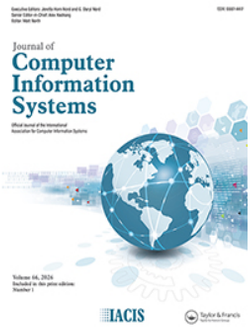

Journal of Computer Information Systems

ISSN: 0887-4417 (Print) 2380-2057 (Online) Journal homepage: www.tandfonline.com/journals/ucis20

LLM Scalability Risk for Agentic-AI and Model Supply Chain Security

Kiarash Ahi , Vaibhav Agrawal & Saeed Valizadeh

To cite this article: Kiarash Ahi , Vaibhav Agrawal & Saeed Valizadeh (17 Feb 2026): LLM Scalability Risk for Agentic-AI and Model Supply Chain Security, Journal of Computer Information Systems, DOI: 10.1080/08874417.2026.2624670

To link to this article: https://doi.org/10.1080/08874417.2026.2624670

JOURNAL OF COMPUTER INFORMATION SYSTEMS
https://doi.org/10.1080/08874417.2026.2624670

LLM Scalability Risk for Agentic-AI and Model Supply Chain Security

Kiarash Ahi[a], Vaibhav Agrawal[b], and Saeed Valizadeh[b]

[a]Virelya AI Labs, San Francisco Bay Area, CA, USA; [b]Google, Mountain View, CA, USA

ABSTRACT

Large Language Models (LLMs) & Generative AI are transforming cybersecurity, enabling both advanced defenses and new attacks. Organizations now use LLMs for threat detection, code review, and DevSecOps automation, while adversaries leverage them to produce malwares and run targeted social-engineering campaigns. This paper presents a unified analysis integrating offensive and defensive perspectives on GenAI-driven cybersecurity. Drawing on 70 academic, industry, and policy sources, it analyzes the rise of AI-facilitated threats and its implications for global security to ground necessity for scalable defensive mechanisms. We introduce two primary contributions: the LLM Scalability Risk Index (LSRI), a parametric framework to stress-test operational risks when deploying LLMs in security-critical environments & a model-supply-chain framework establishing a verifiable root of trust throughout model lifecycle. We also synthesize defense strategies from platforms like Google Play Protect, Microsoft Security Copilot and outline a governance roadmap for secure, large-scale LLM deployment.

KEYWORDS

AI governance; AI-driven malware; anomaly detection; cybersecurity; dual-use AI; explainable AI (XAI); federated learning; large language models (LLMs); zero-day detection

Introduction

The rapid evolution of artificial intelligence has placed Large Language Models (LLMs) and generative AI at the forefront of software innovation and cybersecurity transformation.[1–3] However, this widespread adoption has created a double-edged sword: LLMs empower defenders—especially platform administrators like Google Play, Apple App Store, and other enterprise app platforms—to perform static code scanning, automate threat detection, and improve code quality in real time. Yet simultaneously, those same models are exploited by attackers to generate malware, obfuscate code, and discover vulnerabilities at scale. This duality introduces complex security and governance challenges, underscoring the urgent need for systematic analysis, responsible deployment, and robust defensive frameworks.[4]

This paper presents a comprehensive survey of both the risks and opportunities associated with LLMs in cybersecurity. We explore their dual-use nature, recent industry and academic advances, and how both defenders and adversaries leverage these models for tasks such as code generation, malware design, zero-day detection, and DevSecOps, supported by architectural comparisons, benchmark studies, and cross-industry case examples. We include both standalone LLMs and emerging LLM-powered agents with autonomous planning, memory, and tool-use capabilities under the umbrella of "LLM-based cyber systems." To guide the reader, the paper is structured as follows: Methodology in Brief—We analyzed 70 peer-reviewed papers and industry datasets, built a Policy Maturity Matrix, and supply-chain risks. Methodology appear in Section II. Section III lays the foundational background by reviewing existing literature on the evolution, capabilities, and early governance efforts concerning LLMs.[5] Section IV provides an in-depth analysis of LLM applicability in security, detailing their dual-use potential, and introduces the LLM Scalability Risk Index (LSRI), a parametric mathematical framework for evaluating operational and security trade-offs in production environments. Building on this analysis, Section V presents our focal research thrust: securing the LLM model supply chain. It outlines a roadmap for establishing a "verifiable root of trust" through cryptographic attestation and semantic policy enforcement. Section VI concludes the paper by summarizing our primary two contributions, and finally, Section VII discusses the policy and practice implications for stakeholders navigating the rising complexity of LLM-powered cybersecurity ecosystems and proposing a governance roadmap rooted in explainability, federated learning, and adaptive resilience.

Methodology

This paper employs a multi-pronged methodology combining empirical analysis, literature synthesis, and policy evaluation:

CONTACT Kiarash Ahi ahi@virelya.org Virelya AI Labs, San Francisco Bay Area, California, USA

Literature Survey: We analyzed 70 peer-reviewed papers, policy documents, and technical reports published between 2018–2025, focusing on dual-use LLMs, adversarial robustness, and AI governance.

Threat Categorization: Attack vectors were organized using a structured adversarial taxonomy derived from[6–8] and cross-referenced with OWASP's LLM Top 10 and NIST AI RMF 1.0.

Policy Maturity Matrix: Governance frameworks were scored based on enforcement level, coverage of LLM-specific risks, and implementation transparency, weighted equally across five pillars.

LSRI Design: The LLM Scalability Risk Index (LSRI) was developed as a parametric framework for sensitivity-based stress testing, utilizing industrial-representative performance parameters to quantify the trade-offs between scalability, security, and compliance in high-throughput environments.

Supply-Chain Analysis and Systems Synthesis: We analyzed prior work on model integrity, data poisoning, weight tampering, and agentic vulnerabilities to define the LLM supply chain as a lifecycle-spanning system covering build-time artifacts and run-time dependencies. This synthesis informed a verifiable root-of-trust framework that treats supply-chain assurance as an enforceable systems property rather than a post-hoc governance mechanism.

Background and literature review

Evolution and capabilities of LLMs

Large Language Models (LLMs) have evolved rapidly from their initial applications in natural language translation and generation to highly capable systems supporting complex software engineering tasks. Models such as GPT-4 and PaLM now perform code generation, refactoring, debugging, and even formal verification with increasing accuracy and fluency.[1,9] These advancements are enabled by scaling transformer architectures and training on diverse programming and natural language corpora. Recent research from OpenAI and Google demonstrates how LLMs can integrate into full development pipelines, assisting with test case creation, API documentation, and dynamic bug resolution.[10–12]

Security risks and early governance efforts

The dual-use nature of LLMs has raised significant security concerns. On one hand, they can support code auditing and threat detection; on the other, they can generate obfuscated or insecure code, or be weaponized for malicious purposes. Comprehensive surveys by Yao et al. (2024) and Bryce et al. (2024) highlight these privacy and security trade-offs, categorizing the impact of LLMs across a spectrum of beneficial and adversarial outcomes.[13,14] Prior work has emphasized the need for proactive safeguards, such as Brundage et al.'s recommendations on structured red teaming and audit trails, and the European Union's Artificial Intelligence Act, which mandates risk assessments and transparency reports for high-impact models.[15,16] These frameworks aim to mitigate misuse while supporting responsible innovation.

Ethics and governance of dual-use LLMs

Integrating LLMs into CI/CD pipelines automates crucial security tasks such as code review, threat detection, and compliance enforcement. GitLab and Azure DevOps showcase how GPT based tools can enable real-time security hardening and policy enforcement.[17,18]

As indicated in Table 1, while the EU AI Act and the US NIST AI RMF represent significant strides, the global governance landscape for LLMs in cybersecurity remains dynamic, with other major technological regions developing their own distinct approaches. For instance, countries in Asia, such as China, Japan, South Korea, and Singapore, are actively formulating AI regulations and ethical guidelines that reflect their unique priorities. Understanding these varied international perspectives and fostering dialogue toward greater regulatory interoperability will be crucial for addressing the borderless nature of cyber threats and ensuring a globally coordinated response to the risks posed by dual-use AI.[15–18]

Privacy-aware deployment of LLMs via federated learning

Privacy preserving LLM deployment strategies are increasingly relevant. Federated learning allows training across distributed devices without centralizing data, aligning with laws like GDPR. Kairouz et al. and Bonawitz et al. have demonstrated that these frameworks preserve privacy while maintaining model utility.[23,24]

Explainability and trust in AI driven defense

The adoption of LLMs in automated security systems demands transparency. Explainable AI (XAI) methods like SHAP and LIME have been customized to make LLM based vulnerability classifications interpretable. These models help developers and analysts understand the rationale behind predictions, supporting auditability and compliance.[25,26]

Table 1. Policy Maturity Matrix across different regions.

