ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. No. 4 (32), Vol. 2. pp. 185–193
DOI: https://doi.org/10.5281/zenodo.24115600
Journal website: https://cajournal.kz

UDC 004.056:004.8

AI ATTACK MONITOR: A STANDARDS-ALIGNED ARCHITECTURE FOR MONITORING AND RISK SCORING OF ATTACKS ON LARGE LANGUAGE MODEL APPLICATIONS

Abdrasil Alisher, Altaev Amir, Sisenov Rakhat
Satbayev University, Almaty, Kazakhstan

Abdrasil Alisher — 4th-year student, Department of Cybersecurity, Information Processing and Storage, Satbayev University, Almaty, Kazakhstan.

Altaev Amir — 4th-year student, Department of Cybersecurity, Information Processing and Storage, Satbayev University, Almaty, Kazakhstan.

Sisenov Rakhat — 4th-year student, Department of Cybersecurity, Information Processing and Storage, Satbayev University, Almaty, Kazakhstan.

© Abdrasil Alisher, Altaev Amir, Sisenov Rakhat, 2026

Abstract: Applications built on large language models (LLMs) are exposed to attacks that bypass perimeter defenses, including direct and indirect prompt injection, jailbreaking, data and model poisoning, sensitive information disclosure and excessive agency of AI agents. Commercial AI security products have multiplied, but the market has rapidly consolidated, and organizations lack a transparent way to compare products and to build monitoring that fits their existing security operations. This paper analyzes eight AI security solutions against eight functions derived from the OWASP Top 10 for LLM Applications 2026 and MITRE ATLAS, using an ordinal 0–2 scale and a weighted scoring model. Prisma AIRS obtained the highest weighted score (88.75 of 100), followed by Cisco AI Defense and Check Point AI Security (76.25 each), while protection against data poisoning (50.0% mean coverage) and data leakage prevention (56.25%) were the weakest functions across the market. Based on these gaps, we propose AI Attack Monitor, a pipeline architecture consisting of a security gateway, detection engine, risk analyzer, alerting and dashboard, with a transparent risk-scoring model and a unified event schema mapped to OWASP and ATLAS identifiers. An evaluation plan on public benchmarks is outlined.

Keywords: AI security; large language models; prompt injection; attack monitoring; OWASP Top 10 for LLM; MITRE ATLAS; risk scoring; AI agents

1. Introduction

Large language models (LLMs) are now embedded in chatbots, coding assistants, retrieval-augmented generation (RAG) systems and autonomous agents that call external tools. In such applications natural language serves simultaneously as data and as instructions, which creates an attack surface that has no direct analogue in classical software [1,2]. The OWASP Top 10 for LLM Applications 2026 again ranks prompt injection as the most critical risk [3], MITRE ATLAS catalogs adversarial tactics and techniques against AI systems [4], and the National Institute of Standards and Technology (NIST) maintains a taxonomy of adversarial machine learning attacks and mitigations [5].

Conventional controls such as web application firewalls, intrusion detection systems and signature-based data loss prevention (DLP) inspect syntax rather than intent. They cannot recognize

an instruction hidden inside a retrieved web page or document [2,6], nor an unsafe tool call issued by an agent after reading attacker-controlled content [7]. Effective protection therefore requires monitoring at the level of prompts, model responses, retrieved context and agent actions, together with a way to prioritize events for a security operations center (SOC).

The industry has responded with a new class of AI security platforms. Between 2024 and 2025 the market consolidated quickly: Robust Intelligence became the basis of Cisco AI Defense, Protect AI was integrated into Palo Alto Networks Prisma AIRS, Prompt Security joined SentinelOne, and Lakera became part of Check Point [8–11]. As a result, many runtime protection capabilities are now bundled into large proprietary platforms whose detection logic and risk scoring are not transparent. This is a practical barrier for small and medium-sized organizations and for university SOCs, including those in Kazakhstan, that need to understand what is being detected and why.

