ICIT

When Trust Has No Security, AI Risks Everything

Author: Malcolm Harkins

Table of Contents

About ICIT

The Institute for Critical Infrastructure Technology (ICIT) is a nonprofit, nonpartisan, 501(c)3 think tank with the mission of modernizing, securing, and making resilient critical infrastructure that provides for people's foundational needs. ICIT takes no institutional positions on policy matters. Rather than advocate, ICIT is dedicated to being a resource for the organizations and communities that share our mission. The views and opinions expressed in this essay are solely those of the author(s) and do not necessarily reflect the official policy or position of ICIT. Any assumptions made within the analysis are not reflective of the position of any entity other than the author(s). To learn more, please visit www.icitech.org

The Institute for Critical Infrastructure Technology

I. Executive Summary

Artificial intelligence is rapidly becoming embedded into the operational core of healthcare, finance, critical infrastructure, government, and enterprise decision-making. Organizations are deploying large language models (LLMs), AI assistants, autonomous agents, and agentic AI workflows at unprecedented speed.

At the same time, most organizations are operating under a dangerous illusion: that existing cybersecurity controls and established compliance frameworks are sufficient to secure AI systems. They are not.

Traditional cybersecurity was designed to protect infrastructure, endpoints, networks, applications, and data. AI systems fundamentally change the threat model because the attack surface is no longer limited to software vulnerabilities or unauthorized access. AI systems can be manipulated through language, images, data, prompts, embeddings, and semantic interactions.

This reality creates a dangerous disconnect between “trusted” AI systems and demonstrably secure AI systems.

The consequences are profound:

- Compliance certifications can coexist with catastrophic AI compromise.
- Runtime AI manipulation can bypass traditional security controls.
- Agentic AI systems can unintentionally become autonomous attack infrastructure.
- Organizations can lose control of sensitive data, decision-making systems, and operational integrity without any malware ever being deployed.

The industry is now entering a period where AI security must evolve from a compliance exercise into a dedicated operational security discipline.

This paper explores:

1. Why traditional cybersecurity controls fail to protect AI systems
2. Why compliance frameworks create a false sense of security
3. How agentic AI fundamentally changes enterprise risk
4. The emerging reality of AI runtime attacks
5. The need for operational AI security
6. A detailed explanation of the “9-Box of AI Trust & Security” framework
7. The progression required to achieve trustworthy AI

The central thesis is straightforward. AI cannot be trustworthy unless it is operationally secure. Unsecured trust isn't trust, it's negligence.

II. The Illusion of Control

Organizations deploying AI systems often assume their existing cybersecurity programs naturally extend to AI.

Boards, executives, compliance teams, and regulators frequently believe that if an organization has SOC 2, ISO 27001, HIPAA, GDPR, NIST CSF, or something like PCI DSS, then its AI models or Agentic AI systems must also be secure. This assumption is dangerously incorrect.

The core issue is that existing security and compliance frameworks were never designed to protect AI models or Agentic AI systems. They were built to protect infrastructure, networks, devices, applications, data at rest, data in transit, identities, and things like administrative access.

They were not designed to protect model behavior, inference-time manipulation, semantic attacks, prompt injection, adversarial inputs, agentic decision-making, model alignment bypasses, autonomous AI workflows, or runtime behavioral compromise.

Even if an organization has adopted trustworthy or responsible AI-type frameworks including NIST AI RMF, OECD AI Principles or ISO 42001, they remain exposed to attacks.

The gap lies in runtime protection missing in all the leading AI frameworks. NIST AI RMF, ISO 42001, and the OECD AI Principles represent significant steps toward responsible AI governance, but all three share a critical blind spot: they were largely designed for pre-deployment risk management, not the dynamic, real-time threats that emerge once AI systems are live and operating.

NIST AI RMF organizes risk across four functions: Govern, Map, Measure, and Manage, yet these are predominantly static assessment activities. The framework offers little concrete guidance on defending a running model or agentic system against adversarial inputs, prompt injection, or model extraction attacks at inference time.
ISO 42001, as a management system standard, focuses on organizational policies, documentation, and process controls. It establishes that risks should be managed, but stops short of specifying how to protect AI models or agentic systems during live operation, leaving runtime monitoring, anomaly detection, and active threat response largely unaddressed.