Region	Governance Maturity	LLM-Specific Laws	Red Teaming Mandate	Transparency Requirements
European Union	High	AI Act (2025)	Required	Required
United States	Medium	NIST AI RMF (Voluntary)	Encouraged, not required	Limited (varies by agency)
China	High	Interim Measures (2023)[19]	Required	Required
Japan	Medium	AI Strategy (2022)[20]	Not required	Partially encouraged
South Korea	Medium	National Strategy for AI (2023)[21]	Encouraged	Voluntary guidelines
Singapore	High	Model AI Governance Framework[22]	Required	Required

Adversarial attacks and model vulnerabilities

The integration of LLMs into security critical domains has exposed them to sophisticated adversarial attacks. Carlini et al. highlighted how training data could be extracted from LLMs, undermining confidentiality.[12] Wallace et al. demonstrated that prompt injection and adversarial fine tuning can manipulate LLM outputs, evading content filters. Benchmarks such as PINT and recent tools have emerged to systematically test defenses against prompt injection and jailbreak attacks, measuring both false positives and false negatives.[27] Recent work by Jia et al. organized a global competition revealing how LLMs can be tricked into generating offensive content and misinformation, emphasizing the need for rigorous adversarial testing frameworks.[6]

Analysis of LLM applicability in security

As LLMs become deeply embedded in software development and cybersecurity pipelines, their dual-use potential has triggered increasing scrutiny. A growing body of research has documented how these models can unintentionally or deliberately produce insecure code, including cryptographic flaws, SQL injection vectors, and XSS vulnerabilities.[7,8] More alarmingly, the accessibility of LLMs has democratized the creation of deceptive content—enabling non-experts and malicious actors alike to generate phishing apps, polymorphic malware, and social engineering scripts at scale.[28–30] These developments reflect not isolated failures but systemic risks introduced by generative models when deployed without sufficient constraints. This section analyzes such risks through three lenses: (1) the emerging threat landscape shaped by misuse and amateur error, (2) industry-led defense strategies to mitigate LLM-enabled attacks, and (3) the broader governance and technical challenges that complicate safe deployment.

Amateur developers and security risks

While LLMs accelerate software creation, they have unintentionally enabled a wave of insecure development among amateur coders. By lowering technical barriers, these models allow individuals with minimal training to generate functional code that often lacks essential safety. Studies indicate that inexperienced developers frequently integrate LLM-generated snippets without validating security implications, causing common vulnerabilities—such as improper authentication and insecure API usage to proliferate in production software.[31,32]

This highlights an urgent need for LLM-integrated guardrails to proactively flag unsafe patterns for novice users. While amateur misuse stems from a lack of expertise, it sets the stage for the more calculated, scalable weaponization of generative AI by professional adversaries.

Malicious actors leveraging LLMs

Beyond accidental misuse, malicious actors are leveraging LLMs as force multipliers for deliberate cyberattacks, automating the creation of malware, phishing payloads, ransomware, and code obfuscation. Unlike traditional malware authors who required deep expertise, attackers can now generate malicious scripts with minimal effort, dramatically accelerating development cycles.[33] Recent cybersecurity reports reveal a sharp upward trend in malware generation using LLM, raising concerns about automated threat scaling and democratized access to advanced attack tools.[34]

Security researchers report that these "dark LLMs" are increasingly optimized to evade endpoint detection and static analysis through polymorphic payloads and context-aware generation.[35,36] Simultaneously, deepfake multimedia amplifies social engineering; for instance, a Ferrari executive was targeted by a CEO voice-clone, which failed only when the AI could not answer a specific question.[37]

The emergence of autonomous LLM agents marks a critical inflection point where multi-stage attacks require minimal oversight. This transition drives "Cyber Threat Inflation," characterized by drastically reduced attack costs and an industrial offensive scale.[38]

Researchers from Carnegie Mellon and Anthropic demonstrated that LLMs can autonomously plan and execute attacks, successfully replicating the 2017

Equifax breach using the Incalmo toolkit.[39] In controlled enterprise environments, these systems achieved partial or full compromise across most test networks.[40–42] By producing code that mutates to bypass static defenses, adversaries have shifted threat scalability from human-limited to industrial-scale.

Defensive utilization of LLMs in mobile app security

In response to rising AI-powered threats, mobile platform providers are embedding LLMs directly into their security workflows. One of the most effective use cases is automated code review—where LLMs augment traditional static analyzers by identifying logic flaws, unusual API usage, or obfuscated payloads that escape signature-based detection.[43–46]

These use cases demonstrate how LLMs can shift mobile app security from reactive filtering to intelligent pre-deployment screening, flagging issues before users ever download an app. However, as defensive applications of LLMs grow more powerful, they also inherit risks such as overfitting, bias, or exploitability—making explainability and continuous retraining essential.

Industry case studies: Leveraging LLMs for cyber defense

As LLM-fueled threats escalate, leading technology companies are responding by deploying proprietary AI tools to reinforce digital defenses. These platforms integrate LLMs into core security operations, including code review, static analysis, compliance auditing, and threat intelligence—tailoring strategies to align with specific infrastructure and security priorities.[43–47] Table 2 summarizes several leading companies and their defensive applications of LLM technology.

These systems represent a critical shift from reactive to proactive security postures. For instance, Google's Gemini underpins Play Protect's live threat engine, capable of analyzing millions of apps for suspicious behavior in real time. Microsoft's Security Copilot assists analysts by flagging unsafe code patterns and generating remediation steps, while Amazon's CodeWhisperer identifies vulnerabilities in IDEs during the creation process. Similarly, CrowdStrike's "Charlotte AI" automates incident prioritization to accelerate response times,[48] and Palo Alto's Cortex XDR leverages AI to unify telemetry across network and cloud environments to neutralize threats.

By embedding LLMs into their security stacks, these organizations set new industry standards for AI-augmented defense. However, these same capabilities, if left unchecked, can also empower adversaries, reinforcing the dual-use nature of LLM technology. Furthermore, leading cloud providers are piloting insurance partnerships to share data on AI-related incidents, reflecting the growing financial dimension of GenAI security. Table 2 highlights the specific LLM technologies deployed by these companies for cyber defense.

Table 2. Leading companies leveraging LLMs for security.

Company	LLM Technology	Application
Google	Gemini	Malware Detection, Static Analysis, Threat intelligence
Microsoft	GPT-4	Security Copilot, Code Review
Amazon	CodeWhisperer	Vulnerability Detection
IBM	Watsonx	Compliance & Risk Management
Palantir	AIP	Threat Hunting & Behavioral Analysis

Scalability concerns in LLM-Based security systems

Integrating LLMs into production-level security pipelines such as global app stores or CI/CD environments presents significant technical and operational challenges. Real-world deployment requires low-latency inference, cost-effective infrastructure, and high throughput across diverse architectures. For instance, scanning millions of apps in the Google Play or Apple App Store for malware necessitates robust resource allocation and distributed serving to ensure inferences complete within milliseconds while remaining resilient to adversarial inputs.

Operational overhead increases when maintaining regional consistency under varying regulations like GDPR or CCPA. Furthermore, the EU Artificial Intelligence Act explicitly mandates technical documentation and cryptographic attestation for high-impact AI systems.[49]

The OWASP Top 10 for LLM Applications addresses these scaling risks through the new "Unbounded Consumption" category.[50] This expands the traditional Denial of Service (DoS) threat to include resource mismanagement and unexpected infrastructure costs, reflecting how LLMs can be exploited to consume excessive tokens or processing power.

This creates a critical tension: while powerful, LLMs are not trivially scalable. Integration into global security infrastructure must be engineered to avoid bottlenecks and regional inconsistencies.

To quantify these trade-offs, we propose the LLM Scalability Risk Index (LSRI), formalized as a parametric risk model in Equation (1). Higher LSRI values indicate greater deployment readiness under the specified scalability, cost, and security constraints.

Equation 1

$$LSRI = \Phi \cdot \left(1 - \sum_{i=1}^{n} w_i \cdot f_{i(x_i)}\right)$$

Where:

x_i: is the observed raw metric for a specific factor (e.g., latency in ms)

$f_{i(x_i)} \in [0, 1]$: is a non-linear Risk Mapping Function that normalizes raw data into a risk score. a lower f_i indicates higher deployment readiness

w_i: represents the contextual weight assigned to each factor, where $\sum^{w} \_i = 1$

Note: For the purposes of this study, we utilize a baseline where $w_i = \frac{1}{n}$ (equal weighting) to provide a generalized assessment. However, these weights are intended to be adjusted by security architects and cross-functional teams to align with specific organizational requirements and risk tolerances.

τ, σ, λ: represent the critical performance threshold, the sensitivity of the risk gradient, and the target industrial scale, respectively, allowing the LSRI to be calibrated to specific hardware or regulatory environments. environmental calibration parameters defined in Table 3.

