integration, 7 insufficient depth, 5 predatory venues), yielding 68 retained sources. A random 20% sample of excluded records was re-evaluated one month later; no decisions were reversed.

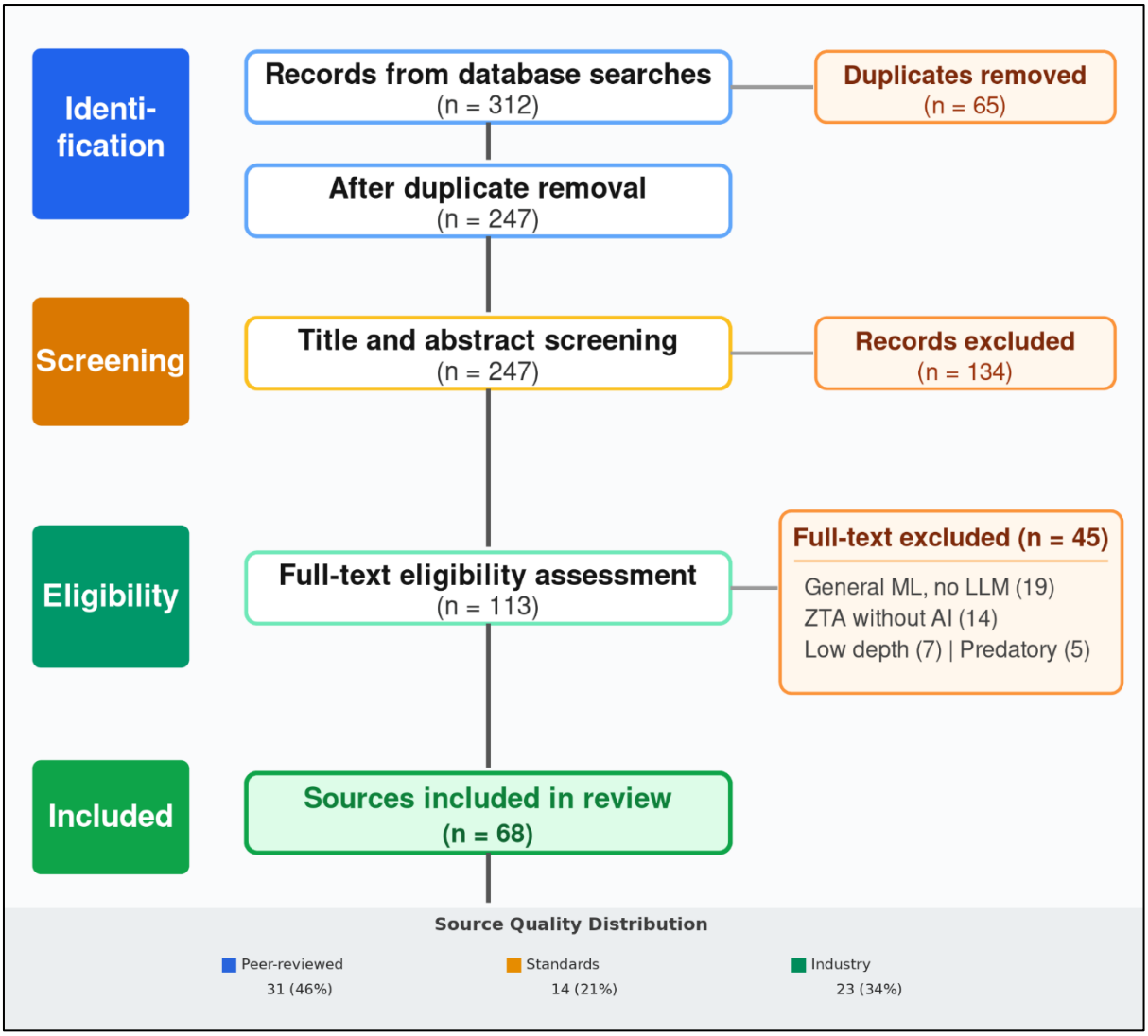

Figure 1 PRISMA flow diagram for systematic review screening process.

