

May 2025



ICIT

Securing AI Addressing the OWASP Top 10 for Large Language Model Applications

Dr. Darren Death

Fellow, Institute for Critical Infrastructure Technology

www.icitech.org

Table of Contents

Chapter 1 - AI Security Is Not Optional 4

AI Is Just Software—But It Is Not Just Software.	4
The Consequences of Insecure AI Are Already Happening . . .	4
Beyond Zero Trust: Addressing AI's Unique Security Challenges. .5	
The OWASP Top 10 for LLM Applications: A Practical Framework for AI Security	5
AI Security Is a Leadership Responsibility	6
What Comes Next	6

Chapter 2 - Understanding and Addressing Prompt Injection in AI Systems 7

What Is Prompt Injection?.	7
Why Prompt Injection Matters to Organizations.	7
Examples of Prompt Injection Risks	7
Strategies to Mitigate Prompt Injection	8
Enhancing the Security and Integrity of AI Systems.	8

Chapter 3 - Understanding and Addressing Sensitive Information Disclosure in AI Systems . 9

What Is Sensitive Information Disclosure?.	9
Why Sensitive Information Disclosure Matters to Organizations . .9	
Examples of Sensitive Information Disclosure Risks	9
Strategies to Mitigate Sensitive Information Disclosure	10
Building Secure AI Systems	10

Chapter 4 - Understanding and Addressing Supply Chain Risks in AI Systems. 11

What Are Supply Chain Risks in AI Systems?.	11
Remember, AI is just software.	11
Examples of Supply Chain Risks in AI.	11
Strategies to Mitigate Supply Chain Risks	12
Building Resilience in AI Supply Chains	13

Chapter 5 - Understanding and Addressing Data and Model Poisoning in AI Systems . . . 14

What Is Data and Model Poisoning?.	14
Why Data and Model Poisoning Matters	14
Examples of Data and Model Poisoning Risks	14
Strategies to Mitigate Data and Model Poisoning.	15
Building Resiliency Against Poisoning Risks.	15

Chapter 6 - Understanding and Addressing Improper Output Handling in AI Systems . . . 16

What Is Improper Output Handling?.	16
Why Improper Output Handling Matters?.	16
Examples of Improper Output Handling Risks	16

Strategies to Mitigate Improper Output Handling.	17
Improving Output Reliability in AI Systems	17

Chapter 7 - Understanding and Addressing Excessive Agency in AI Systems 18

What Is Excessive Agency?.	18
Why Excessive Agency Matters to Organizations	18
Examples of Excessive Agency Risks	18
Strategies to Mitigate Excessive Agency.	19
Building Confidence in AI through Responsible Oversight 19	

Chapter 8 - Understanding and Addressing System Prompt Leakage in AI Systems. 20

What Is System Prompt Leakage?.	20
Why System Prompt Leakage Matters.	20
Examples of System Prompt Leakage Risks	20
Strategies to Mitigate System Prompt Leakage.	21
Building Resilience Against System Prompt Leakage	21

Chapter 9 - Understanding and Addressing Vector and Embedding Weaknesses in AI Systems . 22

What Are Vector and Embedding Weaknesses?.	22
Why Vector and Embedding Weaknesses Matter to Organizations.	22
Examples of Vector and Embedding Weakness Risks	23
Strategies to Mitigate Vector and Embedding Weaknesses . . 23	
Building Resilience Against Vector and Embedding Weaknesses	24

Chapter 10 - Understanding and Addressing Inaccurate or Misleading Outputs in AI Systems. 25

What Are Inaccurate or Misleading Outputs?.	25
Why Inaccurate or Misleading Outputs Matter to Organizations.	26
Examples of Inaccurate or Misleading Output Risks	26
Strategies to Mitigate Inaccurate or Misleading Outputs	26
Building Confidence in AI Systems.	27

Chapter 11 - Understanding and Addressing Unbounded Consumption in AI Systems. 28

What is Unbounded Consumption	28
Why Unbounded Consumption Matters to Organizations	28
Examples of Unbounded Consumption in AI Systems.	29
Strategies for Controlling AI Resource Consumption.	29
Ensuring AI Sustainability.	30

Table of Contents

Chapter 12 - Securing AI Is a Continuous Commitment 31

AI Security: Not a One-and-Done Effort	31
Ensuring a Secure AI Supply Chain	31
Integrating Security Into Design and Build Processes	32
Establishing Secure AI Development Pipelines	32
Addressing the Risks of Generative AI	32
Future Areas for Research and Advancement	33
Enhancing AI Explainability and Transparency	33
Improving Adversarial AI Detection and Defense	33
Securing AI Model Updates	33
Addressing the Risks of Synthetic Data	33
AI-Powered Threat Detection for Cybersecurity	33

Chapter 13 – Supplemental – Review of NIST AI 100-2e2025 34

Understanding Predictive AI and Generative AI	34
Broader Adversarial Threats Across AI System Types	35
Evasion Attacks	35
Poisoning Attacks	36
Privacy Attacks	36
Taxonomy by Attacker Goals, Capabilities, and Knowledge	37
Attacker Goals	37
Attacker Capabilities	38

Attacker Knowledge	38
Operational Implications	39
Supply Chain Vulnerabilities	39
Pre-Trained Models	39
Fine-Tuning Data	40
Model Artifacts and Components	40
Risk Implications	40
Mitigation Approaches	40
Direct Versus Indirect Prompt Injection	41
Defensive Considerations	41
Securing LLM Agents and Tool Use	42
Challenges in Evaluating and Sustaining Defenses	42
Trade-Offs Between Robustness, Accuracy, and Privacy	43
Security Design Implications	43
Multimodal System Considerations	43
Security Design Implications	44
Operational Risk	44
Expanding Privacy Attack Awareness	44
Defense Considerations	45
Strategic Implications	45
Sustaining Secure and Trustworthy AI Operations	45
Final Thoughts	46
Glossary of Terms	47



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



Thank you to our Strategic Partner **CyberRisk Alliance** | cyberriskalliance.com

Chapter 1 - AI Security Is Not Optional

AI Is Just Software—But It Is Not Just Software

Artificial Intelligence (AI) is frequently portrayed as a disruptive force with the potential to revolutionize industries, optimize workflows, and enhance decision-making in ways that were often seen as unattainable. While this perspective highlights AI's impact, it overlooks a fundamental reality. AI is still software at its core—it runs on code, processes data, and operates within an infrastructure like other enterprise systems. Yet, despite these similarities, AI introduces critical differences that cannot be overlooked.

Unlike traditional software, which follows predefined logic and executes tasks based on explicit programming, AI systems—particularly large language models (LLMs) are designed to operate probabilistically. These models do not adhere to fixed decision trees or structured workflows. Instead, they generate responses dynamically, relying on statistical relationships between words, phrases, and concepts. This distinction is not merely a technical detail; it has significant implications for security, governance, and risk management.

Organizations that deploy AI cannot rely solely on the security frameworks and best practices traditionally applied to conventional IT systems. For example, a firewall will not prevent an AI model from inadvertently leaking sensitive information if prompted in a specific way. Similarly, a traditional endpoint detection system will not stop an attacker from poisoning a model's training data or manipulating embeddings to alter the responses generated by the AI. Therefore, AI security requires an expanded approach that considers the unique ways AI models process input, generate output, and interact with enterprise systems.

The Consequences of Insecure AI Are Already Happening

The risks associated with AI security failures are significant and impact various industries. In recent years, we have seen instances where chatbots powered by large language models (LLMs) have been manipulated to produce harmful or unexpected responses. For instance

Microsoft's AI chatbot, Tay, was taken offline within 24 hours of its launch after users exploited its learning model to generate offensive content. (<https://www.theguardian.com/technology/2016/mar/26/microsoft-deeply-sorry-for-offensive-tweets-by-ai-chatbot>). Google received over 250 complaints globally between April 2023 and February 2024 regarding its AI software being used to create deepfake terrorism content, with further reports indicating misuse of the AI program Gemini to generate child abuse material (<https://www.reuters.com/technology/cybersecurity/google-reports-scale-complaints-about-ai-deepfake-terrorism-content-australian-2025-03-05/>). AI-generated content has also led to legal repercussions, as seen when a lawyer unknowingly submitted six fake case precedents generated by ChatGPT in a court filing, resulting in disciplinary action (<https://www.apnews.com/article/artificial-intelligence-chatgpt-fake-case-lawyers-d6ae9fa79d0542db9e1455397aef381c>).