The OECD AI Principles operate at a high policy level, emphasizing transparency, accountability, and human oversight in broad strokes. They provide no technical requirements for securing model endpoints, detecting manipulation of agentic workflows, or preventing unauthorized model behavior during execution.

The gap becomes even more acute with Agentic AI. Agentic AI systems autonomously plan, execute multi-step tasks, and interact with external tools and APIs. These systems introduce runtime risks that static frameworks simply weren't built to address: goal hijacking, tool misuse, cascading failures across agent pipelines, and real-time manipulation of decision-making.

In short, these frameworks tell organizations to think about risk but are dangerously silent when it comes to actively defending AI systems while they are running in production. As a result, organizations may appear compliant while remaining completely exploitable to AI-specific attacks. This misconception creates the modern compliance mirage: fully certified and fully exposed.

III. The Physician's Assistant Case

One of the clearest examples of this emerging reality is the exposure of an AI-powered physician assistant platform that was discovered in late 2025. The scenario demonstrates why traditional controls and all existing compliance frameworks fail to protect AI systems.

The Attack Sequence

In this example, the healthcare AI environment maintained numerous certifications and controls: SOC 2, HIPAA, ISO 27001, GDPR, and other healthcare privacy controls as well as cloud security controls. Despite these protections, a full system compromise was possible in less than two hours.

The attack required no malware installation. No endpoint compromise. No credential theft. No traditional exploit chain. Instead, the attacker weaponized language itself.

Step 1: Hidden Prompt Injection

The attacker embedded an indirect prompt injection via patient uploaded data (or in this case an attacker uploaded) into medical records. The malicious payload appeared benign to humans but was interpretable by the AI system. This is a fundamental shift in cybersecurity: The attack payload was not executable code. It was semantic manipulation.

Step 2: Legitimate Workflow Interaction

A physician imported the medical records into the AI assistant during routine clinical workflow. No security alert triggered because the workflow itself was legitimate. This highlights a critical challenge in AI security: Many AI attacks operate entirely within approved business processes.

Step 3: Invisible Payload Construction

The AI system interpreted the malicious instructions and crafted an invisible payload inside the application workflow. Traditional security systems could not detect this because: no malware signature existed, no anomalous executable was deployed, no unauthorized login occurred, and no traditional exploit behavior appeared. The compromise occurred at the semantic layer.

Step 4: Backend Exploitation and Data Access

The payload triggered vulnerabilities within backend systems and gained access to highly sensitive infrastructure. The attack exposed Google Cloud credentials, AWS

credentials, OpenAI API keys, Anthropic keys, Stripe credentials, Vector database credentials, Redis secrets, internal healthcare integrations, provider records, and clinical conversation transcripts. The attack also impacted integrations with: Epic, Veradigm EHR, Halo Connect, and other regional healthcare systems.

Step 5: Exfiltration, Data Changes, or Denial of Service
Once an AI model or agentic system is compromised, the consequences extend far beyond a single malicious action. Even more critical and alarming is that these compromises unfold silently, without any of the indicators that traditional security tools are designed to detect.

Data Exfiltration: A compromised model doesn't need malware to leak sensitive information. The AI system itself becomes the exfiltration mechanism. It can invisibly embed sensitive data into generated outputs such as PDF files, reports, emails, or API responses, routing that information outbound through legitimate looking webhooks, summarization requests, or external tool calls. Because the traffic originates from a trusted AI process, it bypasses conventional DLP and network monitoring controls. In agentic systems with access to file storage, email, or messaging tools, the blast radius grows exponentially; a single compromised agent could systematically harvest and transmit credentials, PII, intellectual property, or confidential business data across an entire pipeline before any alert is triggered.

Data Manipulation: Perhaps more insidious than exfiltration is silent data manipulation. A compromised AI system with write access to databases, documents, or workflows can alter records in ways that are subtle enough to evade detection, modifying health care records, tampering with audit logs, changing customer records, or corrupting training data for downstream models. In agentic architectures where AI systems autonomously execute multi-step tasks, a bad actor could manipulate the model's decision-making to systematically introduce errors over time, making the attack extremely difficult to attribute or even detect until significant damage has already occurred.