A three-tier quality appraisal framework was applied. Tier 1 (peer-reviewed: 31 sources, 46%) comprises journal articles and refereed proceedings anchoring core claims. Tier 2 (standards/technical reports: 14 sources, 21%) includes NIST, OWASP, MITRE, CSA, and BSI/ANSSI publications. Tier 3 (industry analysis: 23 sources, 34%) encompasses vendor white papers and security research blogs; claims are cross-referenced against higher-tier evidence, and commercial affiliations are noted. The corpus exhibits strong recency concentration: 72% of sources were published between 2024 and 2026, reflecting that the LLM–ZTA intersection has been actively researched only since late 2023.

3. Background

3.1. Zero Trust Architecture

Zero Trust has evolved through competing formulations: Kindervag's elimination of implicit trust [11], Google's BeyondCorp device-centric model [12], and the Forrester ZTX framework. NIST SP 800-207 [3] synthesized these into a vendor-neutral reference architecture adopted here for three reasons: it is the most comprehensive standards-based treatment, underpins U.S. federal mandates, and provides detailed architectural vocabulary (policy decision points, policy enforcement points, trust algorithms). NIST identifies six pillars: identity, device, network (microsegmentation), application/workload, data, and visibility/analytics [3,9].

3.2. LLM-based systems

LLMs are neural networks trained on extensive text corpora generating text through next-token prediction [1]. Enterprise deployments extend base LLMs through retrieval-augmented generation (RAG), which dynamically retrieves

documents from knowledge bases [13]; tool-use capabilities allowing API invocation and database queries; and agentic architectures granting multi-step planning, persistent memory, and multi-agent coordination [14]. Each extension introduces distinct security considerations: RAG exposes retrieval pipelines to data poisoning [15], tool-use bridges natural-language inputs to privileged operations, persistent memory enables memory poisoning attacks [16], and multi-agent systems allow lateral movement between agents [17].

3.3. Threat model and scope

The threat model encompasses four categories: external attackers targeting LLM-facing interfaces through prompt injection; malicious insiders leveraging legitimate access; compromised third-party agents carrying malicious instructions [16]; and accidental misuse by authorized users who inadvertently expose sensitive data through LLM interactions. Industry surveys report unintentional data leakage through LLM interfaces represents a larger proportion of incidents than deliberate attacks [18,19]. The scope includes conversational assistants with enterprise data access, RAG pipelines, agentic AI with tool-use, and LLM-powered SOC tools.

4. How LLM Integration Erodes ZTA Trust Assumptions

4.1. Identity and authentication collapse

ZTA demands every access request be tied to a verified identity with continuous re-authentication. LLM agents exhibit a failure mode extending beyond the classical confused deputy problem [20]. Unlike conventional middleware, an LLM agent's behavior is not a deterministic function of its inputs: the same prompt may produce different actions depending on context-window contents, model temperature, and stochastic sampling. This "non-deterministic delegation" means that even with properly scoped permissions, the mapping between user intent and agent actions is inherently uncertain. A database receiving a query from a conventional proxy can verify the request matches a defined API contract; a database receiving an LLM-generated query cannot verify it reflects the delegating user's intent.

The problem intensifies in multi-agent systems creating opaque delegation chains. The CSA's Agentic Trust Framework proposes continuous trust scoring [21]. Decentralized identifiers (DIDs) and verifiable credentials for AI agents have been proposed [22]. Microsoft's Entra Agent ID assigns discrete identities within Azure [23], and CyberArk's delegation framework addresses chain-of-custody in multi-agent systems [24]. However, these address the identity dimension without solving intent: they verify who the agent is, but not whether its actions represent the principal's intent.

4.2. Context window and memory as untrusted perimeter

The LLM context window, accumulating system prompts, user inputs, retrieved documents, and tool outputs, mixes instructions and data without structural separation. This pattern has a precedent: SQL injection exploits the same fundamental vulnerability [25]. However, prompt injection differs categorically. SQL injection is addressed through parameterized queries providing structural separation; no analogue exists for LLM contexts. Classical injection has binary success; prompt injection operates on a probabilistic spectrum, subtly biasing outputs. Indirect prompt injection—malicious instructions embedded in retrieved data, has no precise classical analogue [8,26]. OWASP ranks prompt injection as the number-one LLM risk [7].