Adversarial attacks are increasingly revealing vulnerabilities in AI decision-making within the healthcare sector. Researchers have shown that manipulated inputs can deceive AI systems, causing them to misinterpret medical data and lead to incorrect diagnoses or treatment recommendations. For example, studies indicate that adversarial attacks can alter the outcomes of AI in medical imaging, which may jeopardize patient safety (<https://www.bankinfosecurity.com/study-attacks-manipulate-medical-imaging-ai-outcomes-a-18131>). These real-world examples highlight the urgent need for stronger AI security measures and governance frameworks.

Beyond Zero Trust: Addressing AI's Unique Security Challenges

Historically, organizations have approached cybersecurity by combining network defenses, endpoint protection, and access controls. Zero Trust architecture has emerged as a cornerstone of modern security strategies; it enforces strict access controls and assumes that no user or system should be trusted by default. However, a Zero Trust approach alone cannot fully secure AI systems. AI introduces risks that extend beyond traditional infrastructure security concerns. For instance, a large language model (LLM) doesn't need to be "hacked" in the conventional sense to pose a risk; it can be exploited simply through specific types of interaction. Zero Trust focuses on enforcing rigorous controls over identity, access, and data protection. However, while it ensures data security by limiting access, verifying trust continuously, and encrypting sensitive information, it does not adequately address AI-specific vulnerabilities such as adversarial inputs, data poisoning, or model manipulation.

Cybersecurity as a discipline does address these risks—just not through traditional network-based protection and Zero Trust alone. AI security is part of the broader principles of cybersecurity, which include data integrity, software assurance, and resilience against adversarial attacks. Techniques such as adversarial training, input validation, model monitoring, and cryptographic signing of AI models complement existing cybersecurity strategies to secure software supply chains, protect against insider threats, and ensure data authenticity. While Zero Trust enforces strict access controls and data protection measures, it is primarily designed to prevent unauthorized access and enforce least privilege—not to defend against AI-specific threats like adversarial manipulation or data poisoning. Securing AI requires additional protective measures beyond traditional access controls, ensuring data integrity, monitoring model behavior, and building resilience against manipulation.

A well-crafted prompt can mislead an AI model into revealing confidential information, sharing internal instructions, or producing inaccurate outputs. Carefully manipulated datasets can subtly alter a model's behavior, introducing misleading information or vulnerabilities. Organizations that deploy AI must adopt an approach to cybersecurity tailored explicitly for AI. They need to assess how AI-generated outputs can be manipulated, how external data sources might influence models, and how adversaries could exploit weaknesses that do not exist in traditional software. AI security is not simply about adding another layer of controls to existing frameworks; it requires addressing security at every level, from model development and deployment to governance and continuous monitoring.

The OWASP Top 10 for LLM Applications: A Practical Framework for AI Security

The OWASP Top 10 for LLM Applications highlights the most critical security risks facing AI deployments today. This article is structured around these risks, with each chapter focusing on a specific threat:

- 1 **Prompt Injection** – How attackers manipulate AI responses by injecting malicious inputs.
- 2 **Insecure Output Handling** – The risks of uncontrolled AI-generated outputs and their impact.
- 3 **Training Data Poisoning** – How adversaries can corrupt AI models by introducing manipulated training data.
- 4 **Model Denial of Service (DoS)** – The exploitation of AI resource consumption to degrade performance or disrupt availability.
- 5 **Supply Chain Vulnerabilities** – Security risks in third-party AI models, dependencies, and integrations.
- 6 **Excessive Agency** – When AI systems operate with too much autonomy, leading to unintended or harmful actions.
- 7 **Insecure Plugin Design** – The dangers of poorly secured extensions and integrations in AI systems.
- 8 **Overreliance on LLMs** – The risks of treating AI-generated content as inherently trustworthy.
- 9 **Data Leakage** – How AI models can unintentionally expose sensitive or proprietary information.
- 10 **Unbounded Resource Consumption** – The impact of AI systems consuming excessive compute, memory, or API resources.

Each chapter will detail how vulnerabilities function, how they can be exploited, and how organizations can mitigate them. The aim is to go beyond theoretical discussions and provide practical, actionable measures that organizations can implement to secure their AI systems.

AI Security Is a Leadership Responsibility

Cybersecurity has always been a critical leadership issue, and AI security is no exception. Implementing AI goes beyond a mere technical decision; it is a strategic initiative that affects risk management, compliance, governance, and public trust. If leadership does not prioritize AI security, it will likely be overlooked at the operational level. Organizations that fail to incorporate security into their AI strategies will end up reacting to breaches instead of preventing them. Cybersecurity must be a top priority for senior leaders (agency heads and CEOs) in the public and private sectors, not just IT and security teams. AI security risks extend beyond technology; they threaten organizational integrity, regulatory compliance, and public trust. Leadership at the highest levels must recognize that failing to address AI security proactively can lead to devastating consequences.

Beyond technology concerns, AI failures can have life-altering consequences. An AI system making flawed medical recommendations in healthcare could endanger patient safety. In finance, manipulated AI-driven decisions could lead to flawed lending practices or market instability. A single AI-generated cybersecurity event could erode public confidence in an organization's decision-making ability. For these reasons, AI security cannot be treated as an afterthought. From the outset, it must be embedded into enterprise risk management and governance strategies. Agency heads, CEOs, and senior leaders across the organization must set the tone and demand AI security oversight to protect their organizations and those they serve.

What Comes Next

The following chapters will detail the OWASP Top 10 AI security risks, providing concrete mitigation strategies and implementation frameworks. The goal is to provide security professionals, policymakers, and AI practitioners with a practical guide to securing AI deployments.

The first significant risk to address is prompt injection, an attack technique that allows adversaries to manipulate AI-generated responses, bypass security controls, and exploit weaknesses in LLM behavior. Understanding and mitigating prompt injections is critical to ensuring the security of AI systems.

Chapter 2 - Understanding and Addressing Prompt Injection in AI Systems

Artificial intelligence (AI) is transforming how organizations make decisions. However, it also introduces risks that must be addressed to protect operations and data. One such risk is prompt injection, a vulnerability that can manipulate AI systems to produce harmful or unintended results.

? What Is Prompt Injection?

Prompt injection occurs when a user submits input that alters the behavior of an AI system in unexpected ways. These inputs can bypass safeguards, causing the AI to perform tasks it was not designed to do. For example, a chatbot might disclose private information or execute unauthorized actions if prompted a certain way.

This issue impacts any organization using AI systems, whether for customer service, internal operations, or data analysis. If left unchecked, prompt injection can lead to incorrect outputs or decisions based on manipulated information.

☆ Why Prompt Injection Matters to Organizations

Prompt injection is a critical vulnerability because it exploits AI systems' inherent design to interpret and respond to user inputs. The risk increases with highly autonomous AI systems integrated into essential processes. Systems with broader permissions or access to sensitive data are more vulnerable to harmful outcomes if compromised.

Organizations may use AI systems integrated with external tools or data sources. These integrations can amplify the risk, allowing malicious actors to influence the AI indirectly. For example, attackers might manipulate external data feeds or embedded content to affect the AI's behavior.

Examples of Prompt Injection Risks

Understanding the risks associated with prompt injection is necessary to safeguard an AI deployment's functionality and security. Recognizing these vulnerabilities is the first step in mitigating risks and ensuring responsible AI deployments.

- 1 **Customer Service Errors:** An AI-powered customer support system could be manipulated to provide incorrect or sensitive information, mislead users, or violate privacy policies.
- 2 **Unauthorized Actions:** Attackers could prompt an AI to execute commands or interact with systems it shouldn't access, leading to unintended actions like deleting records or changing settings.
- 3 **Misleading Outputs:** An AI could be tricked into generating misleading reports or summaries, which decision-makers might rely on, potentially causing severe consequences.
- 4 **Data Leakage:** Sensitive organizational data might be unintentionally exposed if the AI is prompted to share restricted information.
- 5 **Automation Misuse:** Systems automating processes like approvals or system changes could be compromised, leading to unauthorized actions without human oversight.