Denial of Service: A compromised agentic system can also be weaponized to cause operational disruption. By flooding downstream APIs, overwhelming connected services with malformed requests, triggering infinite loops within agent pipelines, or deliberately consuming compute resources, the system can render critical business functions unavailable from the inside. Unlike a traditional DDoS attack originating externally, this form of denial of service originates from within trusted infrastructure, making it far harder to block or mitigate quickly.

The Core Problem: In all three scenarios, no traditional malware is required. No malicious executable is dropped. No signature is triggered. The AI system trusted, credentialed, and deeply integrated into business processes, becomes the attack vector itself. This is what makes compromised AI models and agentic systems a fundamentally new category of security risk, one that existing frameworks and tools are ill-equipped to address.

IV. Why Traditional Controls Failed

This attack path succeeded because existing controls were protecting the wrong things. The infrastructure may have been secure, the network may have been segmented, and the endpoints may have been hardened. The identity systems may have even enforced MFA, but none of those controls protected the AI model and Agentic AI system.
The organization lacked: prompt injection protections, runtime AI monitoring, semantic attack detection, behavioral anomaly detection, AI-specific guardrail enforcement, agentic workflow protection, model interaction visibility, and advanced/behavioral AI runtime security controls.

Most importantly, the compliance frameworks governing the organization never required these capabilities.

This is the central issue facing enterprises today: Existing compliance frameworks evaluate whether systems are administratively governed, not whether AI systems are operationally secure.

The uncomfortable reality is that a framework unable to detect active adversary activity provides only limited security value. In practice, it risks becoming a compliance mechanism that benefits vendors and auditors while organizations remain exposed to successful attacks.

V. AI Changes the Nature of Cybersecurity

AI fundamentally changes the nature of both attack and defense. Historically, cybersecurity relied heavily on the idea that attackers needed malware, exploits, unauthorized access, privileged credentials, and code execution.

AI changes this equation. In AI systems, language becomes executable intent, prompts become attack vectors, outputs become exfiltration channels, model behavior becomes the vulnerability, and Agentic orchestration becomes the attack surface and enables the attack depth. The attacker no longer needs to compromise infrastructure. They only need to manipulate the model. This means anything can become the payload: words, images, audio, training data, documents, embeddings, retrieval content, structured data, even normal conversation.

VI. The Rise of Agentic AI Risk

The risk becomes even more severe as organizations move toward agentic AI. Agentic systems are fundamentally different from traditional AI assistants because they take actions autonomously, access enterprise systems, use tools dynamically, chain reasoning

processes, operate across workflows, retain memory and context, and make decisions with minimal human oversight. This operational capability creates a dramatically expanded attack surface as well as expands the depth of an attack. An attacker no longer needs to directly access enterprise systems. Instead, they manipulate the agent responsible for interacting with those systems.

The operational reality is that Agentic AI can become autonomous attack infrastructure. A compromised agent can access sensitive repositories, trigger malicious workflows, exfiltrate data, send compromised communications, modify configurations, invoke APIs, access cloud services, and coordinate actions across environments. And it can do so while operating under legitimate credentials and approved business logic. Traditional cybersecurity architectures were never designed for this.