Memory poisoning extends the threat temporally. Unit 42 demonstrated that indirect prompt injection can poison long-term memory, causing data exfiltration across subsequent sessions [16]. Plan injection research shows adversaries can bypass defenses by corrupting stored plans rather than immediate context, achieving three times higher success rates [27]. This represents a particularly insidious assume-breach violation: even after closing the initial vector, poisoned memory continues executing the attacker's intent.

4.3. Policy enforcement bypass via agentic automation

ZTA relies on policy decision points (PDPs) and policy enforcement points (PEPs) to evaluate access in real time. Agentic LLMs introduce policy bypass where actions are authorized because they originate from legitimate credentials, yet intent has been subverted. The Incalmo framework demonstrates LLM agents autonomously discovering vulnerabilities, executing exploits, and performing lateral movement [28]. CMU research confirms autonomous multistage attacks with hierarchical agent coordination [29]. Each individual action appears authorized, yet the sequence constitutes an unauthorized campaign. Flatt Security found attack success rates of approximately 73.5% against standard function-calling interfaces in controlled settings without guardrails [30]. LLMs also exhibit parameter hallucination during tool invocation, generating incorrect function arguments triggering unintended actions even without adversarial input [31].

4.4. Data plane aggregation and microsegmentation weakening

Well-architected RAG systems can enforce access controls at the retrieval layer [32]. The vulnerability arises when these controls are absent or misconfigured, common in rapidly deployed LLM integrations. PoisonedRAG achieves 90% success manipulating responses by injecting five malicious documents [15]. BadRAG shows poisoning 0.04% of a corpus achieves 98.2% attack success [33]. Adversarial embedding manipulation enables bypassing retrieval filters with approximately 80% success [34,35]. Beyond retrieval poisoning, LLM-powered analytics synthesizing data across organizational boundaries can surface correlations neither database's controls were designed to expose. Elisity [36] identifies microsegmentation as the critical missing layer for agent traffic governance.

5. Defensive Architectures for LLM-Integrated ZTA

5.1. Guardrails, filters, and orchestration

The most widely deployed defenses involve input/output filtering layers, guardrails, safety layers, or content moderation systems [37,38]. It is essential to evaluate these within their design scope: guardrails are application-layer filters, not enterprise policy engines. Criticizing them for lacking identity-awareness would be a category error. Within their scope, guardrails provide meaningful protection against known prompt injection patterns and content policy violations. However, commercial guardrail solutions show variance in effectiveness, with some failing to catch injection variants using encoding or multi-turn strategies [16]. The gap between guardrails and full ZTA alignment includes: identity-aware policy enforcement, cumulative risk assessment across request sequences, and integration with organizational security monitoring. Orchestration frameworks such as ToolSafe constrain agent tool usage through policy specifications [39,40], but require explicit policy authoring for every tool interaction.

5.2. Zero trust identity frameworks for agents

Several proposals aim to establish zero-trust identity primitives for AI agents at varying maturity levels. At the theoretical end, a DID/VC framework proposes cryptographically bound agent identities [22]. At early-deployment stage, Microsoft's Entra Agent ID [23] and CyberArk's delegation framework [24] are vendor-specific and lack cross-platform interoperability. The CSA's Agentic Trust Framework proposes continuous trust scoring [21], and BSI/ANSSI recommends treating LLM components as untrusted by default [4]. In the taxonomy (Section 7), defenses are labeled by maturity: theoretical (T), early-deployment (ED), or production-ready (PR).

5.3. Microsegmentation and network-level controls

AI-aware microsegmentation extends conventional segmentation for LLM traffic patterns. ColorTokens [41] demonstrates LLM-driven microsegmentation reducing cycle times from days to minutes. Elisity [36] proposes microsegmentation as essential for AI agent containment. Practical controls include API gateway policies, vector store isolation, separate segments for training, inference, and agent execution, and anomalous data flow monitoring.

5.4. Memory and context security

Defenses against memory poisoning include temporal scoping policies, integrity verification for memory stores, and cross-context isolation [42]. Memory hygiene protocols combining periodic audits with provenance tracking have been recommended [43]. Context security controls include input sanitization, content partitioning with differentiated trust levels, and output validators. Hardware-backed attestation for isolating agent execution environments has been proposed [44]. The fundamental tension between the LLM's need to process diverse inputs holistically and ZTA's strict boundary enforcement remains unresolved.