Φ : Integrated Integrity Multiplier, rather than a binary gate, is a continuous coefficient representing the aggregate security health of the system. It is defined as:

Equation 2

$$\Phi = \prod_{j=1}^{m} \max\left(0, 1 - \alpha_j . E_j\right)$$

Where:

- E_j is the measured violation magnitude (e.g., the specific rate of prompt-injection success or PII leakage).
- α is the sensitivity coefficient for risk category j, allowing the multiplier to scale based on the severity of the threat.

Risk Mapping Functions (f_i)

To ensure the index reflects real-world performance non-linearity, we utilize three primary mapping functions to evaluate deployment readiness:

a. Sigmoid Mapping(Latency)

Used for time-sensitive metrics where risk remains low until a critical threshold τ is reached, then increases exponentially.

Equation 3

$$f_{sig(x)} = \frac{1}{\left(1 + e^{-\left(\frac{x-\tau}{\sigma}\right)}\right)}$$

b. Exponential mapping (Throughput)

Used to model mitigation of scaling risks, where risk decreases as capacity approaches industrial levels λ.

Equation 4

$$f_{exp(x)} = e^{-\frac{x}{\lambda}}$$

Throughput values exceeding λ asymptotically reduce risk toward zero, reflecting diminishing marginal scalability risk beyond industrial baselines.

c. Step Mapping (Size)

Used for hard hardware constraints such as VRAM limits, where risk is binary based on hardware compatibility:

Equation 5

$$f_{step(x)} = \{0, x \leq \text{Threshold}; 1, x > \text{Threshold}\}$$

d. Linear Mapping (Cost, frequency, regulation):

Used for proportional resource depletion and maintenance benchmarks. Risk increases linearly with cost and decreases linearly with update frequency

Equation 6

$$f_{cost(x)} = \min\left(1, \frac{x}{Budget\ Ceiling}\right)$$

$$f_{freq(x)} = \max\left(0, 1 - \frac{x}{\text{Target Frequency}}\right)$$

Table 3. LLM scalability risk index (LSRI): deployment readiness rubric.

Risk Factor	Metric (xi)	Mapping Type	Parameter Thresholds	Mapping logic
Inference Latency	ms	Sigmoid	$\tau = 100$ms, $\sigma = 15$	(Eq. 3)
Throughput	req/day	Exponential	$\lambda = 10^6$	(Eq. 4)
Regulatory Risk	CFP	Linear	CFP ∈ [0,1]	(Eq. 8)
Model Size	Params	Step	Threshold = 20B	(Eq. 5)
Update Freq	count	Linear	Target = 2 (updates/day)	(Eq. 7)
Cost Sensitivity	USD	Linear	Ceiling = $1000	(Eq. 6)

$$f_{reg(x)} = 1 - Complaince\ Framework\ Priority$$

This formulation ensures that the index adapts to different deployment profiles while maintaining a "zero-trust" threshold for fundamental security boundaries. A model cannot achieve a "safe" score if it violates fundamental security or compliance boundaries ($\Phi = 0$), regardless of its speed or cost-efficiency.

The LSRI provides a structured, forward-looking risk score that supports comparative assessment of scalability—security stress across deployment scenarios, rather than an empirically calibrated estimator of real-world incident probability.

While LSRI quantifies operational readiness, its integrity multiplier (Φ) motivates the enforceable supply-chain guarantees developed in Section V.

Application and calibration

The LSRI assists architects in balancing the technical, legal, and economic dimensions of large-scale LLM defense. By assigning scores based on these weighted functions, security teams can establish rigorous thresholds for acceptable deployment conditions. Table 3 outlines the specific parameters used for a standard high-impact deployment (e.g., App Store security).

The Compliance Framework Priority (CFP) is a normalized value $\in [0, 1]$ representing the percentage of required regulatory controls (e.g., EU AI Act Article 10) successfully implemented. Update frequency is measured in updates per day. Throughput x is measured as sustained inference requests per day, normalized against an industrial baseline λ representing large-scale app-store or CI/CD deployment.

By assigning scores based on these weighted functions, security teams can establish rigorous thresholds for acceptable deployment conditions. While Table 3 provides baseline parameters for a high-throughput mobile ecosystem, the framework is designed for Sensitivity Analysis, allowing architects to stress-test how specific metric fluctuations (e.g., a 20% increase in latency) impact the overall deployment risk profile.

Practical validation: Worked examples and Sensitivity Analysis

To demonstrate the application of the LSRI, we evaluate two hypothetical model deployment scenarios in Table 4 using the parameters and weights defined in the rubric (Table 3).

The framework is specifically designed to be sensitive to performance "tipping points" via non-linear mapping. Table 5 illustrates how the risk score reacts to fluctuations in latency, demonstrating the Sensitivity Analysis of the Sigmoid function f_sig
with $\tau = 100$ ms, $\sigma = 15$

Regulatory compliance and privacy constraints

The deployment of LLMs in security workflows introduces complex compliance challenges under frameworks like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).[51,52] These regulations mandate strict data minimization, user consent, data residency, and the "right to explanation," all of which constrain how LLMs are trained and applied to sensitive content.

Table 4. Comparison of model readiness scenarios.

Feature	Scenario A: High Readiness	Scenario B: Low Readiness
Model Type	Optimized 15B Parameter Model	15B Model with Logic Vulnerability
Integrity Multiplier (Φ)	1 (Passes all security audits)	0.76 (with 24% prompt injection success rate)
Latency (x_1)	85ms $(f_{sig} = 0.27)$	40ms $(f_{sig} = 0.02)$
Throughput (x_2)	1.2M req/day $(f_{exp} = 0.30)$	2.0M req/day $(f_{exp} = 0.14)$
Regulatory Risk (x_3)	80% Compliance $(f_{reg} = 0.20)$	10% Compliance $(f_{reg} = 0.90)$
Model Size (x_4)	15B $(f_{step} = 0.0)$	15B $(f_{step} = 0.0)$
Update Freq (x_5)	1.5 updates/day $(f_{freq} = 0.25)$	0.2 updates/day $(f_{freq} = 0.90)$
Cost Sensitivity (x_6)	$275/day $(f_{cost} = 0.28)$	$850/day $(f_{cost} = 0.85)$
Final LSRI Score	~0.78 (Ready for Deployment)	~0.40 (Risky)

Table 5. Sensitivity analysis of latency risk Mapping.

Observed Latency (x)	Risk Score (f_{sig})	Resulting LSRI*	Qualitative Risk Impact
50 ms	0.03	0.82	Negligible: Optimal performance
100 ms (τ)	0.50	0.74	Critical Threshold: Performance warning
125 ms	0.84	0.68	High Risk: Significant UX degradation
150 ms	0.96	0.66	Failure: System functionally unusable

*LSRI values are computed by substituting the specified latency value into the Scenario A baseline in Table 4 while holding all other factors and weights constant.

For instance, LLM-based code scans or behavioral analysis may inadvertently process Personally Identifiable Information (PII), triggering legal obligations. While federated learning and on-device inference offer solutions, scaling these privacy-preserving techniques remains technically demanding and legally ambiguous.

Furthermore, fulfilling transparency requirements is difficult due to the "black-box" nature of large transformer models. Without rigorous documentation and explainable AI, organizations risk noncompliance, reputational harm, or discriminatory outcomes. To utilize LLMs in regulated environments, defenders must embed privacy-by-design principles and consent-driven architectures into every stage of deployment.

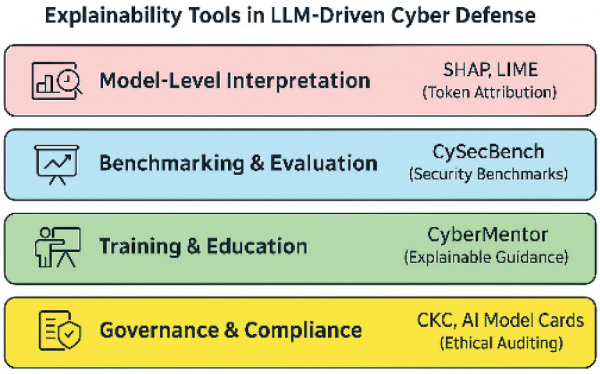

Figure 1. Categorization of explainability tools used in LLM-driven cybersecurity systems.

Explainability and trust in AI-driven defense

As LLMs take on increasingly autonomous roles in cybersecurity—classifying vulnerabilities, triaging threats, or flagging anomalies—the need for explainable artificial intelligence (XAI) has become paramount. Without transparency into how these decisions are made, stakeholders may lose confidence in AI-driven defense systems, especially when they impact compliance, reputation, or user rights.

To bridge this gap, researchers have adapted traditional XAI techniques such as SHAP and LIME to LLMs, enabling visibility into influential tokens, attention patterns, and decision pathways.[53] These interpretations not only enhance trust but also help security analysts validate model behavior, identify edge-case failures, and fine-tune thresholds for deployment. The major categories of explainability tools and their use cases in security pipelines are summarized in Figure 1.