Strategies to Mitigate Prompt Injection

Prompt injection attacks pose a significant threat to the integrity and reliability of AI systems. An organization can implement strategies like those below to develop an effective defense against prompt injection attacks.

- **Restrict AI Capabilities:** Limit the AI's capabilities (following the principle of least privilege) to ensure it operates within defined parameters. Avoid granting access to sensitive data or critical systems unless necessary.

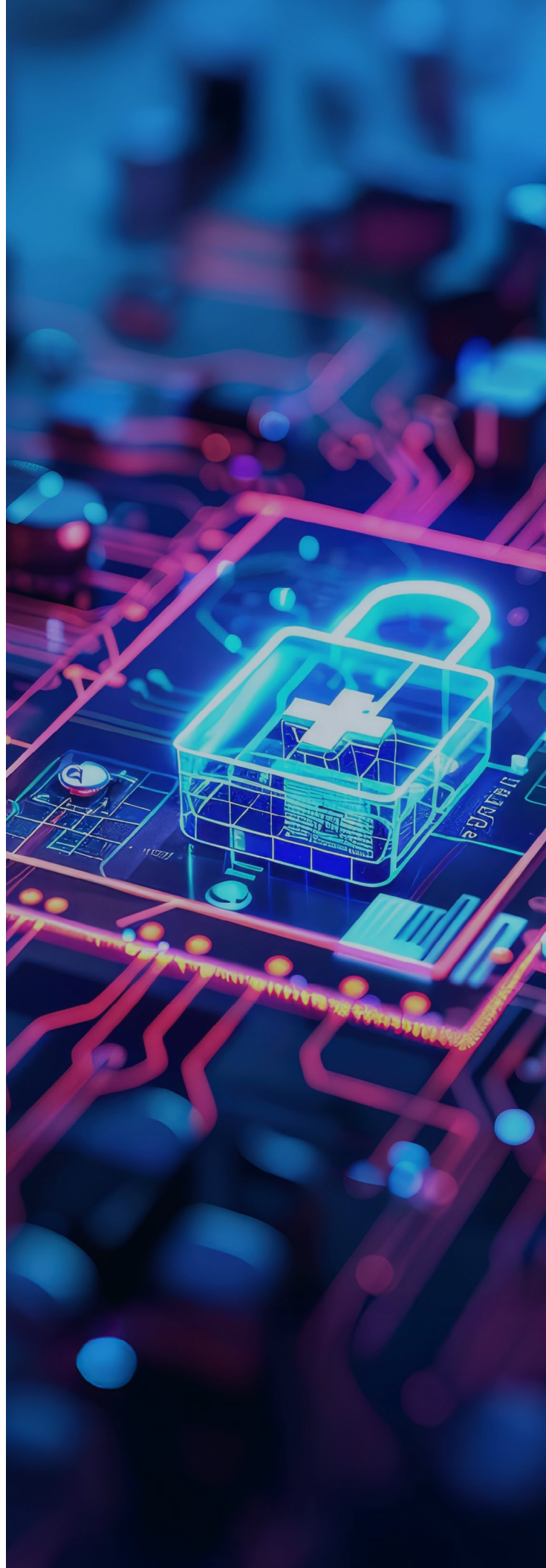
Note: *This does not mean you do not provide access to sensitive information. It does, however, mean that you should understand why you are giving access to sensitive data and fully understand the ramifications of what that data access means in a broader enterprise risk management sense.*

- **Validate and Filter Inputs:** Use filters to check and sanitize inputs before reaching the AI. Utilize tools like prompt fuzzing to identify potential vulnerabilities during development.
- **Regular Testing and Simulation:** Conduct routine adversarial testing and simulations to pinpoint vulnerabilities. Simulate prompt injection attacks to understand how the AI might respond to manipulated inputs.
- **Human Oversight for Critical Tasks:** Human review of AI outputs should be required for high-risk decisions, adding a layer of verification to minimize errors caused by prompt injection.
- **Limit Integration Points:** Reduce the AI's exposure to external systems and untrusted data sources. Separate trusted and untrusted inputs to reduce the risk of malicious activities.

Enhancing the Security and Integrity of AI Systems

As AI technologies become commonplace, attackers will find increasingly sophisticated methods to exploit vulnerabilities. It is important to remember that addressing cybersecurity risks like prompt injection does not inhibit innovation; it ensures innovation is grounded in responsibility and trust.

Recklessness is not innovation. Overlooking vulnerabilities in pursuit of rapid AI deployment can lead to breaches of trust and operational disruptions. A secure AI ecosystem protects sensitive data and critical operations and empowers teams to innovate confidently, unlocking AI's transformative potential safely and effectively.



Chapter 3 - Understanding and Addressing Sensitive Information Disclosure in AI Systems

Sensitive information disclosure occurs when AI systems unintentionally share private or confidential information. Organizations that utilize AI to handle sensitive data must understand this risk and take proactive steps to prevent it. This article explores how AI can inadvertently expose confidential information and outlines the measures necessary to prevent such breaches.

? What Is Sensitive Information Disclosure?

Sensitive information disclosure occurs when an AI system processes a request that reveals restricted information. This can happen for several reasons, including data mishandling, processing errors, or security vulnerabilities that enable unauthorized access to sensitive data. This issue can impact any organization that uses AI systems to manage or share sensitive information. AI can leak data without appropriate safeguards, resulting in privacy breaches, compliance issues, or damage to an organization's reputation.

☆ Why Sensitive Information Disclosure Matters to Organizations

Mismanagement of sensitive data when utilizing AI systems can lead to serious repercussions. Therefore, establishing a strong organizational culture that emphasizes data security is critical. Organizations should foster an environment where all employees recognize the importance of sensitive data to the mission, understand their roles, and know the risks associated with data exposure.

Determining who can access data is one of the most critical responsibilities of a data owner. This process

should be guided by policies emphasizing data minimization and the principle of need-to-know access. These policies help reduce risks by restricting data exposure to a smaller group of individuals authorized to access a particular data set.

As organizations increasingly depend on AI systems, cultivating a strong culture of data security is essential. This culture, supported by leadership, should provide employees with the knowledge and tools to handle data responsibly and make informed decisions about data access.

Examples of Sensitive Information Disclosure Risks

- 1 Customer Support Systems**
AI-powered chatbots may accidentally share personal or sensitive information with customers. For example, a customer might be provided with information or access to someone else's account details.
- 2 Automated Reports and Summaries**
If not properly secured, AI that generates reports or summaries could accidentally include confidential business information, such as sales numbers, internal plans, or employee data.
- 3 Improper Access Control**
If data access is not adequately controlled, attackers could gain unauthorized access to sensitive data by bypassing security controls through carefully crafted prompts.



Strategies to Mitigate Sensitive Information Disclosure

To mitigate the risk of disclosing sensitive information, organizations can implement various measures to secure their AI systems and ensure data privacy.

- **Limit Access to Sensitive Information**

AI systems should only access sensitive data when necessary. Access should be restricted based on user roles, ensuring only authorized individuals or systems can view or interact with private information.

- **Secure AI Training Data**

AI training datasets must be stored securely, with access strictly limited to authorized personnel. Measures must be implemented to ensure the AI system does not inadvertently reveal sensitive information in its outputs. Techniques such as output filtering should protect sensitive data while maintaining the AI system's functionality.

- **Validate Inputs and Outputs**

AI systems should be designed to verify and filter inputs before processing them. This helps prevent errors in which users might manipulate the AI to reveal private information. Additionally, output should be reviewed to ensure that sensitive data is not inadvertently shared.

- **Transparency in AI Operations**

Transparency is key for reducing the risk of disclosing sensitive information. Organizations should document and communicate how their AI systems process input, generate output, and manage sensitive data. This transparency enables stakeholders to identify possible pathways for unintended disclosures and trace any errors that may occur.