This paper addresses that problem. Its contributions are as follows: (1) a structured comparison of eight AI security solutions against eight functions derived from OWASP and MITRE ATLAS, combined with a weighted scoring model; (2) the identification of functions that are weakly covered across the market; (3) the AI Attack Monitor architecture, which includes a unified event schema mapped to OWASP and ATLAS identifiers and an explicit risk-scoring model; and (4) a plan for evaluating the architecture on public benchmarks. Section 2 reviews related work and states the problem, Section 3 describes the methodology, Section 4 presents the results, Section 5 discusses them, and Section 6 concludes the paper.

2. Related Work and Problem Statement

2.1. Threat Taxonomies and Governance Frameworks

Several frameworks define what must be protected in AI systems. The OWASP Top 10 for LLM Applications lists ten application-level risks; in the 2026 edition, excessive agency rose from sixth to third place and system prompt leakage was broadened into hidden context exposure, reflecting the spread of AI agents [3]. MITRE ATLAS organizes adversary behavior into tactics and techniques in the style of MITRE ATT&CK [4]. The NIST AI Risk Management Framework introduces the Govern, Map, Measure and Manage functions [12] and is complemented by a profile for generative AI [13]. Google's Secure AI Framework (SAIF) extends existing security controls to AI and calls for detection and response that covers AI-specific threats [14]. ISO/IEC 42001 specifies requirements for an AI management system [15]. Microsoft publishes a catalog of AI attack techniques and red-teaming guidance [16,17], and a comprehensive survey of LLM security and privacy is given in [18]. These documents define *what* to protect and *why*, but they deliberately leave the operational design of monitoring to implementers.

2.2. Attacks on LLM Applications

Direct prompt injection was demonstrated early through goal hijacking and prompt leaking [1] and was later systematized for real LLM-integrated applications [19]. Indirect prompt injection places malicious instructions in content that the application retrieves on its own, such as e-mails, web pages or documents [2,20]. Liu et al. formalized prompt injection and showed that combined attacks remain effective against many defenses [21]. Jailbreaking exploits competing objectives and mismatched generalization in safety training [22]; it can be automated through gradient-based adversarial suffixes [23] or through iterative black-box refinement that needs only a few queries [24]. Models may also disclose memorized training data [25]. Poisoning web-scale training datasets has been shown to be practical [26], and a few injected texts can corrupt the knowledge base of a RAG system [6]. For tool-using agents, AgentDojo shows that prompt injection can trigger harmful actions [7].

2.3. Detection and Defense Techniques

Defenses fall into several groups. Perplexity-based filtering detects machine-optimized adversarial suffixes [27,28]. LLM-based safety classifiers such as Llama Guard label both inputs and outputs according to a hazard taxonomy [29]. Programmable guardrails such as NeMo Guardrails enforce dialog and topical policies [30]. Prompt-level defenses such as spotlighting mark untrusted input so that the model can distinguish data from instructions [31]. Standardized red-

teaming frameworks such as HarmBench support systematic evaluation [32]. Each technique, however, covers only a subset of threats, and adaptive attackers can evade individual detectors [21,28]. Moreover, these defenses are usually evaluated in isolation rather than as parts of a monitoring pipeline with event correlation, risk prioritization and SIEM integration.

2.4. Problem Statement

We found no open reference architecture that (a) collects prompts, responses, retrieved context and tool calls as normalized security events, (b) labels them with a common taxonomy, (c) prioritizes them with a transparent risk score, and (d) exports them to existing SIEM/SOC tools. Commercial platforms partially address these needs, but their coverage is uneven and their logic is proprietary. This study therefore addresses three research questions: **RQ1**—to what extent do current AI security solutions cover the core monitoring functions; **RQ2**—which functions are least covered; and **RQ3**—what architecture can close these gaps in a transparent, standards-aligned way?

3. Materials and Methods

3.1. Research Design

The study consisted of three stages: (i) selection of AI security solutions; (ii) functional assessment and weighted scoring of the selected solutions; and (iii) design of the proposed architecture based on the gaps identified in stage (ii) and on the principles of SAIF [14] and the NIST AI RMF [12].

3.2. Selection of Solutions

A solution was included if (1) it was commercially available, or historically significant for the category, as of September 2026, (2) its vendor publicly documented runtime protection for LLM applications, and (3) the resulting sample covered both platforms formed through acquisitions and independent vendors. Eight solutions met these criteria (Table 1). Robust Intelligence AI Firewall is no longer sold separately but was retained as a reference point because it pioneered the AI firewall concept and became the basis of Cisco AI Defense [8,33].