6. Standards, Frameworks, and Compliance

Each framework is evaluated against four criteria a comprehensive treatment would require: (C1) explicit mapping of LLM threats to ZTA components; (C2) agent identity/access control guidance; (C3) data flow controls for RAG/LLM contexts; (C4) monitoring requirements for LLM-initiated actions. No existing framework satisfies all four.

The NIST AI Risk Management Framework [45] and its Generative AI Profile [46] provide partial C1 and C4 coverage but do not address C2 or C3 in ZTA-specific terms. OWASP Top 10 for LLM Applications [7] satisfies C1 comprehensively from the application layer; the OWASP GenAI agentic security release partially addresses C2 [47]. MITRE ATLAS [48] provides excellent C1 coverage enabling SOC/SIEM integration. The CSA has published guidance relevant to C2 and C3

partially [5,49,50]. The BSI/ANSSI joint release [4] is the only document explicitly bridging ZTA and LLM security across all four criteria at principles level, though without detailed implementation guidance.

The EU AI Act intersects with ZTA at specific points: Article 15 requires robustness against adversarial manipulation (overlapping with assume-breach), Article 14 mandates human oversight (aligning with continuous monitoring), and Article 13 transparency requirements support visibility. However, the Act does not prescribe network architecture or segmentation, and organizations subject to both the AI Act and ZTA mandates must independently bridge these frameworks.

7. Synthesis: Taxonomy of Trust Enforcement Failures

Table 1 synthesizes the analysis from Sections 4–6, mapping ZTA principles to LLM-induced failure modes, representative attacks, existing defenses labeled by maturity (T = theoretical, ED = early deployment, PR = production-ready), and residual gaps expressed as measurable closure criteria.

Table 1 Taxonomy of LLM-Induced ZTA Trust Enforcement Failures

ZTA Principle	LLM-Induced Failure Mode	Representative Attack	Existing Defense (Maturity)	Residual Gap (Closure Criterion)
Identity: Continuous verification	Non-deterministic delegation	Agent impersonation; delegation chain exploitation	DIDs/VCs (T); Entra Agent ID (ED); CyberArk (ED)	Closed when: ≥2 cloud providers adopt interoperable agent identity
Least Privilege: Minimal access	Over-privileged tool access	Function-calling exploitation; parameter hallucination [30]	ToolSafe (T); RBAC for tools (ED)	Closed when: dynamic permission scoping deployed in production
Assume Breach: Limit blast radius	Memory poisoning persists across sessions	Persistent prompt injection [16]; plan injection [27]	Memory lifetime policies (T); provenance tracking (ED); audits (PR)	Closed when: memory integrity verification <5% false negative
Microsegmentation: Restrict lateral movement	RAG pipelines cross segmentation boundaries	PoisonedRAG [15]; BadRAG [33]; embedding poisoning	Access-controlled RAG (PR); vector isolation (ED); API gateways (PR)	Closed when: semantic-aware segmentation architecture standardized
Policy Enforcement: Real-time PDP/PEP	Authorized credentials, subverted intent	Autonomous multistage attacks [28]; agentic lateral movement	Guardrails (PR); orchestration policy (ED); anomaly detection (ED)	Closed when: intent-aware PEP >90% detection of multi-step bypass
Data Protection: Classify and protect	Context window aggregates multi-tier data	Indirect injection for exfiltration; cross-context leakage	Content partitioning (T); sanitization (PR); output validation (PR)	Closed when: structural instruction/data separation validated
Visibility: Continuous monitoring	Opaque LLM reasoning resists audit	Stealth exfiltration; reasoning manipulation	Prompt/response logging (PR); SIEM integration (ED); explainability (T)	Closed when: causal chain audit framework standardized

Cross-principle attack dynamics. The table simplifies multi-dimensional attack dynamics. In practice, attacks span multiple principles: a prompt injection causing an agent to invoke a privileged tool while evading detection implicates Data Protection, Policy Enforcement, Identity, and Visibility in a single chain. The Incalmo framework [28] demonstrates cascading across identity, segmentation, policy, and monitoring planes. Memory poisoning can persist (Assume Breach) and later trigger data aggregation (Microsegmentation) or tool misuse (Policy Enforcement) invisibly (Visibility). Three

cross-cutting patterns emerge: (1) granularity mismatch—ZTA enforces per-request policies while LLMs process blurred request boundaries; (2) temporal persistence—memory poisoning survives verification cycles; (3) the intent gap—ZTA verifies identity but cannot verify that actions match the principal's intent.