In educational settings, tools like CyberMentor,[54] use Retrieval-Augmented Generation (RAG) and agentic workflows to provide interpretable feedback. These systems teach not just what a threat is, but its underlying mechanics.

Beyond technical utility, the dual-use nature of LLMs necessitates ethical auditing. Frameworks like the Cyber Kill Chain (CKC) and AI model cards are used to document decision logic and misuse potential in auditable formats. As noted by Barrett et al.[55] and Gupta et al.,[56] explainability is critical infrastructure for regulatory compliance and long-term trust in AI-powered defense.

Federated learning and privacy-aware deployment of LLMs

As LLMs increasingly handle sensitive data in mobile and edge environments, ensuring privacy without compromising performance is a top priority. Federated Learning (FL) offers a promising paradigm by enabling decentralized training without transferring raw data to central servers. This approach aligns with GDPR and CCPA regulations regarding data locality and minimization.[23]

Kairouz et al.[23] provided foundational analysis of FL's scalability and security trade-offs, while Bonawitz et al.[24] demonstrated large-scale implementation using secure aggregation protocols. By integrating LLMs with FL, security tools can perform real-time anomaly detection and on-device code analysis without transmitting data to the cloud. This architecture minimizes centralized breach risks and promotes compliance-by-design (see Figure 2).

Hybrid models are emerging that combine FL with on-device fine-tuning, allowing devices to benefit from shared intelligence while customizing insights for local threats. For instance, mobile security platforms using edge-deployed LLMs can detect suspicious behaviors without exposing private logs or PII to external servers.

Complementary techniques like differential privacy and homomorphic encryption further harden FL pipelines against inference and model inversion threats. These layered approaches ensure privacy, accountability, and robustness against sophisticated adversaries. Ultimately, FL is a critical enabler for trustworthy AI, allowing defenders to leverage LLMs while navigating the legal and technical constraints of real-world deployment.

Detection of zero-day vulnerabilities

Zero-day vulnerabilities often bypass traditional rule-based detection, but Large Language Models (LLMs) offer a semantic, context-aware alternative. For

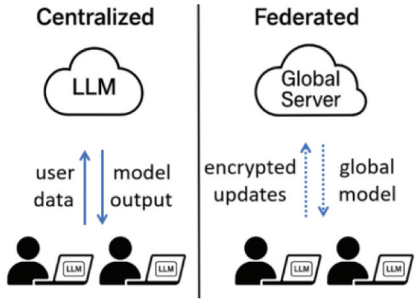

Figure 2. Architectural comparison between centralized and federated LLM deployment. In centralized systems, user data is transmitted directly to a cloud-based LLM for processing—raising privacy, security, and compliance risks. In contrast, federated learning allows users to train models locally and share only encrypted model updates with a global server, preserving data locality and enabling privacy-aware AI deployment. This distinction is crucial in regulated environments where sensitive user data cannot be exported or stored externally.

example, Google's Big Sleep project recently identified a zero-day in SQLite using LLM-driven analysis.[57] Research by Lisha et al.[58] demonstrates that LLMs trained on vulnerability-specific corpora outperform static analyzers in detecting complex logic and control-flow flaws. While these models also assist in predicting exploit propagation paths, researchers warn that over-reliance on synthetic training data can lead to "model collapse," reducing effectiveness against rare, critical vulnerabilities.[59]

In practice, these techniques are integrated into CI/CD pipelines. GitHub's code scanning and Google's Play Protect, for example, experiment with LLM-powered models to detect anomalies in obfuscated binaries. LLMs are also applied in fuzzing, automatically generating exploit-oriented test cases to surface weaknesses preemptively. Figure 3 contrasts traditional detection pipelines with LLM-based approaches.

Ultimately, this capability reinforces the dual-use nature of LLMs: while defenders gain tools for neutralizing unknown threats, attackers could fine-tune models to identify zero-day opportunities faster. This semantic power makes zero-day detection a critical battleground in AI-driven cyber defense.

Table 6 compares traditional static analyzers with LLM-enhanced vulnerability detection across four key metrics. Relative to the static-analyzer baseline (62% recall), LLM systems deliver 26–29%-point absolute gains—a ≈42–47% relative improvement—while maintaining similar latency and false-positive rates.

LLMS in DevSecOps automation

As software delivery accelerates, security must evolve to match the speed of continuous integration and deployment. DevSecOps—the integration of security directly into DevOps workflows—demands automation, precision, and scale across the entire software development lifecycle. LLMs are increasingly being leveraged to meet this need, embedding intelligence into every stage of the pipeline.

In modern DevSecOps environments, LLMs assist in:

- Code scanning at every commit, flagging insecure patterns and suggesting remediations in real time.
- Assessing containerized builds for compliance with internal and external security policies.
- Analyzing dependency trees to identify vulnerable or outdated libraries before code reaches production.

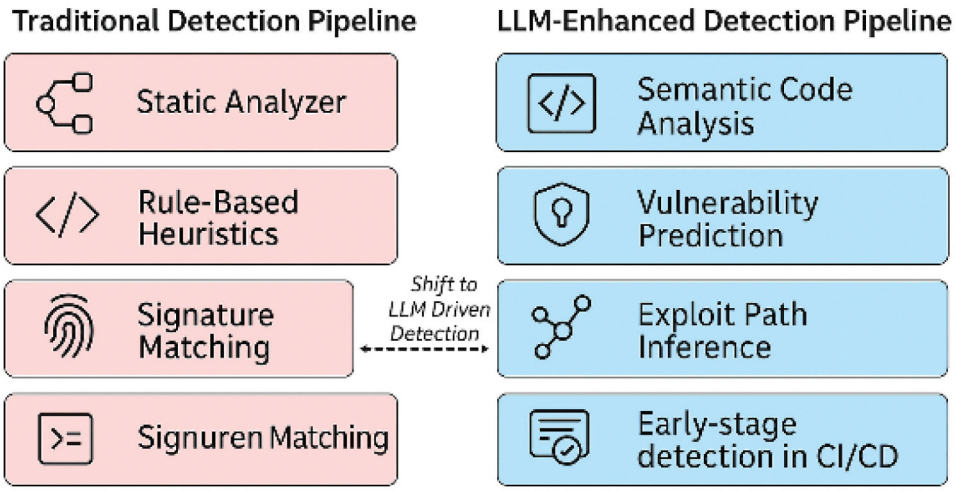

Figure 3. Comparison between traditional and LLM-enhanced zero-day vulnerability detection pipelines.

Table 6. Performance comparison of traditional static analyzers and LLM-enhanced zero-day vulnerability detectors.

Method	Recall (Zero-Day)	Avg Latency (ms)	False Positives	Interpretability
Static Analyzers	62%	80	11%	Limited
LLM + Symbolic Hybrid	88%	105	13%	Moderate (SHAP)
LLM + Graph-Based	91%	96	12%	High (CySecBench)

Prominent platforms have already begun integrating these capabilities. GitLab's Auto DevSecOps system employs GPT-based models for dynamic scanning and compliance-as-code enforcement. Similarly, Microsoft's Azure DevOps, in collaboration with OpenAI, leverages LLMs for predictive vulnerability scoring, contextual remediation advice, and automated security testing.

These integrations shift security from a reactive checkpoint to a proactive, continuous layer—built directly into the tooling developers already use. This minimizes friction, shortens feedback loops, and enables security-by-default at scale.

At the same time, this growing reliance on LLMs in DevSecOps pipelines highlights the broader theme of this paper: the dual-use nature of AI in security. AI powered security testing agent like XBOW, currently leading HackerOne leaderboard, also highlights the dual-use nature, as a capability that can be used for both finding security vulnerabilities in applications and responsibly reporting them and using it for malicious purposes. The same models that harden pipelines could be exploited if misconfigured, biased, or insufficiently governed—making LLM observability, explainability, and governance as important as their functional accuracy.

Ethics and governance of dual-use LLMs

As LLM capabilities scale, their misuse potential grows alongside their utility. This creates a dual-use dilemma: models powering security auditing and malware detection can also generate polymorphic malware or optimize phishing campaigns. Such high-stakes symmetry necessitates governance frameworks as adaptable as the technology itself.

Brundage et al.[16] have proposed concrete mechanisms to address these risks, including:

- Structured red teaming to stress-test model behavior against adversarial use cases,
- Staged release strategies to control the dissemination of high-risk capabilities, and
- Model Watermarking can also be used for responsible posture, by allowing for the traceability of content generated by a model.[60]
- Model evaluation cards and Human-in-the-Loop (HITL) systems to provide oversight and document safety constraints.

These strategies are being codified in the EU AI Act and the U.S. NIST AI Risk Management Framework, which mandate transparency in development and auditability of training data. These policies aim to shift deployment toward accountability-by-design.

Ethical research further emphasizes value alignment in security domains. Techniques like Reinforcement Learning with Human Feedback (RLHF) are being adapted to teach LLMs to:

- Reject harmful or manipulative queries,
- Disclose uncertainty in high-risk scenarios, and
- Explain security decisions with interpretable confidence bounds.