Table 1 – AI security solutions selected for the comparative analysis (status as of September 2026).

Solution	Vendor	Deployment	Market status
Prisma AIRS	Palo Alto Networks	Cloud/API	Active; integrates Protect AI (acquired 2025)
Cisco AI Defense	Cisco	Cloud + Talos telemetry	Active; built on Robust Intelligence (acquired 2024)
Check Point AI Security	Check Point (formerly Lakera)	Cloud/API	Lakera acquired in 2025
SentinelOne Singularity AI Security	SentinelOne (formerly Prompt Security)	Runtime, MCP gateway	Prompt Security acquired in 2025
HiddenLayer	HiddenLayer	Cloud/API	Independent
Azure AI security tools	Microsoft	Cloud (Azure)	Part of the Microsoft ecosystem
Noma Security	Noma Security	Cloud/SaaS	Independent (founded 2023)
Robust Intelligence AI Firewall	Robust Intelligence (now Cisco)	Runtime	Legacy; part of Cisco AI Defense since 2024

3.3. Functional Assessment

Eight functions were derived from the OWASP Top 10 for LLM Applications [3] and MITRE ATLAS [4]: F1, real-time monitoring; F2, prompt injection detection; F3, data leakage prevention (DLP); F4, data and model poisoning protection; F5, AI agent security; F6, red teaming; F7, SIEM integration; and F8, automatic blocking. Each solution j received an ordinal score $x_{jk} \in \{0, 1, 2\}$ for each function k (0, absent; 1, partially implemented; 2, fully implemented). The scores were assigned jointly by the authors based on public product documentation, datasheets and technical blogs available in September 2026 [8–11,33–36].

© Central Asian Scientific Journal, 2026

3.4. Weighted Scoring Model

Because the functions are not equally important for operational protection, a weighted score was computed for each solution as follows:

$$S_j = 100 \cdot \sum_k w_k \cdot \frac{\tilde{x}_{jk}}{2} \quad (1)$$

where $\tilde{x}_{jk}$ is the score aggregated into seven criteria and w_k are the criterion weights, which sum to one. The weights were set as follows: attack detection, 0.25 (mean of F2 and F4); real-time monitoring, 0.20 (F1); data protection, 0.15 (F3); AI agent security, 0.15 (F5); model assurance through red teaming, 0.10 (F6); integration, 0.10 (F7); and automated response, 0.05 (F8). Detection and monitoring received the largest weights because the speed of identifying an attack largely determines the practical value of a product. The market-wide coverage of each function was computed as follows:

$$C_k = 100 \cdot \frac{\sum_j x_{jk}}{2N} \quad (2)$$

where $N = 8$ is the number of solutions. To check the robustness of the ranking, it was recomputed with equal weights, which is equivalent to normalizing the raw sum of scores.

3.5. Architecture Design

The architecture was designed against five requirements derived from the identified gaps and from SAIF [14]: R1, visibility of all LLM interactions, including tool calls; R2, a common taxonomy for events (OWASP and ATLAS); R3, a transparent and configurable risk score; R4, graded responses (log, alert or block); and R5, integration with SIEM through standard formats. The NIST AI RMF functions [12] were used to place each component in the risk management cycle.

4. Results

4.1. Functional Comparison

Table 2 summarizes the functional scores. Prisma AIRS achieved the highest total (14 of 16) and the highest weighted score (88.75), followed by Cisco AI Defense and Check Point AI Security (76.25 each). HiddenLayer (73.75), SentinelOne Singularity AI Security and Noma Security (72.50 each) formed the middle group. Microsoft Azure AI security tools scored 55.00, and the legacy Robust Intelligence AI Firewall scored 41.25. The weighted ranking is shown in Figure 1.

Table 2 – Functional scores of the solutions (0, absent; 1, partial; 2, full), total score and weighted score S_j (Equation (1)). F1, monitoring; F2, prompt injection; F3, DLP; F4, data/model poisoning; F5, AI agent security; F6, red teaming; F7, SIEM integration; F8, automatic blocking.