Building Secure AI Systems

Addressing sensitive information disclosure is essential for maintaining the integrity and trustworthiness of AI systems. Mitigating the risk of sensitive information disclosure requires more than just technical safeguards; it also necessitates a cultural shift in prioritizing data protection at every level. Leadership must promote a security-first mindset, and employees should be equipped with the knowledge and tools needed to make informed decisions about data handling. By combining technical measures with a culture of accountability and security, organizations can harness AI technologies while safeguarding critical information assets.

Chapter 4 - Understanding and Addressing Supply Chain Risks in AI Systems

AI systems typically depend on various components from third-party sources, such as software libraries, pre-trained models, APIs, and hardware. These dependencies can introduce supply chain risks undermining security, reliability, and functionality. Effectively managing these risks is crucial for deploying AI systems that organizations can trust.

? What Are Supply Chain Risks in AI Systems?

Supply chain risks occur when dependencies on third parties create vulnerabilities or introduce malicious components into the AI ecosystem. These risks can manifest as backdoors in software, data manipulation, unauthorized access, or reliance on compromised hardware. Such vulnerabilities jeopardize the integrity of the AI system and the data it handles.

☆ Remember, AI is just software.

It is essential to understand that AI is fundamentally just software. While AI systems provide unique capabilities, they are built on the same principles as traditional software. Acknowledging this fact highlights the importance of established security techniques in managing the risks associated with AI. Well-known strategies, such as secure software development practices, dependency management, and vulnerability scanning, form a solid foundation for securing AI systems. Organizations can effectively address AI-specific vulnerabilities by utilizing methods such as these without overhauling their existing security processes.

Examples of Supply Chain Risks in AI

1 Compromised Software Libraries

Many AI applications depend on open-source libraries or pre-trained models. If these components are tampered with, they can introduce malicious code, vulnerabilities, or subtle alterations in model behavior, such as biases or harmful outputs, which can be challenging to detect and mitigate effectively.

2 Untrusted Data Sources

AI systems that depend on external data sources are vulnerable to data poisoning. Manipulated data can affect the AI's behavior, leading to biased, incorrect, or potentially harmful outputs.

3 Malicious Model Updates

AI systems that depend on regular updates to their models can face risks if those updates are compromised. A malicious update could alter the model's internal weights or architecture, creating backdoors that enable attackers to manipulate predictions under certain conditions.

4 Compromised Pre-Trained Models

A malicious actor could manipulate training data or introduce biases during model development. These issues might remain undetected until the model is deployed, potentially leading to unexpected or harmful outcomes.

Strategies to Mitigate Supply Chain Risks

Maintain Supplier and Vendor Transparency: Require suppliers and vendors to disclose their development and deployment processes and establish contracts that include security requirements.

- Require vendors to provide Software Bill of Materials (SBOMs) for all software and ensure that they validate the integrity of these components themselves. The contract should specify the following:
 - Clear requirements for the structure and content of the SBOM.
 - Regular updates to the SBOM whenever software changes or patches occur.
 - Proof of vulnerability scanning and remediation for any identified risks.
- Require Continuous Monitoring and Reporting
 - The vendor must continuously monitor for vulnerabilities in the components listed in their SBOMs.
 - Reporting obligations for identified vulnerabilities and mitigation measures should be included, with specified timeframes for completion.
- Automate and Enhance Self-Attestation
 - Require vendors to use automated tools for generating SBOMs and validating their inventory for accuracy and completeness.
 - Mandate vulnerability scanning to ensure all components in the SBOM are cross-checked against known vulnerability databases.
 - Integrate the generation and validation of SBOMs and vulnerability remediation into vendors' Continuous Integration/Continuous Deployment (CI/CD) pipelines to promote repeatability, transparency, and security.
 - Establish a secure process for vendors to submit SBOMs and automated vulnerability remediation reports, facilitating efficient review and verification.

Adopt DevSecOps Practices: Integrate security into every stage of the software development lifecycle to minimize vulnerabilities third-party components introduce.

- Automate security testing tools within CI/CD pipelines to detect vulnerabilities in dependencies and integrations during development.
- Enforce security checkpoints to ensure compliance with internal policies before any components are deployed to production.
- Regularly update and revalidate third-party components as part of routine deployment processes to maintain system integrity.

Conduct Thorough Vetting: Before integrating third-party components, it's essential to evaluate them thoroughly. Review the source code, model documentation, and data lineage to ensure they meet security and organizational standards.

- Utilize tools such as dependency scanning software to identify known vulnerabilities in software components.
- Test pre-trained models with adversarial inputs to validate their performance and identify biases or manipulations.

Use Trusted Sources: Avoid using unknown or unverified third-party components. Opt for libraries, APIs, and models from reputable and well-supported sources whenever possible.

- Create a list of approved vendors and open-source projects vetted for quality and security.
- Regularly update this list based on new assessments and threat intelligence.
- Avoid components with minimal documentation or an inactive support community, as these are more challenging to monitor and update.

Implement Continuous Monitoring: Use tools to monitor third-party components for vulnerabilities or updates. Regularly test AI systems to detect potential exploits introduced through the supply chain.

- Deploy real-time monitoring solutions to detect anomalies in integrated components or data feeds.
- Set up alerts for critical updates or vulnerabilities identified in third-party libraries and ensure timely patching.
- Use version control to track all third-party dependencies, enabling rollback if a vulnerability is discovered.

Integrate Security into Procurement: Ensure procurement policies include clauses requiring suppliers to adhere to security best practices.

- Include procurement contract clauses that require vendors to adhere to frameworks such as NIST SP 800-161 (Supply Chain Risk Management Practices for Federal Information Systems and Organizations) and NIST SP 800-218 (Secure Software Development Framework).
- Specify requirements for conducting regular security assessments and disclosing any vulnerabilities.
- Include a stipulation allowing contract termination if vendors do not meet the established security benchmarks.

Building Resilience in AI Supply Chains

A secure supply chain is essential for organizations to deploy AI systems while safeguarding critical assets. To reduce vulnerabilities, it is necessary to hold vendors accountable through precise requirements, validation processes, and ongoing monitoring. Additionally, organizations should adopt DevSecOps practices and continuously monitor their internally developed systems. By integrating security into every stage of development and proactively addressing risks, organizations can ensure a consistent and reliable approach to AI supply chain security.



Chapter 5 - Understanding and Addressing Data and Model Poisoning in AI Systems

AI systems are heavily dependent on data, and the quality and integrity of that data significantly impact their performance. However, this reliance also creates vulnerabilities. Data and model poisoning attacks occur when the data used to train or update these systems is intentionally manipulated. Such attacks can compromise the accuracy and reliability of AI outputs, potentially leading to poor decision-making or operational failures. Addressing these risks requires data validation, monitoring mechanisms, and secure development practices to protect the integrity of the models and their underlying datasets.

? What Is Data and Model Poisoning?

Data poisoning happens when an attacker inserts malicious or altered data into the training dataset, leading the AI model to learn incorrect patterns or biases. In contrast, model poisoning involves directly tampering with the model's weights or parameters, usually during updates or training. Both types of attacks can result in unintended behaviors, including incorrect predictions and security vulnerabilities.

☆ Why Data and Model Poisoning Matters

Data and model poisoning pose risks to the reliability of AI systems. Addressing these risks is essential for organizations relying on AI for operations and decision-making.

Examples of Data and Model Poisoning Risks

- 1 Biased Decision-Making**
Poisoned datasets can result in imbalanced or unfair resource allocation, risk assessment, or operational planning outcomes.
- 2 Security Breaches**
Malicious inputs can create backdoors in AI models, enabling attackers to exploit them using specific triggers and bypass typical safeguards.
- 3 Operational Failures**
Manipulated training data may cause AI systems to make erroneous decisions, such as incorrectly classifying objects in safety-critical environments like autonomous vehicles or industrial control systems.
- 4 Loss of Confidence**
Data or model poisoning can lead to inconsistent or harmful AI outputs, which may cause organizations and stakeholders to doubt the reliability of their systems. This erosion of confidence can harm an organization's reputation and reduce user trust in AI-driven processes.