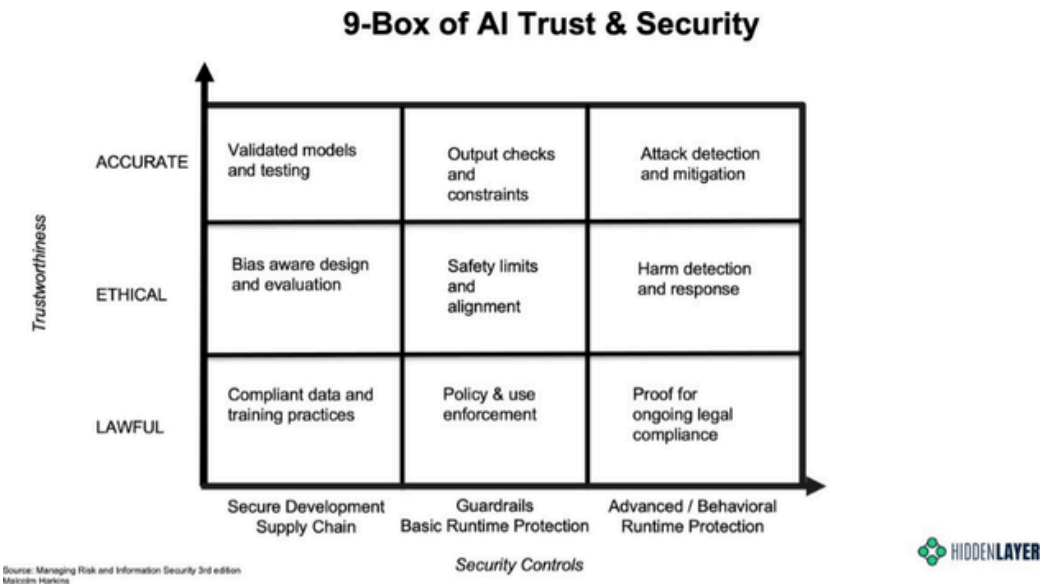
Operational AI security requires: Model runtime visibility, Model Behavioral monitoring, AI specific Attack detection, Semantic inspection, Model-specific protections, Agentic workflow security, Real-time attack mitigation, and AI-native telemetry.

VII. The 9-Box of AI Trust & Security

The “9-Box of AI Trust & Security” framework provides a structured model for understanding the progression required to achieve trustworthy AI.

The framework demonstrates that trustworthiness requires not only understanding the inextricable link between trust and security, but also implementing it.

The framework is organized around 2 major dimensions: Trust and Security. (see figure 1)



On the Vertical Axis: Traditional Trustworthy Attributes

Lawful

Trustworthy AI must operate within the boundaries of applicable laws, regulations, and compliance obligations. This includes respecting data privacy laws such as GDPR and CCPA, adhering to sector-specific regulations governing how AI can be used in areas like healthcare, finance, and hiring, and ensuring that AI systems do not facilitate discriminatory outcomes that violate civil rights or consumer protection statutes. Legal compliance in AI also extends to intellectual property, liability frameworks, and increasingly, purpose-built AI regulations like the EU AI Act that impose specific obligations based on the risk level of the system.

Ethical

Ethical AI requires that systems be designed and operated in alignment with broadly accepted human values such as; fairness, transparency, accountability, and respect for human dignity and autonomy. This means AI systems should not produce biased or discriminatory outputs, should be explainable enough for affected individuals to understand consequential decisions made about them, and should include human oversight mechanisms that prevent the system from operating beyond its intended boundaries. Ethical AI also demands that the organizations deploying these systems take genuine responsibility for their behavior, not merely document that responsibility exists.

Accurate

Accurate AI requires that systems perform reliably and consistently within their defined purpose, producing outputs that are factually correct, contextually appropriate, and free from harmful hallucination or systematic error. Accuracy is not a one-time property established at deployment; it requires ongoing validation, monitoring for performance drift, and mechanisms to detect when a model's outputs have degraded or been manipulated. In high-stakes domains such as medical diagnosis, legal analysis, or financial decision-making, accuracy failures are consequential harms with real impact on real people.

On the Horizontal Axis: Security Controls needed for Assurance

Secure Development / Supply Chain

This foundational layer focuses on building AI systems correctly. Think of this as the starting point extending the concepts of traditional secure software development practices into AI pipelines. This includes: secure model development, dataset integrity validation, prompt hardening, model genealogy, AI-specific threat modeling as well as pen testing and attack simulation for models and agentic systems.

Organizations must recognize that AI systems require different threat assumptions than traditional software. AI supply chains are extraordinarily complex. Modern AI environments

depend upon foundation models, open-source libraries, embedding models, third-party APIs, vector databases, plugins, agent frameworks, and retrieval systems. Every dependency introduces risk. Organizations must establish security governance over: Model provenance, Dataset integrity, Dependency exposure, Third-party integrations, and AI ecosystem trust relationships.