At the international level, coalitions such as the Global Partnership on AI (GPAI) and recent AI safety summits have introduced shared guardrails, including:

- Pre-registration of frontier models,
- Mandatory incident reporting, and
- Centralized auditing repositories to detect and flag unsafe usage patterns.

These governance efforts are not merely bureaucratic safeguards, they are essential infrastructure for responsibly integrating LLMs into national security, digital forensics, and trust-sensitive ecosystems. Figure 4 illustrates the timeline of key governance milestones that have emerged between 2023 and 2025, highlighting a growing global effort to institutionalize safety practices around powerful LLMs.

Securing defensive LLM systems

As LLMs become integral components of cybersecurity infrastructure itself (e.g., in threat detection, code analysis, and incident response), their own security posture becomes paramount. Protecting these "defender" LLMs from targeted attacks is crucial to maintain their efficacy and trustworthiness. Key considerations in safeguarding these sentinel AI systems include:

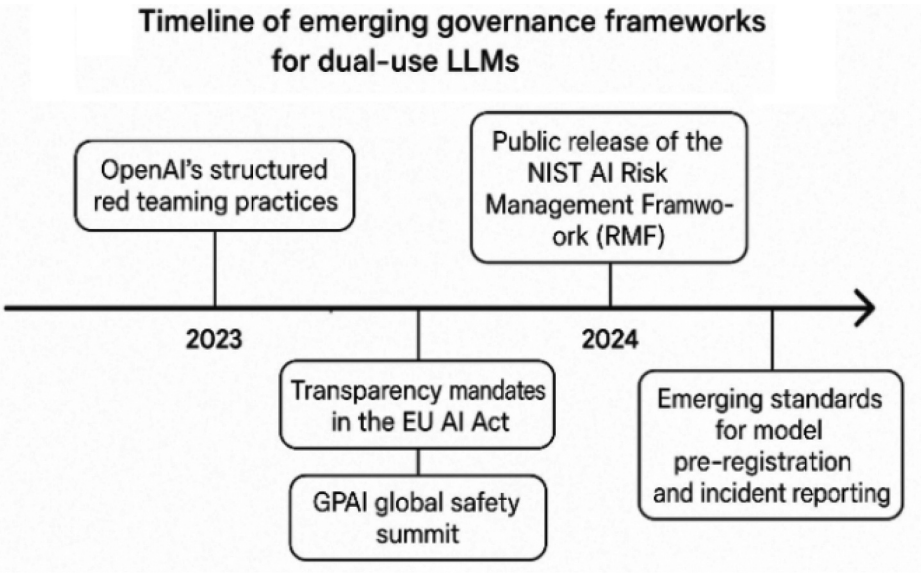

Figure 4. Timeline of emerging governance frameworks for dual-use LLMs, spanning initiatives from 2023 to 2025. Key milestones include OpenAI's structured red teaming practices, the public release of the NIST AI risk management framework (RMF), transparency mandates in the EU AI Act, the GPAI global safety summit, and emerging standards for model pre-registration and incident reporting. Together, these efforts represent a global shift toward enforceable AI safety, accountability, and dual-use risk mitigation.

Training Data Integrity and Poisoning Defense: Ensuring the provenance and integrity of data used to train and fine-tune security LLMs to prevent sophisticated poisoning attacks that could create blind spots or backdoors.[61]

Model Evasion and Robustness: Continuously evaluating and hardening defensive LLMs against adversarial evasion techniques specifically designed to bypass AI-based detection.[6,12] Fine- tuned LLM models with a labeled dataset can be used to detect against prompt injection attacks.[62] The INJECAGENT benchmark, for example, demonstrates that tool-integrated LLM agents are vulnerable in many scenarios, with attack success rates of ~24% under certain indirect prompt injection settings.[17]

Model Confidentiality and Integrity: Protecting the proprietary architecture and weights of security LLMs from extraction,[12] and ensuring their operational integrity against unauthorized modifications.

Secure Deployment and Monitoring: Implementing secure deployment practices for LLM-based security tools, including robust access controls, audit trails, and continuous monitoring for anomalous behavior or potential compromise of the AI system itself.[9]

These findings underscore that security LLMs must be protected with the same rigor as the systems they defend. Despite alignment tuning, OpenAI's red teaming efforts have shown that LLMs still exhibit failure modes under adversarial prompting and jailbreak scenarios.[63]

Model-supply chain security

While all these risks manifest across multiple layers from prompt injection and adversarial evasion to governance and compliance failures, many of them ultimately trace back to weaknesses in how LLMs are built and distributed. The LLM supply chain from data collection to model release is a critical vector for compromise. Attackers can inject poisoned samples during pre-training, introduce malicious fine-tuning data, or distribute backdoored weights via public repositories. Recent research shows minor dataset manipulations can induce persistent "logic bombs" that evade traditional red-teaming.[61]

Tramèr et al. highlight the risk of tampered checkpoints and propose cryptographic signing of training pipelines to ensure integrity.[62] Similarly, Goldblum et al. demonstrate automated detection of Trojaned models and recommend mandatory weight attestation for high-risk systems.[63]

Defending AI-driven platforms requires securing the entire lifecycle, not just inference endpoints. Figure 5 illustrates this end-to-end supply chain, highlighting potential compromise points and recommended defenses to ensure model provenance within the supply chain and security.

While the LSRI provides a parametric framework for evaluating operational readiness, the validity of its "Integrity Multiplier" Φ depends on the verifiable security of the model's origin. This necessitates a transition from external performance evaluation to internal enforceable supply-chain assurance.

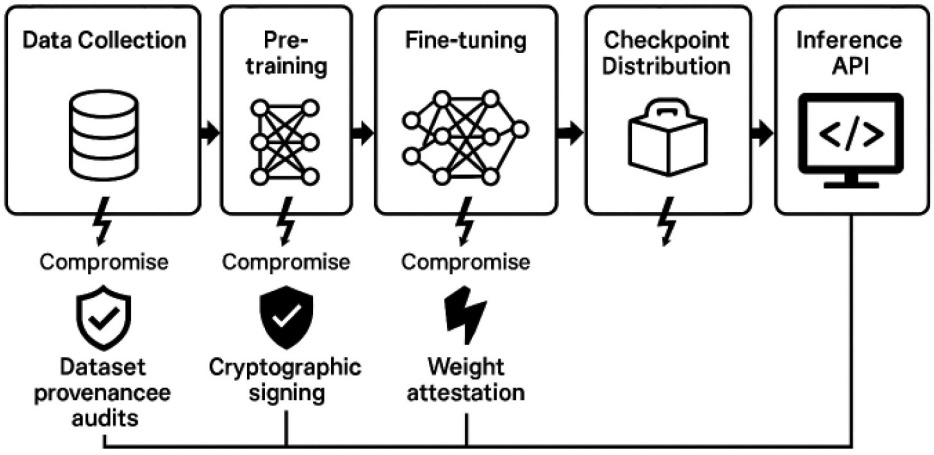

Figure 5. End-to-end LLM model-supply chain showing potential compromise points (data collection, pre-training, fine-tuning, checkpoint distribution, inference API) and recommended defenses such as dataset provenance audits, cryptographic signing, and weight attestation.[61–65].

Focal research thrust: Securing the LLM model supply chain

This section advances the research claim that security and governance of large language models cannot be meaningfully achieved through post-deployment monitoring or policy mechanisms alone. Instead, enforceable, pre-execution supply-chain guarantees spanning both build-time artifacts and run-time agentic dependencies, are a necessary condition for scalable deployment, operational safety, and effective governance of LLM-based systems. We formalize this claim through a verifiable root-of-trust architecture and demonstrate its feasibility using existing cryptographic primitives.

Unlike traditional software, LLM supply chains allow for latent vulnerabilities introduced during foundational stages to remain dormant until execution. Security therefore spans two coupled phases: build-time (trust in data and training) and run-time (interaction with tools and networks). Failures at build time propagate into autonomous agent behavior, necessitating a verifiable root of trust for AI systems.

This section grounds the supply-chain research thrust in a focused proof-of-concept (PoC) demonstrating how existing cryptographic tooling can already establish a verifiable root of trust for LLM artifacts. The remainder of the section extends this concrete baseline toward broader challenges in provenance, distribution, and agentic robustness, reframing them as incremental extensions.

Proposed supply-chain assurance architecture

Figure 6 presents a reference architecture for LLM supply-chain assurance inspired by modern software supply-chain security practices (e.g., SLSA, in-toto), adapted to the scale and opacity of foundation models. The architecture introduces three enforceable control points:

(1) Cryptographic attestation of model artifacts at the completion of training or fine-tuning.
(2) Signed provenance metadata binding model weights to their training context, datasets, and alignment configuration.
(3) Mandatory verification gates enforced prior to deployment or inference.