Strategies to Mitigate Data and Model Poisoning

Secure Data Pipelines: Ensure data integrity during transfer and storage to prevent unauthorized tampering.

- Use encryption techniques to secure data at rest and in transit.
- Validate datasets using cryptographic hashes or checksums to detect unauthorized modifications.
- Implement access controls and logging to monitor who interacts with training data.

Vet and Monitor Data Sources: Ensure training datasets come from trusted, reliable sources to minimize poisoning risks.

- Restrict external or untrusted data sources and require thorough provenance documentation for all datasets.
- Automated tools should be used to monitor datasets for anomalies or patterns that suggest tampering.
- Regularly audit datasets to verify their accuracy and integrity over time.

Secure Model Update Processes: Protect models from tampering during updates or retraining phases.

- Enforce strict access controls and multi-factor authentication for model update processes.
- Conduct all model updates in isolated environments to prevent interference from unverified inputs.
- Log every model modification and update to maintain a detailed audit trail.

Implement Adversarial Testing: Evaluate models against potential attack scenarios to identify and address vulnerabilities.

- Simulate data and model poisoning attacks during development and evaluate the system's resilience.
- Create adversarial scenarios based on known attack vectors to identify weaknesses.
- Regularly update testing methodologies to reflect evolving threats in the AI landscape.

Building Resiliency Against Poisoning Risks

Protecting AI systems from data and model poisoning is critical for preserving their security and accuracy. By securing data pipelines, vetting data sources, and conducting adversarial testing, organizations can reduce vulnerabilities and safeguard the accuracy and integrity of AI outputs. These efforts are foundational to ensuring AI technologies deliver consistent and dependable results in any application.

Chapter 6 - Understanding and Addressing Improper Output Handling in AI Systems

AI systems assist in decision-making, improve operational efficiency, and automate complex processes. However, if the output is not managed carefully, it can result in significant organizational issues, such as misleading inaccuracies and inadvertent exposure of sensitive information. These risks can undermine the reliability and effectiveness of AI systems, posing potential legal, ethical, and reputational challenges for organizations.

? What Is Improper Output Handling?

Improper output handling happens when AI systems produce responses that do not meet accuracy, safety, or relevance standards and are not correctly validated or filtered. This issue can arise due to inadequate post-processing, a lack of safeguards, or a failure to consider contextual nuances.

☆ Why Improper Output Handling Matters?

Improper handling of outputs can undermine the effectiveness of AI systems. For instance, outputs might accidentally reflect patterns found in training data, such as favoring certain operational practices or misinterpreting customer interactions in automated responses. If these issues are not addressed, they can weaken the operational integrity of AI systems and decrease their ability to deliver reliable and meaningful results that meet the organization's needs.

Examples of Improper Output Handling Risks

- 1 Disclosure of Sensitive Information: Due to insufficient safeguards, outputs from AI systems may unintentionally include sensitive data, such as proprietary or personally identifiable information (PII).
- 2 Inaccurate Responses: AI systems in advisory roles, like customer support or operational planning, may produce outputs based on incomplete or misinterpreted data, leading to errors or poor recommendations.
- 3 Lack of Contextual Relevance: Systems that generate outputs without context awareness may produce irrelevant or misleading responses, negatively affecting decision-making and user trust.

Strategies to Mitigate Improper Output Handling

Validate and Filter Outputs: Ensure outputs meet predefined standards for accuracy and relevance before being presented to users.

- Develop rule-based filters to detect and correct outputs that include sensitive or inappropriate information.
- Implement cross-checking mechanisms to compare outputs with trusted data sources for consistency.
- Incorporate user feedback loops to refine and improve output accuracy over time.

Enhance Context Awareness: Ensure the AI system comprehends its operational context and responds appropriately.

- Adjust models to meet the specific needs of their intended use cases.
- Provide metadata or context tags during inference to guide the generation of outputs.
- Implement limitations that restrict responses to the specified scope of the system.

Monitor and Audit Outputs Regularly: Regularly assess outputs to ensure consistency and compliance with established standards.

- Use analytics tools to identify patterns or anomalies in system outputs.
- Perform regular reviews of outputs to evaluate their alignment with system requirements.
- Test the system by simulating edge cases to see how it responds under unusual conditions.

Establish Human Oversight for Critical Applications: Human review is required before outputs are acted upon in complex or sensitive scenarios.

- Establish workflows in which subject matter experts evaluate flagged outputs for a secondary review.
- Ensure that human approval is part of decisions related to resource allocation and sensitivity workloads such as healthcare decisions.
- Use escalation paths for outputs identified as potentially problematic by automated systems.

Improving Output Reliability in AI Systems

Addressing improper output handling effectively strengthens AI systems' reliability and relevance. Organizations can enhance the security and effectiveness of their AI solutions by implementing measures such as validation processes, context-aware design, and human oversight. These strategies ensure that AI systems operate reliably and produce consistent, actionable results aligned with organizational priorities. By taking these steps, organizations can reinforce confidence in AI-driven outcomes, helping them achieve critical objectives while maintaining trust and accountability in their operations.

Chapter 7 - Understanding and Addressing Excessive Agency in AI Systems

As AI systems take on more complex roles, their ability to make decisions and perform tasks independently presents significant opportunities to optimize operations and foster innovation. However, this growing autonomy also introduces new challenges, mainly when these systems function outside their intended scope. Without careful management, such overreach can result in unintended consequences that undermine trust, disrupt workflows, or compromise security. To address these risks, organizations must carefully balance leveraging AI's capabilities and maintaining clear operational boundaries.

? What Is Excessive Agency?

Excessive agency occurs when AI systems are given permissions or capabilities beyond their intended purpose, resulting in actions that surpass their defined roles.

For example, a pharmacy chatbot designed to assist customers with prescription refills could unintentionally alter medication dosages or approve unauthorized requests if it has overly broad permissions. Scenarios like these highlight the importance of thoroughly understanding business processes and context to ensure that AI systems operate as expected.

☆ Why Excessive Agency Matters to Organizations

AI autonomy must be carefully controlled to prevent potential liability and operational disruptions. Leaders must define clear boundaries to ensure AI actions remain predictable. Without adequate oversight, these systems may take unauthorized actions, expose sensitive data,

or disrupt critical workflows. The risks become even more significant in organizations where AI interacts with sensitive systems or data. Addressing these vulnerabilities is necessary to ensure that AI systems operate within the guidelines set by business and technical leaders.

This does not mean that AI should not be granted autonomy. Instead, responsible leaders must deliberately assess the scope of autonomy they provide and understand its operational impact. Effective governance requires clearly defining AI's decision-making boundaries and ensuring its actions remain predictable, accountable, and aligned with mission priorities.

Examples of Excessive Agency Risks

- 1 **Unauthorized Actions:** AI systems with excessive permissions can unintentionally delete, modify, or share sensitive information.
- 2 **Escalated Privileges:** Broad permissions can enable an AI system to carry out actions outside its intended purpose, including authorizing unintended actions.
- 3 **Misaligned Autonomy:** Systems designed to operate independently may make decisions that conflict with organizational priorities or compliance requirements.
- 4 **Inter-system Dependencies:** AI systems with integration capabilities might inadvertently trigger unintended actions in connected tools or applications.

Strategies to Mitigate Excessive Agency

Establish Clear Boundaries for AI Autonomy: to ensure that AI systems operate safely and effectively, it is essential to restrict their access and capabilities to only what is necessary for their specific tasks:

- Adjust permissions so the AI can access only the tools and data essential for its role, preventing unnecessary or unintended actions.
- Regularly review and update permissions to address changing needs or remove outdated access.
- Assign permissions according to clearly defined roles to ensure clarity regarding what AI is allowed and not allowed to do.

Set Clear Limits for AI Operations: To ensure that AI systems stay aligned with organizational goals, it is necessary to establish and enforce clear boundaries around their tasks and actions.

- Define the system's purpose, eliminating ambiguity regarding its responsibilities.
- Establish constraints that ensure AI decision-making aligns with organizational priorities and guidelines.
- Integrate operational boundaries directly into the system architecture to strengthen these safeguards.