Solution	F1	F2	F3	F4	F5	F6	F7	F8	Sum	S_j
Prisma AIRS	2	2	2	1	2	1	2	2	14	88.75
Cisco AI Defense	2	2	1	1	1	2	2	1	12	76.25
Check Point AI Security	2	2	2	1	1	1	1	2	12	76.25
SentinelOne Singularity	2	2	1	0	2	1	2	1	11	72.50
HiddenLayer	2	1	1	2	2	1	1	1	11	73.75
Azure AI security	1	1	1	1	1	1	2	1	9	55.00
Noma Security	2	1	1	1	2	2	1	1	11	72.50
Robust Intelligence (legacy)	1	2	0	1	0	2	0	1	7	41.25
Coverage C_k, %	87.5	81.2	56.2	50	68.8	68.8	68.8	62.5		

© Central Asian Scientific Journal, 2026

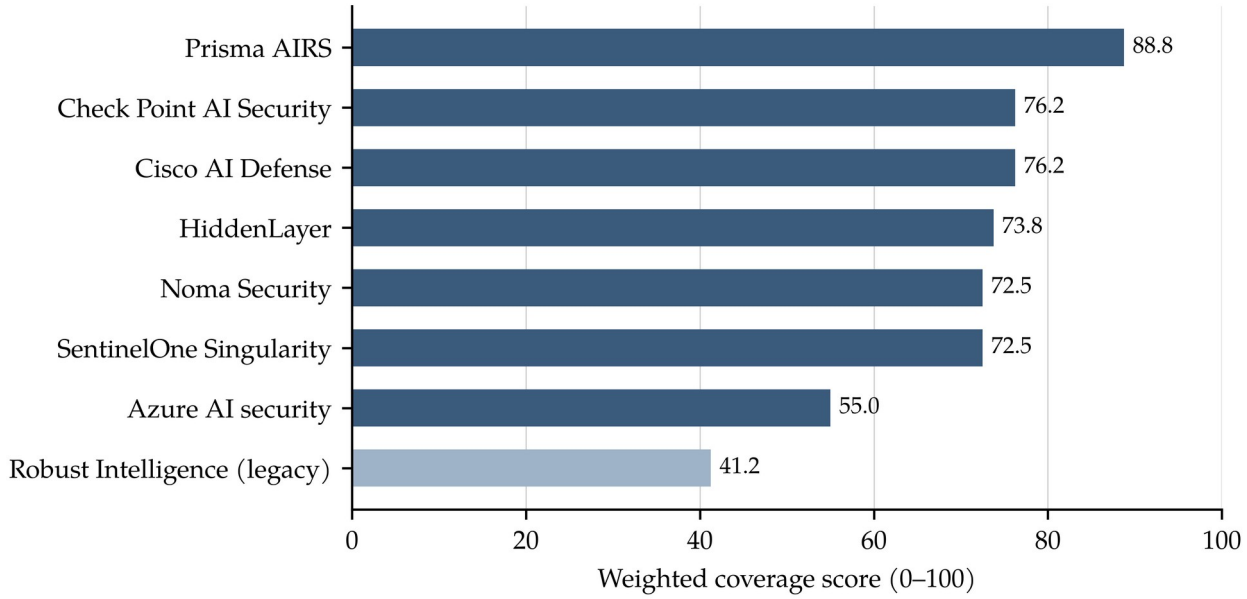


Figure 1 – Weighted coverage scores of the eight AI security solutions (Equation (1)). The lighter bar denotes a legacy product that is no longer sold separately.

Weighting changed the order among solutions with equal raw totals: HiddenLayer, SentinelOne and Noma Security each scored 11 points, but HiddenLayer ranked higher because of its full coverage of data poisoning, which contributes to the attack detection criterion. With equal weights, Prisma AIRS (87.50) remained first, Cisco AI Defense and Check Point AI Security (75.00) remained second, and the three independent or recently acquired solutions were tied at 68.75. The leading positions were therefore stable with respect to the choice of weights.

4.2. Coverage Gaps

Figure 2 shows the market-wide coverage of each function (Equation (2)). Real-time monitoring (87.5%; fully implemented in six of eight solutions) and prompt injection detection (81.25%; five of eight) are mature. In contrast, data and model poisoning protection (50.0%) was fully implemented only by HiddenLayer, and DLP (56.25%) and automatic blocking (62.5%) were each fully implemented by only two solutions. No solution reached the maximum score for all eight functions. Consolidated platforms scored higher on SIEM integration and blocking, whereas independent vendors scored higher on agent security and red teaming.