Together, these controls establish a fail-closed trust boundary: a model that cannot be cryptographically verified against approved provenance and policy constraints is prevented from executing, regardless of performance or cost considerations.

Prototype implementation: Cryptographic weight attestation

To demonstrate feasibility, we implemented a minimal prototype for cryptographic checkpoint attestation using existing open-source tooling commonly employed in software supply-chain security.

Build-time attestation. Upon completion of model fine-tuning, the resulting checkpoint (e.g., `model.ckpt`) is hashed using SHA-256. The hash is then signed using Sigstore, binding the artifact to an ephemeral signing identity derived from the CI/CD environment.[66] Alongside the signature, structured metadata such as model version, dataset identifier, alignment policy version, and training

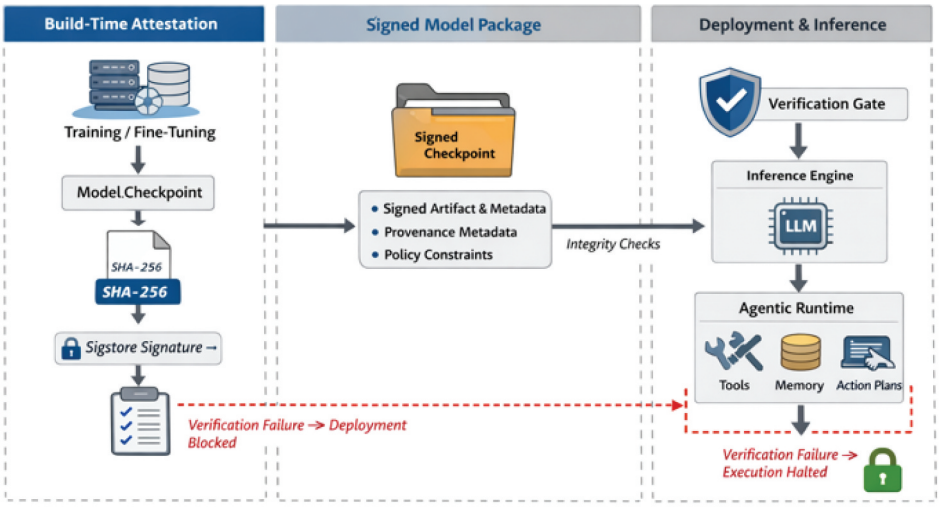

Figure 6. Proposed verifiable root-of-trust architecture for LLM supply-chain security. The architecture enforces pre-execution verification across the model lifecycle, binding build-time provenance to signed metadata and verifying model artifacts and dependencies during distribution and deployment to ensure fail-closed execution.

timestamp—is recorded in an append-only transparency log.

Run-time verification. Prior to loading the model for inference, the serving environment verifies (i) that the checkpoint hash matches the signed digest, (ii) that the signature is valid and anchored in the transparency log, and (iii) that the attested metadata satisfies deployment policy (e.g., approved dataset lineage or safety configuration). If any verification step fails, model loading is aborted.

This prototype requires no modification to model internals and introduces negligible overhead at inference time. Importantly, it demonstrates that cryptographic weight attestation is immediately deployable within existing MLOps pipelines, transforming supply-chain assurance from a policy aspiration into an enforceable technical control.

Verifiable provenance and data integrity

Building on the checkpoint-attestation prototype, verifiable provenance emerges as a critical extension of supply-chain assurance. While the PoC binds a deployed model to a cryptographically verified artifact, it does not yet capture why a model behaves as it does—namely, which datasets, transformations, and alignment steps influenced its training.

In practice, extending the architecture requires binding dataset identifiers, preprocessing pipelines, and fine-tuning stages to signed metadata associated with the final checkpoint. Such provenance records allow downstream consumers to verify not only that a model's training lineage is intact, but that it was trained under approved data and policy constraints. This capability is particularly important for mitigating data poisoning and backdoor insertion attacks, where small training-time manipulations can induce persistent, latent behaviors.[67,68]

By anchoring data provenance to the same cryptographic attestation framework used for model weights, integrity guarantees extend beyond file-level verification to encompass the semantic origins of model behavior. In practice, these guarantees can be further strengthened by committing cryptographic hashes of training datasets to an append-only transparency log prior to the commencement of training, preventing retroactive modification of provenance metadata and ensuring that data lineage is immutable from inception. These provenance guarantees establish trust in how a model was created, but must be preserved as the model is distributed and deployed across operational environments.

Secure model distribution under attested supply chains

While the prototype secures a single checkpoint at rest, real-world deployments involve model redistribution and downstream fine-tuning across multiple organizational and infrastructural boundaries. As models traverse registries, mirrors, or undergo

downstream fine-tuning, attestation can be applied recursively, binding each derived model to the cryptographically verified state of its predecessor. This preserves a transitive chain of trust across distribution boundaries, ensuring that security guarantees remain intact as models evolve across operational environments. In such environments, traditional file-level checksums provide insufficient protection, as adversaries may substitute or subtly modify weights while preserving apparent functionality.

Extending the prototype into a full weight-attestation framework requires enforcing signature verification at each distribution boundary and rejecting unsigned or policy-noncompliant artifacts during deployment. This approach ensures that every executable model instance remains cryptographically bound to its originating training pipeline and approved configuration.

This mechanism contrasts with model watermarking, which embeds detectable patterns into model outputs for post-hoc attribution. While watermarking can assist in tracing misuse after deployment, it does not prevent tampered or malicious models from executing and can often be removed or weakened through fine-tuning.[69] Weight attestation, by contrast, enforces integrity prior to execution, aligning more directly with zero-trust supply-chain principles and providing enforceable pre-execution guarantees against checkpoint substitution and supply-chain tampering.

Robustness against agentic LLM attacks

While weight attestation secures the static integrity of LLM artifacts, agentic LLM systems introduce a dynamic supply-chain surface at runtime. Tool access, persistent memory, and autonomous planning effectively extend the supply chain beyond model weights into execution-time dependencies.[67]

From a systems perspective, agentic robustness represents the runtime continuation of build-time trust guarantees. Practical extensions of the attestation architecture include multi-stage plan validation, where an agent's proposed action sequence is checked against dependency trust policies and privilege boundaries prior to execution. Similarly, tool invocation and external API access can be gated by signed manifests and runtime verification, preventing compromised tools from corrupting agent behavior.

Persistent memory introduces an additional integrity risk: malicious prompts or artifacts may poison agent state long after initial deployment. Addressing this risk requires constraint-aware memory management and continuous validation of stored context against policy and provenance metadata. Together, these mechanisms extend supply-chain assurance from static artifacts to autonomous behavior.

From proof-of-concept to scalable, enforceable assurance

The presented prototype demonstrates that cryptographic weight attestation and verification are already achievable using existing tooling. Scaling this approach to frontier-scale models and continuously evolving agentic systems introduces open challenges, including transparency-log scalability, semantic binding of safety properties to cryptographic attestations, and continuous enforcement under model updates. For frontier-scale models, naive hashing of multi-gigabyte checkpoints can introduce significant I/O overhead that negatively impacts deployment latency. Merkle-tree—based partial attestation provides a practical refinement, enabling parallelized verification of model shards or layers without rehashing the entire artifact. This approach preserves integrity guarantees while mitigating verification costs, directly addressing scalability constraints in high-throughput deployment pipelines.

Crucially, these challenges build directly upon the PoC architecture described above. Rather than representing a speculative research agenda, they define a sequence of incremental engineering extensions that can be deployed, evaluated, and strengthened as LLM-based security systems scale. In this sense, LLM supply-chain security transitions from an abstract governance concern to a concrete systems problem with enforceable guarantees.

Reframing prior cybersecurity trajectories

The proposed research agenda does not emerge in isolation, but rather builds upon and fundamentally reinterprets several foundational cybersecurity research trajectories. However, the integration of large language models into security-critical workflows violates key assumptions underlying prior approaches, necessitating new theoretical and methodological directions.

(1) **From Adversarial ML to Cyber Threat Inflation**: Traditional adversarial machine learning has largely focused on perturbation-based attacks and evasion of fixed classifiers under bounded threat models. These frameworks assume that adversarial effort scales linearly with attack complexity and that model misuse requires significant expertise. LLMs invalidate these assumptions by enabling the low-cost,

automated generation of polymorphic malware, exploits, and social-engineering artifacts. As a result, the dominant challenge is no longer isolated evasion, but cyber threat inflation, where the marginal cost of producing diverse and adaptive attacks approaches zero. Addressing this shift requires research agendas that emphasize systemic resilience, rate-limiting of attack generation, and defenses robust to continuously evolving threat distributions rather than static adversarial examples.