7.1. Worked example: ChatGPT memory poisoning

In September 2024, researcher Johann Rehberger disclosed that ChatGPT's persistent memory could be exploited through indirect prompt injection [51,16]. Applying the taxonomy: Identity—ChatGPT acted under user credentials for both memory writes and exfiltration; Assume Breach, injected memories persisted indefinitely; Data Protection, poisoned memory caused sensitive data aggregation with exfiltration instructions; Visibility, memory modifications were not surfaced in audit logs; Policy Enforcement, each action was individually authorized, with the violation occurring only in adversarial orchestration. This single incident maps to five taxonomy rows, demonstrating the cross-principle cascading pattern.

8. Open Research Challenges and Future Directions

8.1. Bounded trust models for agentic LLMs

Current ZTA frameworks compute trust from identity, context, and device posture. For LLM agents, trust must additionally account for probabilistic decision-making, context integrity, and intent fidelity. A formal trust calculus remains uncertain for non-deterministic systems. As an intermediate step, bounded verification frameworks for constrained agent behaviors, such as formally verifying tool-calling policy restrictions, would provide practical security benefits while broader challenges are pursued.

8.2. Verifiable AI identity and intent attestation

DID/VC proposals address identity but leave intent unresolved. Intent attestation has deep philosophical roots: even between humans, verifying that an intermediary faithfully represents the principal's intent is contested in agency law and organizational theory. Practical intermediate mechanisms include: constrained action spaces restricting outputs to pre-approved sets; human-in-the-loop confirmation for high-risk actions; and behavioral baselines flagging statistical deviations from expected action distributions.

8.3. AI-native microsegmentation

Segmentation must extend to the semantic layer, controlling information flows into and out of LLM contexts based on data sensitivity, query purpose, and principal authorization. An intermediate step is retrieval-time access control for RAG systems enforcing document-level permissions, combined with output classification blocking responses above the user's authorization tier. Such systems exist in embryonic form but lack standardization.

8.4. Evaluation metrics

No standardized method exists to assess whether an organization's zero-trust posture is degraded by LLM integration. Candidate metrics include: percentage of LLM-initiated actions subject to independent verification; data aggregation degree across trust boundaries; guardrail effectiveness ratios; and mean time to detect LLM-mediated policy violations.

8.5. Standardization

The defensive landscape is fragmented across vendor-specific implementations. An integrated reference architecture bridging NIST ZTA, NIST AI RMF, OWASP, and MITRE ATLAS would advance deployment. International consensus with cross-vendor interoperability is needed, particularly for agent identity.

9. Conclusion

This systematic review has examined the intersection of LLM security and Zero Trust Architecture, identifying trust enforcement failures that emerge when LLM-based systems are integrated into ZTA-governed environments. These failures arise not because ZTA principles are inadequate, but because implementation mechanisms have not been extended for probabilistic, context-dependent, and autonomously acting LLM agents. Failures manifest across four dimensions: non-deterministic delegation collapsing identity verification; context and memory exploitation; policy enforcement bypass via agentic automation; and microsegmentation weakening through data aggregation.

The taxonomy maps seven ZTA principles to corresponding failure modes with measurable closure criteria. The worked case study demonstrates that real-world incidents cascade across multiple principles, reinforcing the need for cross-plane correlation. The cross-cutting patterns, granularity mismatch, temporal persistence, and the intent gap, point toward AI-native trust primitives, including bounded agent behavior verification, retrieval-time access control standardization, human-in-the-loop confirmation, behavioral monitoring, and formal trust models as longer-term objectives.

Compliance with ethical standards

Disclosure of conflict of interest

The author declares no conflict of interest.

Statement of ethical approval

This study is a systematic literature review and did not involve human subjects, animal experimentation, or the collection of primary data. Ethical approval was not required.