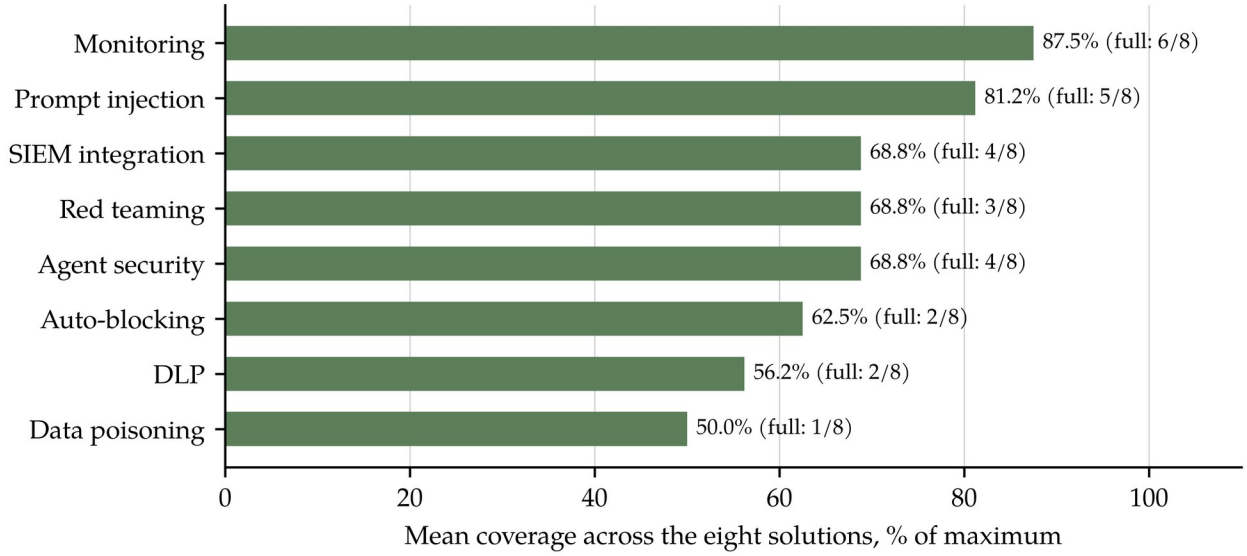


Figure 2 – Market-wide coverage of the eight functions (Equation (2)); the numbers in parentheses show how many solutions fully implement each function.

4.3. Proposed Architecture: AI Attack Monitor

To address RQ3, AI Attack Monitor is designed as a pipeline of specialized modules rather than as a monolithic platform (Figure 3). Every module writes normalized events to a shared event store, so the full chain from interception to decision remains auditable.

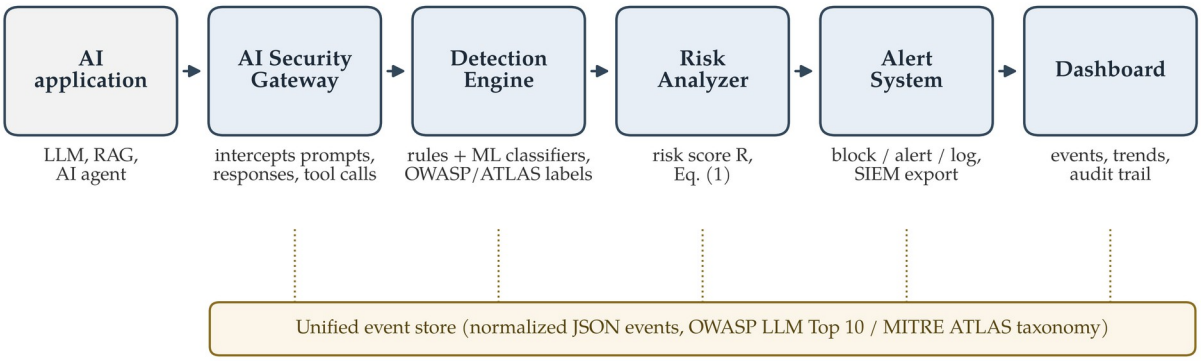


Figure 3 – Architecture of AI Attack Monitor. Solid arrows show the processing pipeline; dotted lines show how events are written to the unified event store.