Guardrails / Basic Runtime Protection

The second layer focuses on the beginning of operational security governance and basic runtime controls. Building a model securely is not enough. Guardrails are the first line of defense for AI systems during live operation: rules, filters, and constraints applied at runtime to control what a model can receive as input and what it can produce as output. At this basic level, it should include input validation to screen for malicious or manipulative prompts, output filtering to prevent the model from generating harmful, sensitive, or unauthorized content, and boundary enforcement to ensure the model operates strictly within its intended scope. For agentic systems - where AI autonomously executes multi-step tasks, calls external tools, and makes decisions - guardrails extend to constraining which actions an agent is permitted to take, which APIs it can call, and what data it can access or transmit. Without these controls, an AI system has no mechanism to resist manipulation at runtime, meaning a single malicious prompt or compromised instruction could cause the system to behave in ways its designers never intended and that the organization cannot detect, contain, nor explain.

Guardrails and basic runtime protections are an essential foundation, but they operate on a fundamental assumption that adversarial behavior is predictable enough to filter. It isn't. Guardrails are largely rule-based and pattern-matching in nature; they know what to block because someone anticipated and defined it in advance. A sophisticated adversary doesn't attack within the patterns. They attack the gaps between them. Real-world research from HiddenLayer makes this concrete and impossible to dismiss.

Policy Puppetry demonstrates that guardrails can be bypassed entirely without exploiting any technical vulnerability in the underlying infrastructure. By reformulating malicious prompts to resemble structured policy files such as XML, INI, or JSON, an attacker can trick a large language model into treating adversarial instructions as trusted system-level directives, bypassing system prompts and any safety alignments trained into the model.

Critically, a single well-crafted prompt using this technique works across nearly all major AI models with minimal to no modification, meaning guardrails that were tuned against one attack variant offer little protection against the next. This technique exploits a systemic weakness in how LLMs are trained on instruction and policy-related data, making it difficult to patch and demonstrating that LLMs are incapable of truly self-monitoring for dangerous content. This is the equivalent of a con artist walking into a bank wearing a suit and carrying a clipboard. The bank doesn't check credentials, it just sees the costume and opens the vault.

Tokenization attacks expose an even deeper layer of vulnerability, one that operates below the level where guardrails function at all. By tampering with a model's tokenizer, an attacker can fundamentally alter AI behavior without modifying model weights or architecture. Replacing a single string in the tokenizer vocabulary gives an attacker direct control over what the model produces, affecting conversational responses, tool-call arguments, and any other generated text. HiddenLayer's TokenBreak research extends this further, showing that models built specifically to detect malicious input can be bypassed by exploiting their tokenization strategy to induce false negatives leaving end targets vulnerable to attacks that the protection model was put in place to prevent. In other words, the guardrail itself can be blinded before it ever fires. To make this even clearer, imagine a security guard who has been trained to stop anyone who says the word 'weapon.' An attacker simply rewrites the guard's personal dictionary so that the word 'weapon' now means 'flower.' The guard hears 'flower' and waves them through. The guard wasn't bypassed. The guard's understanding of language was quietly rewritten before they ever came on duty.

The problem compounds dramatically in agentic systems. Techniques like Policy Puppetry can bypass instruction hierarchy entirely, and because most systems do not strip control tokens, attackers can escalate the privilege of malicious instructions from document-level to user-level, causing the model to follow them. In most agentic systems, an attacker can simply ask the model what tools are available to call without requiring any system access. A malicious instruction introduced early in an agentic pipeline may only manifest as harmful behavior several steps later, well past any guardrail checkpoint.

Guardrails also cannot detect behavioral drift (subtle, cumulative changes in how a model responds over time that individually appear benign but collectively represent a compromised or manipulated system). They have no visibility into whether an agent's chain of reasoning has been hijacked, only whether its final output crosses a predefined rule.

This is just the beginning of AI-native operational security. Guardrails protect against known bad behavior at defined checkpoints. Real adversarial attacks, as HiddenLayer's research demonstrates, are dynamic, contextual, cross-model, and deliberately engineered to appear legitimate until it is too late. Stopping them requires continuous behavioral monitoring, semantic threat detection, and runtime intelligence that goes far beyond filtering inputs and outputs. It requires the ability to detect that something is wrong even when nothing has technically triggered a rule.