References

[1] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems. 2020;33:1877–1901. Available from: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[2] OpenAI. GPT-4 technical report. arXiv preprint arXiv:2303.08774. 2023. Available from: https://doi.org/10.48550/arXiv.2303.08774

[3] Rose S, Borchert O, Mitchell S, Connelly S. Zero Trust Architecture. NIST Special Publication 800-207. 2020. Available from: https://doi.org/10.6028/NIST.SP.800-207[papers.nips]

[4] BSI/ANSSI. Design principles for LLM-based systems with zero trust. Federal Office for Information Security / Agence nationale de la sécurité des systèmes d'information. 2024. Available from: https://www.aigl.blog/design-principles-for-llm-based-systems-with-zero-trust/

[5] Cloud Security Alliance. Using zero trust to secure enterprise information in LLM environments. CSA Artifacts. 2025. Available from: https://cloudsecurityalliance.org/artifacts/using-zero-trust-to-secure-enterprise-information-in-llm-environments

[6] Red Hat. Zero trust for autonomous agentic AI systems: Building more secure foundations. Red Hat Emerging Technologies. 2026. Available from: https://next.redhat.com/2026/02/26/zero-trust-for-autonomous-agentic-ai-systems-building-more-secure-foundations/

[7] OWASP. Top 10 for LLM Applications 2025. OWASP Foundation. 2025. Available from: https://owasp.org/www-project-top-10-for-large-language-model-applications/[aigl]

[8] Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injections. arXiv preprint arXiv:2302.12173. 2023. Available from: https://arxiv.org/abs/2302.12173

[9] Alshuwaikh M, et al. A systematic literature review on the implementation and challenges of Zero Trust Architecture across domains. Sensors. 2025;25(12). Available from: https://doi.org/10.3390/s25123714[next.redhat]

[10] Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ. 2021;372:n71. Available from: https://doi.org/10.1136/bmj.n71

[11] Kindervag J. No more chewy centers: Introducing the Zero Trust model of information security. Forrester Research. 2010. Overview at: https://www.linkedin.com/pulse/birth-zero-trust-2010-forrester-report-nikhil-rana-qivsf)[13]

[12] Ward R, Beyer B. BeyondCorp: A new approach to enterprise security. ;login:. 2014;39(6):6–11. Available from: https://www.usenix.org/publications/login/dec14/ward

[13] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems. 2020;33:9459–9474. Available from: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[14] Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, et al. A survey on large language model based autonomous agents. Frontiers of Computer Science. 2024;18(6):186345. Available from: https://doi.org/10.1007/s11704-023-3214-3

[15] Zou W, Geng R, Wang B, Jia J. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation systems. In: Proceedings of the 34th USENIX Security Symposium. 2025. (Forthcoming). Available from: https://www.usenix.org/conference/usenixsecurity25/cycle1-accepted-papers

[16] Palo Alto Networks Unit 42. When AI remembers too much: Persistent behaviors in agents' memory. 2025. (Web article)[arxiv]

[17] PipeLab. Lateral movement in multi-agent LLM systems. PipeLab Research Blog. 2025. [i.blackhat]

[18] Google Threat Intelligence Group. Advances in threat actor usage of AI tools. Google Cloud Blog. 2025.[semanticscholar]

[19] Microsoft. Manipulating AI memory for profit: The rise of AI recommendation poisoning. Microsoft Security Blog. 2026. [ar5iv.labs.arxiv]

[20] Hardy N. The confused deputy: (Or why capabilities might have been invented). ACM SIGOPS Operating Systems Review. 1988;22(4):36–38. Available from: https://doi.org/10.1145/63039.63042

[21] Cloud Security Alliance. The agentic trust framework: Zero trust governance for AI agents. CSA Blog. 2026. Available from: https://cloudsecurityalliance.org/blog/2026/02/02/the-agentic-trust-framework[dl.acm]

[22] ArXiv 2505.19301. A novel zero-trust identity framework for agentic AI: Decentralized authentication and fine-grained access control. 2025.[academia]

[23] Microsoft. Zero-trust agents: Adding identity and access to multi-agent workflows. Microsoft Tech Community. 2025.[usenix]

[24] CyberArk. Zero trust for AI agents: Delegation, identity and access control. CyberArk Developer Blog. 2026. [arxiv]