(2) **From Zero-Trust to Semantic Trust**: Traditional Zero-Trust Architectures (ZTA) have traditionally centered on identity verification, authentication, and access control, operating under the assumption that software artifacts are static, human-authored, and auditable prior to execution. LLM-generated code and agentic behaviors violate these assumptions by introducing probabilistic, opaque, and dynamically synthesized artifacts whose intent may not be inferable from syntax or origin alone. In LLM-integrated systems, trust decisions must therefore extend beyond identity and provenance to encompass semantic trust: continuous verification of the intent, side effects, and policy compliance of AI-generated actions at run time. This shift motivates new research into semantic policy enforcement, intent verification, and dynamic trust evaluation for AI-driven systems.

(3) **From Static Robustness to Adaptive Resilience**: Conventional cyber defenses rely heavily on static signatures, fixed rule sets, and periodic retraining cycles, reflecting an assumption that threat evolution is incremental and observable. LLM-enabled attackers undermine this model by rapidly generating novel attack variants and adapting behaviors in response to deployed defenses. Similarly, LLM-based defensive systems may themselves evolve through continual learning, fine-tuning, or agentic feedback loops. These dynamics require a move toward adaptive and lifelong robustness, where defensive mechanisms continuously update their detection logic, threat models, and trust assumptions in response to both environmental changes and emergent supply-chain vulnerabilities.

Conclusion and research implications

The integration of large language models into cybersecurity represents a structural shift in both the threat landscape and the defensive toolkit. As this paper has argued, LLMs simultaneously amplify defensive capabilities such as, automated vulnerability discovery, code analysis, and security orchestration, while dramatically lowering the cost, expertise, and scale required for sophisticated cyberattacks. This dual-use dynamic accelerates cyber threat inflation, widens the asymmetry between attackers and defenders, and exposes new classes of systemic risk that cannot be addressed through incremental extensions of existing security frameworks.

This work makes two primary contributions that together motivate a focused research agenda. First, we introduce the LLM Scalability Risk Index (LSRI), which provides a structured lens for stress-testing the operational, economic, and compliance trade-offs associated with deploying LLMs in security-critical environments. Second, we develop a model-supply-chain security architecture that establishes a verifiable root of trust across data acquisition, training, and deployment, offering a unifying perspective on how vulnerabilities introduced at early stages can propagate and amplify downstream.

More broadly, this paper situates LLM-enabled cybersecurity challenges within the lineage of prior research agendas, including adversarial machine learning, zero-trust architectures, and cyber resilience. While these paradigms remain essential, LLMs fundamentally transform their underlying assumptions by introducing probabilistic generation, opaque decision-making, and autonomous action at scale. As a result, securing AI-integrated systems demands research that moves beyond isolated attack classes or defensive techniques toward a systems-oriented understanding of trust, robustness, and resilience across the entire LLM lifecycle. In this sense, the paper contributes toward shifting the community from descriptive taxonomies of LLM risks toward a coherent and actionable research program for AI-enabled cyber defense.

Additionally, the feasibility of AI governance is not purely a policy question, but is fundamentally constrained by the technical properties of LLM systems.

Design Constraints for Feasible AI Governance: Although governance mechanisms for LLM-enabled cybersecurity systems are often framed as regulatory or institutional challenges, the feasibility of governance for LLM-enabled security depends on system-level guarantees. Effectiveness requires alignment with verifiable, scalable technical mechanisms. Any practically feasible approach must satisfy three core constraints:

(1) Verifiability: Governance must rely on verifiable evidence rather than self-attestation. This necessitates auditing the LLM lifecycle, including provenance and fine-tuning through cryptographically

verifiable artifacts like dataset commitments and weight attestation. Without these, oversight remains manual and unscalable.

(2) Scalability: Frameworks must match the scale of model size and autonomy. Ad hoc risk assessments cannot keep pace with continuous fine-tuning and reuse. Feasible governance requires automation-friendly mechanisms, such as secure aggregation and machine-verifiable compliance signals, to minimize human-intensive overhead

(3) Adaptivity: Static compliance checklists cannot keep pace with agentic threat evolution. Effective governance must support Adaptive Enforcement, where trust assumptions—quantified via parametric models like the LSRI (Eq. 1), evolve in response to real-time telemetry such as latency shifts or supply-chain integrity alerts. This ensures that governance is a dynamic runtime process rather than an offline audit.

Taken together, these constraints highlight that governance feasibility is inseparable from advances in technical assurance. Rather than viewing governance as an external layer imposed on AI systems, these constraints emphasize the need for co-design between governance mechanisms and supply-chain security primitives, including provenance tracking, cryptographic integrity checks, and runtime monitoring.

This research agenda also highlights the need to formalize a standardized AI Model Bill of Materials (AI-BOM) to provide a machine-readable format for the signed metadata and provenance records proposed in this work, facilitating interoperability across security tooling and governance frameworks. In addition, future research should characterize the performance—security Pareto frontier, quantifying the trade-offs between cryptographic verification overhead and gains in integrity assurance. Understanding these trade-offs is essential for the practical adoption of enforceable supply-chain guarantees in large-scale AI deployments.

Beyond build-time and pre-execution controls, future research may explore complementary runtime assurance techniques for long-lived agentic sessions. One possible direction is the use of periodic heartbeat checks, where a secondary, lightweight verifier model audits an agent's recent action plans against its signed safety policies. Importantly, such mechanisms are not substitutes for enforceable supply-chain guarantees, but potential secondary defenses for detecting semantic drift or policy violations that emerge during extended autonomous operation.

Policy and practice implications

While the primary contribution of this paper is research-facing, the proposed agenda carries important implications for practitioners, organizations, and policymakers. As large language models are increasingly embedded into security-critical workflows, governance and assurance mechanisms must evolve to account for risks arising from opaque training pipelines, third-party model dependencies, and autonomous agent behavior.

(1) For Policymakers and Regulators: Efforts should focus on establishing agile and internationally harmonized regulatory frameworks that encourage responsible AI innovation while mandating baseline security, transparency, and accountability standards for high-risk LLM applications in cybersecurity. Public—private partnerships play a critical role in enabling information sharing and aligning regulatory expectations with operational realities. For example, the UK National Cyber Security Center's recent enterprise-level guidance on LLM usage emphasizes prompt sanitization, API handling, and audit logging as practical risk-reduction measures.[70]

(2) For Security Organizations and Practitioners (CISOs, SecOps Teams): Organizations should prioritize comprehensive strategies for integrating LLMs into security workflows, including rigorous testing and validation of AI-enabled tools, continuous red-teaming against AI-augmented threats, and workforce upskilling to manage model behavior, dependencies, and failure modes. Supply-chain visibility and runtime monitoring should be treated as core security functions rather than optional enhancements.

(3) For LLM Developers and AI Researchers: Developers should emphasize security-by-design principles throughout the LLM lifecycle, from data curation and training to deployment and monitoring. This includes investment in safer model architectures, bias detection and mitigation techniques tailored to security contexts, and robust mechanisms for content authenticity and provenance to counter AI-generated disinformation and malware.

Ultimately, securing the future of LLM-enabled cybersecurity systems is not solely a technical challenge, but a socio-technical one that requires alignment between research, engineering practice, and governance. Key

Table 7. Recommended immediate actions for key stakeholder groups to mitigate dual-use risks of LLMs.

Stakeholder	What They Must Do Now
Governments	Enact dual-use-specific AI policies & register frontier models
Enterprises (CISOs)	Integrate LLMs into SecOps with real-time XAI + red teaming
Researchers	Build verifiable benchmarks for LLM explainability and zero-day detection
App Platforms	Embed LLMs into app review pipelines with FL + differential privacy
LLM Developers	Secure training pipelines, sign checkpoints, and release model cards

near-term actions for major stakeholder groups are summarized in Table 7.