The **AI Security Gateway** is deployed as a reverse proxy or as a software development kit (SDK) between the application and the model provider. It intercepts user prompts, retrieved RAG context, model responses and agent tool calls and converts them into JSON events containing a session identifier, source, content hash, trust level and timestamp (R1). The **Detection Engine** applies layered detectors: deterministic rules and allow-lists; perplexity and heuristic filters for adversarial suffixes [27]; an LLM-based input and output classifier [29]; canary tokens for hidden context exposure; and provenance checks for retrieved documents. Each finding is labeled with an OWASP category and an ATLAS technique (R2), as listed in Table 3.

Table 3 – Mapping of OWASP Top 10 for LLM Applications 2026 categories to MITRE ATLAS techniques, detection modules of AI Attack Monitor and default severity *s*.

OWASP 2026 category	MITRE ATLAS technique	Detection module	*s*
LLM01 Prompt Injection (direct)	AML.T0051.000	Rules, perplexity filter, input classifier	0.8
LLM01 Prompt Injection (indirect)	AML.T0051.001	Untrusted-content marking, classifier on retrieved context	1.0
LLM01 Jailbreak	AML.T0054	Input classifier, session aggregation	0.8
LLM02 Sensitive Information Disclosure	AML.T0057	Output DLP (patterns, named entities)	0.9
LLM03 Excessive Agency	AML.T0053	Tool-call policy engine (privilege *g*)	1.0
LLM04 Supply Chain	AML.T0010	Hooks for model and package scanners	0.9
LLM05 Data and Model Poisoning	AML.T0020	Dataset/model integrity hooks	0.9
LLM06 Unbounded Consumption	AML.T0029	Rate limits and token budgets	0.5
LLM07 Misinformation	—	Out of scope (optional grounding check)	0.3
LLM08 Hidden Context Exposure	AML.T0056	Canary tokens in system prompts and retrieved context	0.7
LLM09 Vector and Embedding Weaknesses	AML.T0070	RAG source provenance and access checks	0.8
LLM10 Improper Output Handling	—	Output sanitization rules (HTML, SQL, shell)	0.7

The **Risk Analyzer** computes a risk score for each event (R3):

(3)

where $p \in [0, 1]$ is the detector confidence, s is the severity of the category (Table 3), $c \in [0, 1]$ is the criticality of the affected asset, and $g \in [0, 1]$ is the privilege level of the agent or tool involved. The **Alert System** maps the score to a graded response (R4): events with $R \geq 70$ are

blocked, events with $40 \le R < 70$ raise an alert, and events with lower scores are only logged. Repeated low-risk events within one session are aggregated so that a persistent jailbreaking campaign eventually escalates. Alerts are exported to a SIEM in Common Event Format (CEF) or as JSON webhooks (R5), and the **Dashboard** displays events, trends by category and an audit trail.

Two examples illustrate the model. First, consider an indirect prompt injection found in a retrieved document that instructs an e-mail agent to forward data, with $p = 0.9$, $s = 1.0$, $c = 0.6$ and $g = 0.8$. This gives $R = 75.6$, so the request is blocked. Second, consider a jailbreak attempt in a public chatbot without tools, with $p = 0.7$, $s = 0.8$, $c = 0.3$ and $g = 0$. This gives $R = 34.3$, so the event is logged and counted toward session escalation.

4.4. Expected Results and Evaluation Plan

A prototype is being developed in Python, with the gateway implemented as an HTTP proxy compatible with the OpenAI-style chat API. It will be evaluated on public benchmarks: BIPIA for indirect prompt injection [20], the benchmark of Liu et al. for direct and combined injections [21], AgentDojo for agent hijacking [7] and HarmBench for jailbreaks [32]. The metrics will be the true positive rate, false positive rate on benign prompts, F1 score, added latency at the 95th percentile and mean time to detect. The target values for the prototype are an F1 score of at least 0.85 for prompt injection, a false positive rate below 5% on benign traffic and an added latency below 150 ms at the 95th percentile for the rule and classifier layers. These values are design targets, not measured results.