[25] OWASP. SQL injection. OWASP Foundation. 2017. Available from: https://owasp.org/www-community/attacks/SQL_Injection[arxiv]

[26] CETaS. Indirect prompt injection: Generative AI's greatest security flaw. Centre for Emerging Technology and Security, The Alan Turing Institute. 2024. [arxiv]

[27] ArXiv 2506.17318. Context manipulation attacks: Web agents are susceptible to corrupted memory (plan injection). 2025. [arxiv]

[28] Happe A, Cito J. On the feasibility of using LLMs to execute multistage network attacks (Incalmo). arXiv preprint arXiv:2501.16466. 2025. Available from: https://arxiv.org/abs/2501.16466

[29] Carnegie Mellon University. When LLMs autonomously attack. College of Engineering News. 2025. (News article; no DOI.) Available from (example coverage): https://finance.yahoo.com/news/carnegie-mellon-researchers-demonstrate-llms-130500524.html

[30] Flatt Security. Securing LLM function-calling: Risks and mitigations for AI agents. GMO Flatt Security Research. 2025. [usenix]

[31] Giskard. Function calling in LLMs: Testing agent tool usage for AI security. Giskard Research. 2025. [usenix]

[32] Microsoft. Security planning for LLM-based applications. Microsoft Learn. 2025. [semanticscholar]

[33] ArXiv 2509.20324. RAG security and privacy: Formalizing the threat model and attack surface. 2025. [research.google]

[34] Prompt Security. The embedded threat in your LLM: Poisoning RAG pipelines via vector embeddings. 2025. [hcisj]

[35] The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). In: Findings of ACL 2024. 2024;pp. 4505–4523.

[36] Elisity. AI agent network security: Why microsegmentation is the missing layer. Elisity Blog. 2026. Available from: https://www.elisity.com/blog/ai-agent-network-security-microsegmentation-2026

[37] Datadog. LLM guardrails: Best practices for deploying LLM apps securely. Datadog Blog. 2025. Available from: https://www.datadoghq.com/blog/llm-guardrails-best-practices/

[38] Safeguarding large language models: A survey. PMC/MDPI. 2025. Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC12532640/

[39] ArXiv 2601.10156. ToolSafe: Enhancing tool invocation safety of LLM-based agents. 2025. Available from: https://arxiv.org/abs/2601.10156

[40] ArXiv 2503.18666. Customizable runtime enforcement for safe and reliable LLM agents. 2025. [cs.reviewer]

[41] ColorTokens. AI-assisted microsegmentation against AI hackers. ColorTokens Blog. 2025. [ieee-jas]

[42] ArXiv 2601.05504. Memory poisoning attack and defense on memory-based LLM-agents. 2025. Available from: https://arxiv.org/abs/2601.05504

[43] Lakera AI. Agentic AI threats: Memory poisoning and long-horizon goal hijacks. Lakera Blog. 2025. [ieeexplore.ieee]

[44] IACR. Systems security foundations for agentic computing. IACR ePrint 2025/2173. 2025. Available from: https://eprint.iacr.org/2025/2173

[45] NIST. Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1. 2023. Available from: https://doi.org/10.6028/NIST.AI.100-1

[46] NIST. Artificial intelligence risk management framework: Generative AI profile. NIST AI 600-1. 2024. Available from: https://doi.org/10.6028/NIST.AI.600-1

[47] OWASP GenAI Security Project. Top 10 risks and mitigations for agentic AI security. 2025. Available from: https://genai.owasp.org/

[48] MITRE. ATLAS: Adversarial Threat Landscape for AI Systems. 2025. Available from: https://atlas.mitre.org/

[49] Cloud Security Alliance. Securing LLM-backed systems: Essential authorization practices. CSA Artifacts. 2024.

[50] Cloud Security Alliance. AI Controls Matrix. CSA Artifacts. 2025.

[51] Lakera AI. ChatGPT's memory exploit: Persistent prompt injection attack. Lakera Blog. 2024.

[52] Nemure T, Mupa MN, Agyei KG, Chisora HH, Mudzingwa K, Chiwanga R. Cybersecurity of critical infrastructures: Challenges and future perspectives. World Journal of Advanced Research and Reviews. 2026;29(03):870–883. Available from: https://doi.org/10.30574/wjarr.2026.29.3.0621