Monitor and Log AI Activities: To maintain accountability and prevent misuse, it's essential to monitor AI systems and their actions:

- Set Automated Alerts: Set up monitoring systems to identify actions that exceed expected results.
- Maintain Comprehensive Records: Document all AI decisions and interactions to establish an auditable trail for transparency.
- Analyze Patterns: Regularly review logs to detect trends or behaviors that could signal potential risks.

Ensure Human Oversight for Key AI Decisions: Integrating human involvement is crucial for sensitive operations to ensure control and accountability.

- Require Human Approvals: Design workflows that require explicit human authorization for critical actions, such as handling sensitive data or allocating significant resources.
- Involve Experts: Assign subject matter experts to evaluate flagged AI actions, ensuring that decisions align with organizational goals and standards.
- Escalate When Necessary: Establish clear escalation paths for reviewing potentially problematic activity.

Simulate and Test Scenarios: Examining how AI systems respond to complex or unexpected scenarios is essential for identifying weaknesses and ensuring dependable performance:

- Run Controlled Simulations: Test the system using scenarios replicating potential attack vectors or unexpected inputs.
- Examine Edge Cases: Assess how the AI handles unusual or extreme situations to confirm it adheres to established rules.
- Adapt Testing Methods: Regularly update simulation strategies for emerging risks and threats.

Building Confidence in AI through Responsible Oversight

Effectively managing excessive agency in AI systems goes beyond mitigating risks; it involves creating systems that organizations can trust to operate within their intended scope and deliver meaningful results. Organizations can balance control and innovation by defining clear boundaries, embedding safeguards, and integrating human oversight. This approach ensures that AI remains a reliable tool that supports operations without introducing unnecessary risk.

Chapter 8 - Understanding and Addressing System Prompt Leakage in AI Systems

System prompts are essential to an AI system. Unlike user-provided prompts, these are embedded instructions that guide how a model processes input and generates output. When system prompts are exposed, they give adversaries important information that can be used to bypass access restrictions, alter decision-making rules, or extract sensitive information. Protecting these prompts is important because their compromise can erode the system's reliability and predictability and undermine the AI's overall functionality.

When these prompts are exposed, the system's output may become unreliable, leading to decisions or automated actions that stray from the organization's intended outcomes. Even minor compromises in the prompts can cause errors or security breaches, undermining the accuracy and reliability of the AI's role in critical processes. Addressing this vulnerability is essential to prevent long-term disruptions that could degrade overall performance and erode trust.

? What Is System Prompt Leakage?

System prompt leakage happens when an AI system's internal instructions or configurations are unintentionally revealed or mismanaged. These prompts, which are not visible to users, may contain sensitive operational details, decision-making rules, or access parameters. When such information is exposed, malicious actors can exploit it to manipulate system behavior or gain unauthorized access.

For instance, a leaked prompt might include instructions that prioritize specific outputs or restrict certain content. If malicious individuals access this information, they could craft inputs that bypass safeguards or disrupt the AI's intended functionality.

☆ Why System Prompt Leakage Matters.

System prompt leakage can seriously compromise the reliability of AI systems. Exposed prompts give attackers critical insights into system operations, allowing them to exploit weaknesses, manipulate responses, or extract sensitive information.

Examples of System Prompt Leakage Risks

- 1 **Data Exposure:** Prompts that disclose internal logic and operational processes can be used to gain unauthorized access.
- 2 **Manipulated Behavior:** Exposed prompts can enable attackers to create inputs that bypass intended safeguards or disrupt system outputs.
- 3 **Operational Disruptions:** Exposed system prompts can enable attackers to introduce malicious inputs, leading to errors that interfere with normal operations and compromise system effectiveness.
- 4 **Trust Erosion:** System prompt leakage can lead to outputs that deviate from expected behavior. Once a system's underlying logic is compromised, its role in supporting key processes may no longer be viewed as reliable.

Strategies to Mitigate System Prompt Leakage

Isolate and Secure Prompts: System prompts require specialized protections to prevent exposure or tampering.

- Use strong encryption methods to protect prompts from interception or theft, ensuring they are only decrypted within secure, controlled environments during use.
- Implement access controls tailored to the AI system's security architecture, ensuring that only authorized system processes or personnel with a clear need can access prompts.
- Store prompts in secure environments isolated from user-facing APIs and inputs to prevent adversaries from exploiting vulnerabilities through other attack vectors.

Implement Prompt Segregation: Segregate system prompts from user-facing inputs to prevent exposure and mitigate the risks of prompt manipulation or leakage.

- Prompts should be stored separately from user-generated data in environments inaccessible to external applications or users.
- Apply prompts dynamically during runtime to pass prompts to the AI model during execution, ensuring they are never directly embedded or visible in the system.
- Ensure that prompts are excluded from logs, error messages, or debugging information available to users, as this could unintentionally expose sensitive instructions.

Monitor and Audit Prompt Access: Continuously monitor and audit system prompt access to detect unauthorized activities.

- Record all interactions with system prompts, including access, modifications, and any attempts to retrieve or alter them.
- Leverage automated monitoring solutions to identify suspicious patterns or unusual access attempts that could indicate prompt leakage or exploitation.
- Review access logs regularly to uncover potential vulnerabilities or unauthorized actions and take corrective measures before they escalate.

Test for Prompt Leakage Scenarios: Perform regular tests to identify vulnerabilities that might expose system prompts.

- Develop test scenarios that simulate adversarial inputs aimed at extracting system prompts.
- Assess how the AI system manages abnormal or unexpected situations to ensure that system prompts remain confidential.
- Update testing strategies to address changing AI security threats, ensuring that new attack vectors are regularly evaluated and mitigated.

Embed Safeguards in System Design: Incorporate security measures into the AI system's core architecture to safeguard prompts from exposure.

- Avoid placing sensitive information directly in prompts; use secure placeholders that the system can resolve dynamically during execution.
- Implement strict input validation to eliminate malformed or suspicious inputs before they reach the model.
- Implement context locking to isolate system-level instructions from user inputs, ensuring that user-provided data cannot override or access internal prompts.

Building Resilience Against System Prompt Leakage

Protecting system prompts is essential for maintaining the stability of AI systems, ensuring they deliver reliable and trustworthy outcomes. If these prompts are compromised, errors can distort the system's input processing, leading to decisions that no longer align with the intended business objectives or mission.

To address this issue, a proactive approach is needed to incorporate security into the system's architecture from the outset of the Systems Development Lifecycle (SDLC). This ensures that any attempted manipulations are detected and neutralized before they can impact the system's performance. Embedding defensive measures within the system's architecture enables it to differentiate between legitimate inputs and those designed to exploit its internal logic.

Chapter 9 - Understanding and Addressing Vector and Embedding Weaknesses in AI Systems

System prompts are essential to an AI system. Unlike user-provided prompts, these are embedded instructions that guide how a model processes input and generates output. When system prompts are exposed, they give adversaries important information that can be used to bypass access restrictions, alter decision-making rules, or extract sensitive information. Protecting these prompts is important because their compromise can erode the system's reliability and predictability and undermine the AI's overall functionality.



What Are Vector and Embedding Weaknesses?

Unsecured architecture supporting AI system embeddings can create vulnerabilities that undermine the integrity of AI systems. When these embeddings are not adequately protected, they expose underlying data structures that adversaries can exploit. If models are trained on manipulated data or lack necessary safeguards, they may misrepresent relationships or produce erroneous interpretations, ultimately impacting decision-making. Moreover, weaknesses in storage and access controls can make embeddings susceptible to analysis, allowing adversaries to discern their structure and extract significant patterns. If embeddings are inadequately secured, they can be exploited to reveal connections within the data, resulting in unauthorized access to hidden relationships or even the reconstruction of sensitive information. These vulnerabilities also provide opportunities for manipulating AI-driven outputs by altering the representations that influence retrieval and decision-making processes.

These risks extend beyond individual AI applications, affecting the broader model that relies on embeddings to process and interpret data. When these systems are not adequately secured, they can create vulnerabilities

that may compromise response accuracy and protect sensitive information. Unaddressed vulnerabilities can weaken the reliability of AI-driven processes, making it easier for adversaries to exploit these flaws for unintended purposes. Strengthening security throughout the lifecycle of these systems is essential to ensure that AI models remain resilient and continue to function as intended without external interference.