5. Discussion

The comparison answers RQ1 and RQ2. Basic monitoring and prompt injection detection are already standard, so competition is shifting toward deeper capabilities. Consolidated platforms benefit from the telemetry and SIEM ecosystems of their parent vendors, whereas independent vendors innovate faster in agent security and red teaming. The weakest function, poisoning protection, has a structural explanation: training-time poisoning [26] occurs before deployment, where a runtime gateway cannot observe it. Such protection therefore requires controls in the supply chain and the training pipeline. AI Attack Monitor covers this function only partially, through provenance checks on RAG sources, which address the runtime attack described in [6], and through hooks for external model scanning tools.

The main difference between AI Attack Monitor and the commercial solutions is transparency rather than detection accuracy. Equation (3) and the mapping in Table 3 can be inspected, tuned and audited, which supports the Measure and Manage functions of the NIST AI RMF [12] and simplifies explaining blocking decisions to stakeholders. The modular design also allows detectors to be replaced as new techniques appear, which matters because adaptive attacks can evade any single detector [21,28].

This study has several limitations. The scores are based on public documentation rather than on hands-on testing, so vendor claims may overstate or understate actual capabilities. The ordinal 0–2 scale is coarse, and the weights reflect expert judgment, although the sensitivity check showed that the leading positions are stable. The market changes rapidly, so the results reflect its state in September 2026. Finally, the architecture has not yet been evaluated empirically, and the risk score coefficients will need to be calibrated on benchmark data.

6. Conclusions

This paper compared eight AI security solutions using a standards-derived functional model and a weighted scoring model. Prisma AIRS showed the broadest coverage, and none of the solutions fully covered all eight functions. Protection against data and model poisoning and data leakage prevention were the weakest functions across the market. To address these gaps, we proposed AI Attack Monitor, a modular pipeline with a unified event schema, OWASP and ATLAS labeling, a transparent risk score and graded responses integrated with SIEM. Future work includes completing the prototype, evaluating it on the BIPIA, AgentDojo and HarmBench benchmarks, calibrating the risk coefficients and extending the architecture to model supply chain scanning.

© Central Asian Scientific Journal, 2026

References:

1. Perez, F.; Ribeiro, I. Ignore Previous Prompt: Attack Techniques for Language Models. *arXiv* **2022**, arXiv:2211.09527.

2. 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 Injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23), Copenhagen, Denmark, 30 November 2023; pp. 79–90.

3. OWASP GenAI Security Project. OWASP GenAI LLM Top 10 2026. Available online: https://genai.owasp.org/resource/owasp-genai-llm-top-10-2026/ (accessed on 10 October 2026).

4. MITRE Corporation. MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems. Available online: https://atlas.mitre.org (accessed on 10 October 2026).

5. Vassilev, A.; Oprea, A.; Fordyce, A.; Anderson, H.; Davies, X.; Hamin, M. *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*; NIST AI 100-2e2025; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025.

6. Zou, W.; Geng, R.; Wang, B.; Jia, J. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. In Proceedings of the 34th USENIX Security Symposium, Seattle, WA, USA, 13–15 August 2025.

7. Debenedetti, E.; Zhang, J.; Balunović, M.; Beurer-Kellner, L.; Fischer, M.; Tramèr, F. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In Proceedings of the Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track, Vancouver, BC, Canada, 10–15 December 2024.

8. Cisco Systems. Cisco AI Defense. Available online: https://www.cisco.com/site/us/en/products/security/ai-defense/index.html (accessed on 10 October 2026).

9. Palo Alto Networks. Prisma AIRS: AI Runtime Security. Available online: https://www.paloaltonetworks.com/ai-security/prisma-airs (accessed on 10 October 2026).

10. SentinelOne. Securing AI (Prompt Security). Available online: https://www.sentinelone.com/platform/securing-ai-prompt/ (accessed on 10 October 2026).

11. Check Point Software Technologies. Lakera: AI Security. Available online: https://www.lakera.ai (accessed on 10 October 2026).

12. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.

13. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*; NIST AI 600-1; NIST: Gaithersburg, MD, USA, 2024.