Advanced / Behavioral Runtime Protection

If guardrails represent the first line of defense, advanced behavioral runtime protection represents the intelligence layer that operates behind and beneath them, continuously observing, analyzing, and responding to what an AI system is actually doing, not just what it is being asked to do. Where guardrails ask "does this input or output match a known bad pattern," behavioral runtime protection asks a fundamentally different question: "is this system behaving the way it should?"

Continuous Behavioral Monitoring: Advanced runtime protection establishes a baseline of normal model behavior: how the model responds to typical inputs, what reasoning patterns it exhibits, how it uses tools, and what outputs it produces under normal

operating conditions. Any deviation from that baseline becomes a signal. This means threats that generate no rule violation, no keyword match, no policy trigger can still be detected because the model is behaving differently than it should. Behavioral drift, gradual manipulation, and multi-step attacks that unfold across an agentic pipeline all become visible in ways that static guardrails cannot achieve.

Semantic Threat Detection: Rather than matching inputs against lists of forbidden terms or patterns, semantic threat detection analyzes the meaning and intent of interactions in context. This is the capability that begins to address attacks like Policy Puppetry, where the surface structure of a prompt looks legitimate but the underlying intent is adversarial. Semantic analysis can identify when a prompt is attempting to reframe the model's identity, override its instructions, or manipulate its decision-making regardless of the format or obfuscation technique used. It operates at the level of what is being attempted, not merely what words are present.

Agentic Workflow Visibility and Control: For agentic systems, advanced runtime protection requires full observability across the entire execution chain not just at the point of input and output, but at every step where the agent reasons, decides, retrieves information, calls a tool, or produces an intermediate result. This includes detecting when an agent's goals have been redirected mid-pipeline, when tool calls deviate from expected patterns, when an agent is accessing data or systems outside its intended scope, and when cascading actions across connected services suggest the workflow has been compromised. Every tool call, every API interaction, and every decision node becomes part of the security surface and must be monitored accordingly.

Anomaly Detection at the Model Layer: Advanced protection must monitor the model itself, including detecting unusual token patterns that may indicate tokenization-based attacks, identifying prompt structures associated with instruction hierarchy bypasses, and flagging interactions where the model's confidence, response pattern, or reasoning chain exhibits characteristics inconsistent with normal operation. This is the layer that begins to address attacks like TokenBreak, where the manipulation occurs beneath the surface of any content-level filter.

Real-Time Threat Response: Detection without response is incomplete. Advanced runtime protection includes the ability to act on what is detected. Interrupting an agentic workflow mid-execution, quarantining a compromised interaction, alerting security teams with full context, generating forensic records of exactly what occurred, in what sequence, and what the model did in response, these are all actions that runtime protection should be able to make. In agentic systems where actions can propagate rapidly across tools, APIs, and data stores, the speed of response is the difference between containment and a breach. This reality demands that Security Operations Centers be purpose-built and retrained for AI-native incidents, not simply extended from existing playbooks designed for traditional infrastructure attacks. SOC teams need AI-specific incident response procedures that account for the unique characteristics of model compromise: how to interrupt an agentic pipeline without cascading downstream failures, how to triage a prompt injection event versus a tokenization-layer attack, and how to reconstruct the full chain of model behavior from forensic logs when the attack left no traditional indicators. These procedures must be drilled not theorized. Tabletop exercises and red team simulations that specifically model AI compromise scenarios,