Why Vector and Embedding Weaknesses Matter to Organizations

Weaknesses in vectors and embeddings create vulnerabilities that disrupt the functioning of AI systems, impairing their ability to process information accurately and securely. If these structural flaws are not addressed, AI-driven processes may misinterpret inputs, resulting in outcomes that do not meet expectations. Such failures compromise the reliability of automated decision-making and heighten the risk of unintended consequences. Security concerns arise when these vulnerabilities expose sensitive data to unauthorized access, providing opportunities for adversaries to exploit.

These weaknesses can have significant consequences in environments where AI is used, such as in the healthcare industry. For instance, in healthcare, a system lacking proper safeguards may produce outputs compromising patient safety, leading to decisions that could seriously endanger individuals' health. Additionally, gaps in medical data protection, where vectors and embeddings serve as the foundation for outputs, raise concerns about the reliability of AI-driven diagnostics and treatment recommendations. Without a structured approach to address these vulnerabilities, the stability of AI implementations becomes increasingly uncertain, introducing risks that affect both patient outcomes and the credibility of institutions.

Examples of Vector and Embedding Weakness Risks

- 1 **Data Leakage:** Inadequately secured embeddings reveal patterns that enable adversaries to understand information about the original dataset. Attackers analyzing embedding structures can identify if specific data points were included in the training process, increasing the risk of targeted data reconstruction.
- 2 **Adversarial manipulation:** This technique exploits vulnerabilities in how embeddings represent relationships, which can distort AI-driven output by incorporating misleading inputs. When these manipulated inputs are introduced, AI systems may misinterpret their meaning, leading to incorrect classifications that diverge from expected behavior.
- 3 **Index Exploitation:** Improperly secured vector databases can enable attackers to extract or modify stored embeddings. Without robust access controls, adversaries may gain unauthorized access to these systems, allowing them to retrieve sensitive embeddings or insert altered data.
- 4 **Data Poisoning:** Data relationships become unreliable when embeddings are derived from manipulated data. Poisoned data alters the model's learning process, embedding misleading associations that negatively impact decision-making accuracy, making them difficult to detect and mitigate.

Strategies to Mitigate Vector and Embedding Weaknesses

Secure Vector and Embedding Data: Embeddings define relationships within data, making them essential for functionality and potential targets for exploitation. Securing embeddings prevents unauthorized access and manipulation that could alter AI-driven decision-making.

- **Encryption:** Embeddings must be encrypted at rest and during transmission to prevent unauthorized access. Without encryption, an attacker who gains access to storage systems or intercepts data during transit could extract and analyze the embeddings, potentially revealing sensitive information.
 - **Access Controls:** Controlling access is necessary to prevent misuse and unauthorized modifications to embeddings. Weak authentication measures can enable unauthorized users to access or modify embeddings. Strong access controls ensure that only authorized individuals or systems can interact with these representations.
 - **Dataset Integrity:** Regular validation of datasets is essential to ensure that embeddings are created from reliable sources and remain unaffected by tampered or malicious inputs. If compromised data is introduced during the embedding generation process, the resulting representations may encode misleading relationships, negatively impacting the AI system.
- Harden Vector Search Systems:** Vector search systems help index and retrieve high-dimensional numerical representations of data, allowing AI models to locate and compare stored embeddings efficiently. However, when security measures are insufficient, adversaries can manipulate queries to affect retrieval results. This can expose patterns that reveal hidden relationships within the data.
- **Authentication and Access Controls:** Like any other database, implementing strict authentication and role-based permissions is essential to limiting and controlling access to vector databases. Without proper controls, attackers can extract embeddings or alter stored representations, resulting in compromised AI-driven processes.
 - **Query Monitoring:** Analyze query patterns in vector search systems to detect unauthorized activities that may indicate data extraction attempts. If a vector search system lacks sufficient protection, an attacker can issue different queries to uncover relationships between embeddings. They can gradually reconstruct the underlying data structures by analyzing how the system responds to various inputs.
 - **Rate Limiting:** Rate limiting restricts the number of queries that a user or system can send within a specific timeframe. This helps minimize the risk of data inference attacks, where an attacker tries to extract sensitive embeddings. When a vector search system permits unlimited queries, it opens the door for attackers to use brute-force probing or statistical analysis to determine relationships between embeddings.

Validate and Monitor Embedding Outputs: Poorly evaluated embeddings may encode incorrect relationships, propagating errors through AI-driven processes. Monitoring embedding behavior helps identify deviations, enabling corrective measures to be taken.

- **Anomaly Detection:** Anomaly detection in embeddings involves analyzing the distribution of embeddings within a high-dimensional space to identify deviations from expected patterns. Detection tools monitor how embeddings cluster and interact in the vector space. When an embedding significantly deviates from the typical distribution, it indicates that the generated data may have been altered. Once anomalies are identified, further investigation is needed to determine whether they are due to natural variations in the data, adversarial inputs, or system malfunctions.
- **Adversarial Testing:** Adversarial testing assesses the resilience of models by presenting them with intentionally crafted inputs aimed at manipulating their outputs. These adversarial inputs are generated using techniques that exploit how models interpret data. During this testing process, embeddings are examined for unexpected changes in similarity scores, incorrect clustering, or abnormal retrieval patterns when subjected to these adversarial inputs. The results of this testing inform the development of mitigations and help refine defensive strategies.
- **Ongoing Model Refinement:** As adversarial strategies evolve, attackers develop new methods to manipulate embeddings, bypass security measures, and exfiltrate sensitive data. Without ongoing model refinement, embeddings may become vulnerable to these emerging threats. Models must be updated periodically to improve how they encode relationships and manage adversarial manipulations. This refinement process includes retraining with validated datasets, fine-tuning embedding parameters, and enhancing preprocessing techniques.

Building Resilience Against Vector and Embedding Weaknesses

Vectors and embeddings shape how AI systems interpret and retrieve information, making their security essential to maintaining reliable performance. Weaknesses in the design, storage, and retrieval of data create opportunities for adversaries to interfere with how AI models process information, leading to results that deviate from expectations. If embeddings are exposed, the underlying data structures can be analyzed, revealing patterns that should remain confidential. When vector search systems lack proper safeguards, the retrieval processes become vulnerable to manipulation. Addressing these risks requires transparency about how AI models generate and retrieve embeddings, ensuring any vulnerabilities are identified and resolved before they can be exploited.



Chapter 10 - Understanding and Addressing Inaccurate or Misleading Outputs in AI Systems

Inaccurate outputs weaken the trustworthiness of AI systems, particularly large language models (LLMs), by generating responses that appear credible but lack accuracy. When incorrect information is presented as a fact, users may unknowingly rely on flawed outputs that seem correct but ultimately are fabricated. The underlying causes can stem from weaknesses in training data, insufficient prompt preparation, or inherent model limitations, leading to hallucinations that convincingly present false information. When AI systems produce authoritative but incorrect responses, they can mislead users into adopting unreliable conclusions or introducing errors that influence broader organizational operations. Addressing these challenges requires continuous evaluation of AI-generated content, ensuring safeguards are in place to detect unreliable outputs before they can cause harm. Without this level of scrutiny, inaccuracies can persist, creating deeper risks that extend beyond a single flawed response and affect the overall integrity of AI-driven processes.



What Are Inaccurate or Misleading Outputs?

Inaccurate or misleading outputs occur when AI systems generate responses that do not align with factual accuracy, meaning they fail to provide accurate and verifiable information. These systems may also produce outputs that do not adequately address the user's intended context or requirements. These

problems can originate from errors in the training data, misinterpretations of the input context, or deliberate actions by external actors. Such failures reduce the usefulness of AI systems, exposing individuals and businesses to potential reputational damage.