14. Google. Secure AI Framework (SAIF). Available online: https://saif.google (accessed on 10 October 2026).

15. *ISO/IEC 42001:2023*; Information Technology—Artificial Intelligence—Management System. International Organization for Standardization: Geneva, Switzerland, 2023.

16. Microsoft. Catalog of AI Attack Techniques. Available online: https://learn.microsoft.com/en-us/security/zero-trust/catalog-ai-attack-techniques/catalog-ai-attack-techniques (accessed on 10 October 2026).

17. Microsoft. AI Red Teaming. Available online: https://learn.microsoft.com/en-us/security/ai-red-team/ (accessed on 10 October 2026).

18. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly. *High-Confid. Comput.* **2024**, *4*, 100211.

19. Liu, Y.; Deng, G.; Li, Y.; Wang, K.; Wang, Z.; Wang, X.; Zhang, T.; Liu, Y.; Wang, H.; Zheng, Y.; Liu, Y. Prompt Injection Attack against LLM-Integrated Applications. *arXiv* **2023**, arXiv:2306.05499.

20. Yi, J.; Xie, Y.; Zhu, B.; Kiciman, E.; Sun, G.; Xie, X.; Wu, F. Benchmarking and Defending against Indirect Prompt Injection Attacks on Large Language Models. *arXiv* **2023**, arXiv:2312.14197.

21. Liu, Y.; Jia, Y.; Geng, R.; Jia, J.; Gong, N.Z. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. In Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 14–16 August 2024; pp. 1831–1847.

22. Wei, A.; Haghtalab, N.; Steinhardt, J. Jailbroken: How Does LLM Safety Training Fail? In Proceedings of the Advances in Neural Information Processing Systems 36 (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023.

23. Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J.Z.; Fredrikson, M. Universal and Transferable Adversarial Attacks on Aligned Language Models. *arXiv* **2023**, arXiv:2307.15043.

24. Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G.J.; Wong, E. Jailbreaking Black Box Large Language Models in Twenty Queries. *arXiv* **2023**, arXiv:2310.08419.

25. Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú.; et al. Extracting Training Data from Large Language Models. In Proceedings of the 30th USENIX Security Symposium, Virtual Event, 11–13 August 2021; pp. 2633–2650.

26. Carlini, N.; Jagielski, M.; Choquette-Choo, C.A.; Paleka, D.; Pearce, W.; Anderson, H.; Terzis, A.; Thomas, K.; Tramèr, F. Poisoning Web-Scale Training Datasets is Practical. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024.

27. Alon, G.; Kamfonas, M. Detecting Language Model Attacks with Perplexity. *arXiv* **2023**, arXiv:2308.14132.

28. Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.; Goldblum, M.; Saha, A.; Geiping, J.; Goldstein, T. Baseline Defenses for Adversarial Attacks Against Aligned Language Models. *arXiv* **2023**, arXiv:2309.00614.

29. Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; et al. Llama Guard: LLM-Based Input-Output Safeguard for Human-AI Conversations. *arXiv* **2023**, arXiv:2312.06674.

30. Rebedea, T.; Dinu, R.; Sreedhar, M.N.; Parisien, C.; Cohen, J. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Singapore, 6–10 December 2023; pp. 431–445.

31. Hines, K.; Lopez, G.; Hall, M.; Zarfati, F.; Zunger, Y.; Kiciman, E. Defending Against Indirect Prompt Injection Attacks With Spotlighting. *arXiv* **2024**, arXiv:2403.14720.

32. Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; et al. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. In Proceedings of the 41st International Conference on Machine Learning (ICML 2024), Vienna, Austria, 21–27 July 2024.

33. Robust Intelligence. AI Firewall (Archived Product Page). Available online: https://www.robustintelligence.com/platform/ai-firewall (accessed on 10 October 2026).

34. HiddenLayer. Agentic Runtime Security. Available online: https://hiddenlayer.com/agents (accessed on 10 October 2026).

35. Microsoft. AI Security Best Practices on Azure. Available online: https://learn.microsoft.com/en-us/azure/security/fundamentals/ai-security-best-practices (accessed on 10 October 2026).

36. Noma Security. AI Security Platform. Available online: https://noma.security (accessed on 10 October 2026).

© Central Asian Scientific Journal, 2026