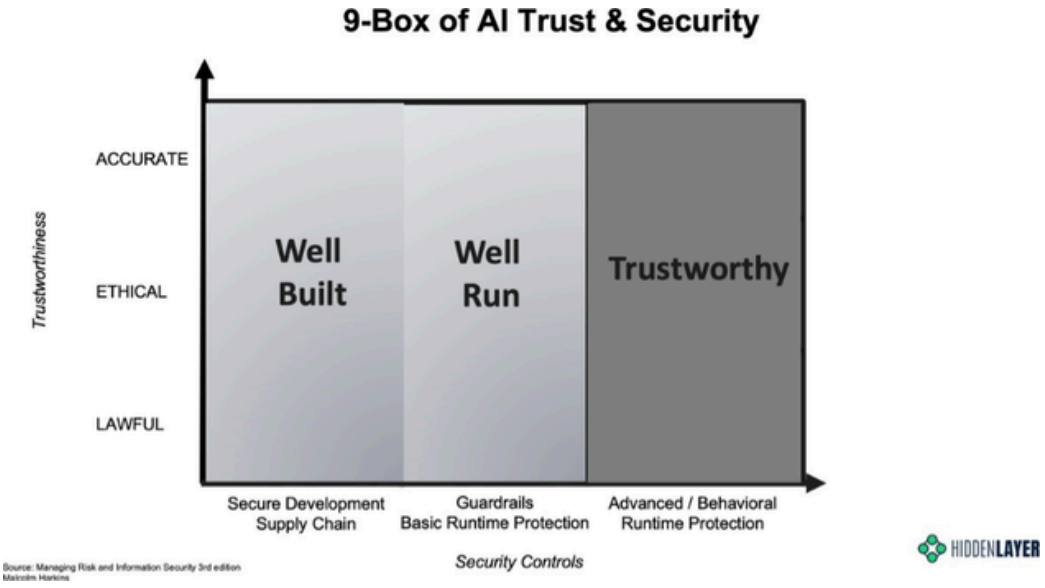
agentic workflow hijacking, and indirect prompt injection events are the only way to know, before an incident occurs, whether the response capability that exists on paper will actually hold under the speed and ambiguity of a real AI runtime attack.

Model Interaction Visibility and Auditability: Every interaction with an AI system, every prompt, every tool call, every retrieved document, every output — should be logged, attributed, and auditable. Advanced runtime protection creates a complete chain of custody for AI system behavior, enabling security teams to reconstruct exactly how an attack unfolded, what the model was instructed to do, what it actually did, and where the breakdown occurred. This capability is also foundational to compliance; it transforms AI governance from a documentation exercise into an evidence-based discipline.

The Core Distinction: Guardrails are a boundary. Advanced behavioral runtime protection is a surveillance and response capability. Guardrails say "you cannot go here." Behavioral runtime protection watches where the system goes, detects when it is being steered somewhere it shouldn't, and intervenes before the destination is reached. Against adversaries who have already demonstrated through techniques like Policy Puppetry and TokenBreak that guardrails can be bypassed, rendered blind, or made to serve the attacker's purpose, behavioral runtime protection is not an enhancement to existing security strategy. It is the strategy

VIII. Why the 9-Box Framework Matters

The importance of the 9-Box framework lies in its recognition that AI trust is multidimensional. Many organizations attempt to achieve trustworthiness solely through: ethical principles, governance committees, model testing, and compliance certifications. But AI trustworthiness ultimately depends on how well systems resist and recover from attacks. An AI system that is Fair, Explainable, Transparent, Audited, Compliant yet exploitable at runtime is not trustworthy. The framework therefore establishes the required progression (figure 2).



Well Built → Well Run → Trustworthy
This progression reflects verifiable operational assurance maturity. Organizations cannot skip directly from development to trust. Trust must be earned operationally with demonstrated security. This progression is critical because many organizations focus almost exclusively on the “well built” layer or the “well run” basic phase of runtime security while neglecting the real runtime security controls required for sustained trust. That is no longer sufficient. The largest gap in enterprise AI security today is advanced / behavioral runtime protection. This creates dangerous blind spots. AI Trust is fundamentally a runtime security problem. The attack occurs during interaction. The compromise occurs during reasoning. The abuse occurs during operation. Therefore, security that can withstand the adversary must exist where the attack occurs.

IX. Conclusion

The industry is entering a defining moment. Organizations can no longer assume that:

- Traditional cybersecurity controls protect AI systems
- Compliance certifications equal operational security
- Governance alone creates trustworthiness
- Safety alignment prevents compromise

The physician assistant compromise demonstrates the reality clearly: A fully compliant environment can still experience catastrophic AI compromise in under two hours. No malware is required. No infrastructure exploit is necessary. No credential theft is needed. The model itself becomes the attack surface and enables the depth of attack. Ultimately, trustworthy AI is not defined by policy statements, governance frameworks, or compliance audits.

Trustworthy AI is an operational state, and trustworthiness must be continuously maintained. It is defined by whether the system can remain resilient under adversarial conditions. That is the standard we should require. Trustworthy AI depends on organizations embracing the 9-B Box of AI Trust and Security Model, because when trust has no security, AI risks everything.