Author contributions

CRediT: **Kiarash Ahi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Vaibhav Agrawal:** Conceptualization, Data curation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing; **Saeed Valizadeh:** Conceptualization, Formal analysis, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

1. OpenAI. Gpt-4 technical report. arXiv:2303.08774. 2023.
2. Google AI Blog. Gemini overview. Google LLC; 2023.
3. Microsoft. Introducing security copilot. Microsoft; 2023.
4. Gartner. Emerging security risks with AI. Gartner Report; 2023.
5. Zhang J, Bu H, Wen H, Liu Y, Fei H, Xi R, Li L, Yang Y, Zhu H, Meng D. When LLMs meet cybersecurity: a systematic literature review. Cybersecurity. 2025;8 (1):55. doi: 10.1186/s42400-025-00361-w.
6. Jia X, Dong Y, Liu Z, Zhu J, Chen J, Su H. Global challenge for safe and secure LLMs track 1. arXiv:2411.14502. 2024.
7. Zhang X, Wang X, Pang W, Zhang X. Policy enforcement in app stores. Ieee Tse. 2020;46:1005–1025.
8. Jiang L, Sheng VS, Liu S, Xu J. Fake review detection. Acm Cikm. 2019;28:2657–2665.
9. Narayanan S, Santhosh Kumar SVN, Dakshinamurthy J. Dynamic analysis of malicious apps. IEEE Access. 2023;11:21545–21558.
10. Pearce K, Ahmad B, Tan B, Dolan-Gavitt B, Karri R. Automated code generation security risks. IEEE Secur Privacy. 2022;20:30–40.
11. Chen J, Huang J, Jeng JE. Static analysis using LLMs. IEEE Security & Privacy. 2023;21: 92–96.
12. Carlini N, Tramèr F, Wallace E, Jagielski M, Herbert-Voss A, Lee K, Roberts A, Brown T, Song D, Erlingsson Ú, et al. Risks of LLM data leakage. USENIX Secur. 2021;30:2633–2650.
13. 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;2(2):100211. doi: 10.1016/j.hcc.2024.100211.
14. Bryce C, Kalousis A, Leroux I, Madinier H, Pasche T, Ruch P. Exploring the dual role of LLMs in cybersecurity: threats and defenses. In: Vorobeychik Y, Kantarcioglu M, editors. Large language models in cybersecurity. Springer; 2024. p. 235–242. doi: 10.1007/978-3-031-54827-7_26.
15. European Commission. Artificial intelligence act. EU; 2023.
16. Brundage M, Avin S, Clark J, Toner H, Eckersley P, Garfinkel B, Dafoe A, Scharre P, Zeitzoff T, Filar B, et al. Toward trustworthy AI development. arXiv:2004.07213. 2020.
17. GitLab. Auto DevSecOps powered by AI. GitLab Docs; 2023.
18. Microsoft Azure. Secure development lifecycle. Azure Blog; 2023.
19. Cyberspace Administration of China. Interim measures for the management of generative artificial intelligence services. Beijing China; 2023.
20. Cabinet Office. Government of Japan. AI strategy 2022. Tokyo Japan; 2022.
21. Ministry of Science and ICT, Republic of Korea. National strategy for artificial intelligence. Seoul Korea; 2023.
22. Infocomm Media Development Authority (IMDA) Singapore. Model AI governance framework. 2023.
23. Kairouz P, McMahan HB. Advances and open problems in federated learning. Found Trends ML. 2021;14(1–2):1–210. doi: 10.1561/2200000083.
24. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečný J, Mazzocchi S, McMahan B, et al. Towards federated learning at scale. SysML. 2019;1:1–5.
25. Ribeiro MT, Singh S, Guestrin. Why should I trust you? Explaining the predictions of any classifier. KDD. 2016;22: 1135–1144.
26. Samek W, Montavon G, Lapuschkin S, Anders CJ, Müller KR. Explainable artificial intelligence. Springer; 2021.
27. Zhu H, Liu J, Yi P, Zhao X. Pi-bench: evaluating the robustness of large language models to prompt injection. arXiv:2402.00349. 2024.

28. Cybersecurity Ventures. Cybersecurity almanac: 100 facts, figures, predictions and statistics. Cybercrime Magazine. 2024 June 26.
29. Amazon AWS. Automated vulnerability detection. AWS; 2023.
30. IBM Research. Watsonx security applications. IBM; 2023.
31. Pearce K, Ahmad B, Tan B, Dolan-Gavitt B, Karri R. Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. Proc IEEE Symp Security and Privacy. 2022;43:754–768.
32. Sandoval S, Pearce H, Nys T, Mansouri M, Dolan-Gavitt B. Security implications of AI-generated code. Proc ACM Int Conf Foundations of Software Engineering (FSE); 2024, Vol. 32, p. 2205–2222.
33. Jagielski M, Nasr M, Choquette-Choo CA, Lee K, Carlini N, Tramèr F. Code contamination: legal and security risks of LLM code generation. arXiv:2401.05078. 2024.
34. Cybersecurity Ventures. Cybersecurity almanac: 2024 edition. Cybersecurity Ventures; 2024 Jan 2.
35. Abnormal Security. Combating WormGPT: What you need to know. 2025.
36. Lookout. What is WormGPT? 2025.
37. Ferrari executive nearly fooled by AI voice clone of CEO. Cybernews. 2023 Nov.
38. Xu M, Fan J, Huang X, Zhou Z, Kang J, Niyato D, Mao S, Han Z, Shen X, Lam K. Forewarned is forearmed: a survey on large language model-based agents in autonomous cyberattacks. arXiv:2505.12786. 2025.
39. Singer B, Lucas K, Adiga L, Jain M, Bauer L, Sekar V. On the feasibility of using LLMs to autonomously execute multi-host network attacks. arXiv:2501.16466. 2025.
40. NIST. AI risk management framework. NIST Special Publication; 2023.
41. World Economic Forum. Global AI security standards. WEF; 2023.
42. Google Security Blog. Safe framework implementation. Google; 2023.
43. Zhang Y. Operationalizing large language models for cybersecurity: Infrastructure, scalability, and performance benchmarks. J Cyber Inf Syst (JCIS). 2025;15(1):102–118.
44. Palo Alto Networks. Fairness and safety of LLMs. 2024 June.
45. Mohindroo S. Data privacy and compliance for large language models (LLMs). Medium. 2024 Sep.
46. Qualys. What is large language model (LLM) security. 2025 Apr.
47. Rajkomar A, Anthi E, Burnap P. A framework for trustworthy AI in cybersecurity operations. IEEE Commun Surv Tutorials. 2024;26:412–438.
48. Cynet. CrowdStrike vs Palo Alto: 5 key differences and pros & cons. Cynet blog. 2025.
49. European Union. Artificial intelligence act final text. Official Journal of the EU OJ L. 2025:55–56.
50. OWASP Foundation. Owasp top 10 for large language model applications. OWASP; 2025.
51. European Union. General Data Protection Regulation (GDPR); 2016. Publications Office of the European Union.
52. State of California Department of Justice. California consumer privacy act (CCPA). 2018.
53. Silva J. Explainable AI in cybersecurity: bridging transparency and trust. Proc IEEE Conf Cybersecurity Innovations; Lisbon, Portugal; 2025. p. 78–83.
54. Wang F, Zhao L, Chen M. CyberMentor: enhancing cybersecurity learning through explainable AI. Proc IEEE Int Conf Emerging Trends in Cyber Training Zhengzhou, China; 2025. p. 102–107.
55. Barrett R, Lee S, Harmon T. Ethical auditing in AI: the role of model cards and the cyber kill chain. IEEE Trans Technol Soc. 2023;10(2):123–132.
56. Gupta P, Sharma N, Desai K. A framework for ethical AI compliance under the EU AI Act. Proc IEEE Workshop on AI Governance Washington, DC, USA; 2023. p. 44–49.
57. Google Cloud. Cloud CISO perspectives: Our big sleep agent makes big leap. Google cloud blog. 2024.
58. Lisha M, Agarwal V, Kamthania S, Vutkur P, Chari M. Benchmarking LLMs for zero-day vulnerabilities. Proc IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT); Bengaluru, India; 2024. p. 95–102.
59. Shumailov I, Shumaylov Z, Zhao Y, Gal Y, Papernot N. The curse of recursion: training on generated data makes models forget. arXiv:2305.17493. 2023.
60. Kirchenbauer J, Geiping J, Wen Y, Katz J, Miers I, Goldstein T. A watermark for large language models. 2023. p. 11561–11575. In: Proc Int Conf on Machine Learning (ICML);.
61. Zhang R, Li H, Wen R, Jiang W, Zhang Y, Backes M, Shen Y, Zhang Y. Instruction backdoor attacks against customized LLMs. USENIX Secur. 2024; 33:3085–3102.
62. Protect AI. deberta-v3-base-prompt-injection-v2. Hugging Face; 2024.
63. OpenAI. Red teaming network report: findings from phase I. OpenAi. 2024 Jul.
64. Debenedetti E, Severi J, Carlini N, Choquette-Choo CA, Jagielski M, Nasr M, Wallace E, Tramèr F. Privacy side channels in machine learning systems. Proc USENIX Security; 2024:33:3121–3138.
65. Goldblum J, Fowl L, Goldblum M, Goldstein T. Dataset and model supply chain security for foundation models. Proc NeurIPS; 2024;37:12431–12445.
66. Dodds L, Torres-Arias S, Newman Z, Moore M, Kuppusamy TK. Sigstore: software signing for everybody. Proc USENIX Security Symposium. 2022;31:1827–1844.
67. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T. Stealing and misusing machine learning models. Proc USENIX Security Symposium; Austin, Texas, USA; 2016:25:601–618.
68. Goldblum M, Tsipras D, Xie C, Chen X, Schwarzschild A, Song D, Madry A, Li B, Goldstein T. Dataset security for machine learning. Proc NeurIPS. 2020:33:14781–14792.
69. Fan L, Ali K, Atallah MJ. Rethinking deep neural network ownership verification. Proc NDSS; San Diego, California, USA; 2019.
70. UK National Cyber Security Centre. Guidelines for secure use of large language models. NCSC; 2024.