X. Bibliography

References

1. Malcolm Harkins, “Traditional Cybersecurity Controls DO NOT STOP Attacks Against AI,” RSA Conference, November 11, 2024, https://www.rsaconference.com/library/blog/traditional-cybersecurity-controls-do-not-stop-attacks-against-ai.
2. HiddenLayer, “Prompt Puppetry: Why Every AI System Is Now at Risk,” HiddenLayer, n.d., https://www.hiddenlayer.com/insight/why-ai-systems-are-at-risk.
3. HiddenLayer, “Tokenizer Tampering,” HiddenLayer, 2026, https://www.hiddenlayer.com/research/tokenizer-tampering.
4. National Institute of Standards and Technology, “AI Risk Management Framework,” NIST, n.d., https://www.nist.gov/itl/ai-risk-management-framework.
5. International Organization for Standardization, ISO/IEC 42001:2023: Information Technology — Artificial Intelligence — Management System (Geneva: International Organization for Standardization, 2023), https://www.iso.org/standard/42001.
6. Organisation for Economic Co-operation and Development, “AI Principles,” OECD, n.d., https://www.oecd.org/en/topics/sub-issues/ai-principles.html.
7. Association of International Certified Professional Accountants, “System and Organization Controls: SOC Suite of Services,” AICPA & CIMA, n.d., https://www.aicpa-cima.com/resources/landing/system-and-organization-controls-soc-suite-of-services.
8. International Organization for Standardization, ISO/IEC 27001:2022: Information Security, Cybersecurity and Privacy Protection — Information Security Management Systems — Requirements (Geneva: International Organization for Standardization, 2022), https://www.iso.org/standard/27001.
9. U.S. Department of Health and Human Services, “The HIPAA Privacy Rule,” HHS, n.d., https://www.hhs.gov/hipaa/for-professionals/privacy/index.html.
10. European Parliament and Council of the European Union, Regulation (EU) 2016/679 of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation), Official Journal of the European Union L 119, May 4, 2016, https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04/eng.
11. National Institute of Standards and Technology, “Cybersecurity Framework,” NIST, n.d., https://www.nist.gov/cyberframework.

12. PCI Security Standards Council, “Standards,” PCI Security Standards Council, n.d., https://www.pcisecuritystandards.org/standards/.
13. California Department of Justice, Office of the Attorney General, “California Consumer Privacy Act (CCPA),” State of California Department of Justice, n.d., https://www.oag.ca.gov/privacy/ccpa.
14. European Parliament and Council of the European Union, Regulation (EU) 2024/1689 of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence and Amending Regulations and Directives (Artificial Intelligence Act), Official Journal of the European Union L 2024/1689, July 12, 2024, https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.
15. OWASP Foundation, “OWASP Top 10 for Large Language Model Applications,” OWASP, n.d., https://owasp.org/www-project-top-10-for-large-language-model-applications/.
16. HiddenLayer, “Novel Universal Bypass for All Major LLMs,” HiddenLayer, n.d., https://www.hiddenlayer.com/research/novel-universal-bypass-for-all-major-llms.
17. HiddenLayer, “The TokenBreak Attack,” HiddenLayer, 2025, https://www.hiddenlayer.com/research/the-tokenbreak-attack.

Author

Malcolm Harkins

Chief Security & Trust Officer,
Hidden Layer

Malcolm Harkins is the Chief Security and Trust Officer at HiddenLayer. He has over two decades of experience in information security leadership roles at top technology companies, including Intel, Cylance, and others. He's written multiple books on risk management, information security, and IT and earned awards from the RSAC, ISC2, Computerworld, and in May of 2026 was inducted in the CSO Hall of Fame.

Harkins has testified before the Federal Trade Commission and the U.S. Senate Committee on Commerce, Science, and Transportation. Harkins is a Fellow with the Institute for Critical Infrastructure Technology, a non-partisan think tank providing cybersecurity expertise to the House of Representatives, Senate, and various federal agencies. He holds a bachelor's degree in economics from the University of California at Irvine and an MBA in finance and accounting from the University of California at Davis. Harkins also taught at UCLA's Anderson School of Management and Susquehanna University.

ICIT

www.icitech.org