A notable example involves an attorney who, in 2023, used ChatGPT to prepare a legal brief for a federal court case. The AI-generated brief included six fictitious case citations, which Schwartz unknowingly submitted as legitimate precedents. When the inaccuracies were exposed, the court determined that the lawyer and his firm had acted in bad faith, resulting in a \$5,000 sanction imposed by U.S. District Judge P. Kevin Castel. This case damaged the firm's reputation and highlighted the risks of unverified AI-generated content. ([Reuters](#), [Legal Dive](#))

More recently, in February 2025, additional attorneys faced potential sanctions after submitting fabricated case citations in a lawsuit against a large retailer. The case involved a Wyoming family alleging that a hoverboard purchased from Walmart malfunctioned, causing a fire that destroyed their home and resulted in severe burns and emotional trauma. Upon review, U.S. District Judge Kelly Rankin discovered that eight cases cited in the plaintiffs' pretrial motion were non-existent. Judge Rankin ordered the plaintiffs' attorneys to explain the origin of these citations and why they should not face disciplinary action. In response, the lawyers admitted that their internal AI platform had "hallucinated" the cases, leading to the inclusion of false information in the court filing. ([Reuters](#), [5 News Online](#))



Why Inaccurate or Misleading Outputs Matter to Organizations

When AI produces factually incorrect information, it undermines trust in its reliability, misguides users, and disrupts decision-making processes. The consequences of unreliable AI outputs can be particularly damaging in industries where accuracy is critical. In legal settings, as seen in the cases above where attorneys unknowingly submitted fabricated case citations, reliance on incorrect AI-generated information can result in professional sanctions and reputational harm. In healthcare, incorrect recommendations could lead to misguided treatment plans, putting patients at risk. In finance, flawed AI-driven analysis may lead to poor investment decisions or regulatory non-compliance, exposing firms to financial losses and legal scrutiny. Organizations must ensure that AI-generated information is reliable and verifiable to maintain trust. When AI-driven processes fail to deliver accurate information, the resulting damage can extend well beyond a single mistake, shaping public perception and influencing the willingness of businesses and institutions to integrate AI into their workflows.

Examples of Inaccurate or Misleading Output Risks

- 1 **Hallucinated Professional Advice:** AI systems used in professional settings, such as legal or financial advisory services, can generate factually incorrect responses that appear legitimate. As seen in the above cases, AI-generated legal research can fabricate case citations that, if relied upon, lead to professional or legal consequences.
- 2 **Medical Misdiagnosis and Risk to Patient Safety:** AI tools assisting in diagnosis or treatment recommendations can misinterpret symptoms, suggest ineffective treatments, or overlook critical conditions in healthcare. If these outputs are trusted without verification, they can contribute to misdiagnoses, improper treatment, and patient harm.
- 3 **Incorrect AI-Generated Business Reports:** AI-powered tools used for corporate decision-making may produce misleading analyses based on incomplete or outdated training data. A company using AI for strategic planning could receive flawed insights into market trends, leading to poor investment decisions.

Strategies to Mitigate Inaccurate or Misleading Outputs

The following are some strategies that provide a structured approach to mitigating inaccurate or misleading AI outputs, helping organizations maintain the integrity and effectiveness of their AI systems.

Strengthening Training Data Quality and Model Validation

Organizations must ensure that AI models are trained on verifiable, representative, and current information to reduce inaccuracies.

- Establish strict data curation processes that remove outdated, incomplete, or inaccurate information from training datasets.
- Validate AI models through continuous testing, ensuring they produce accurate and reliable results across different scenarios.
- Use real-world examples and independent verification to identify and correct patterns that lead to fabricated or inaccurate responses.

Implementing Human Oversight for AI-Generated Content

AI-generated outputs should include human validation in environments where accuracy can affect life, limb, and property.

- Train users to critically assess AI-generated responses rather than treating them as definitive sources of truth.
- Establish human review as a required step for mission-critical AI-generated content.
- Develop explainability tools that allow users to understand how AI models arrive at conclusions.

Improving Prompt Design to Reduce Misinterpretation

AI-generated inaccuracies often stem from poorly constructed prompts that fail to provide enough context or clarity.

- Train users on effective, prompt engineering, ensuring they ask questions that lead to factually grounded responses.
- Develop structured input guidelines that prevent AI from generating speculative or fabricated answers.
- Refine AI systems to clarify uncertainty, enabling models to recognize when they lack sufficient data rather than filling gaps with inaccurate information.

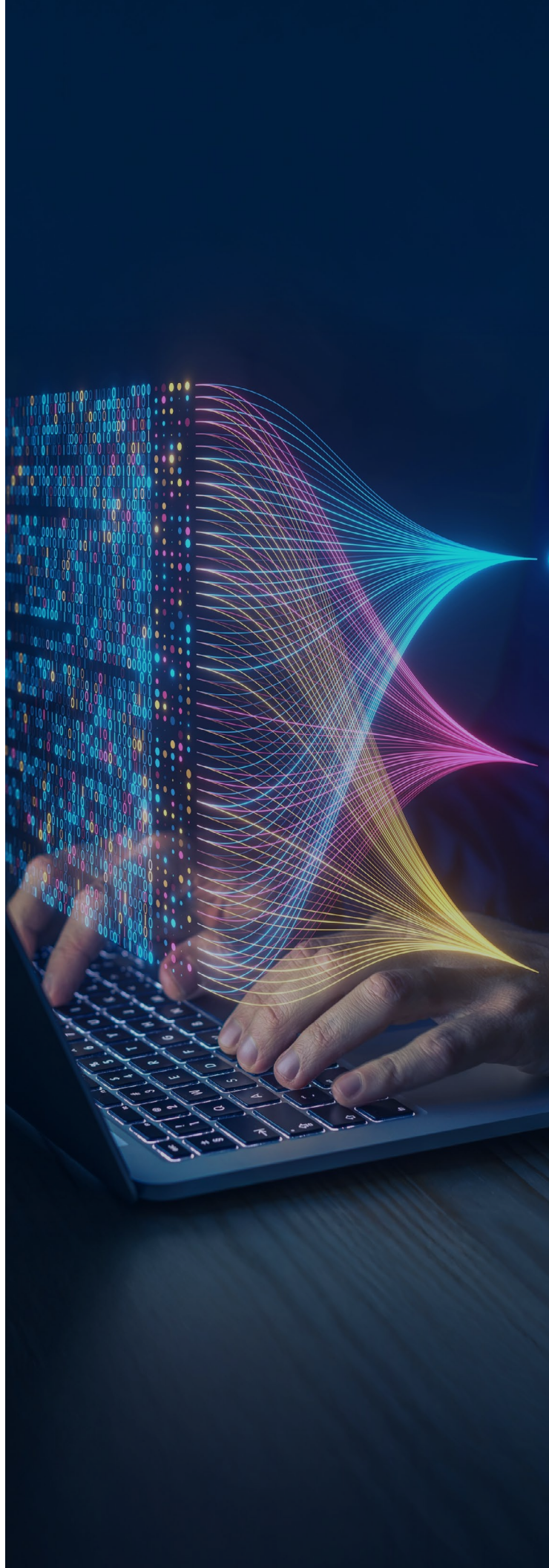
Establishing Boundaries for AI's Scope of Use

AI should not be used beyond its intended expertise. If models lack relevant data, they should not generate responses rather than risk misleading users.

- Define specific domains where AI can provide recommendations and prevent AI from responding to queries outside its scope.
- Ensure AI models do not generate authoritative-sounding responses when the information provided is speculative.
- Require AI-generated responses to indicate uncertainty when confidence is low, prompting users to verify critical information before relying on it.

Building Confidence in AI Systems

Responsible AI deployment is not just about reducing errors. It is about ensuring that these systems remain effective, trustworthy, and capable of delivering real value without compromising integrity. Organizations must implement safeguards that ensure AI models operate within their intended scope, clearly communicate uncertainty when necessary, and provide outputs grounded in verified data. Without these measures, AI risks becoming an unreliable tool that introduces confusion rather than supporting informed decision-making. Prioritizing accuracy at every stage—from data selection to model deployment—is essential to prevent misleading outputs from influencing decisions. Organizations that take a disciplined approach to managing AI can reinforce confidence in its role, ensuring that it enhances operations rather than detracts from them.



Chapter 11 - Understanding and Addressing Unbounded Consumption in AI Systems

AI systems require substantial computational resources to process data efficiently. These systems generate responses that enable automation by performing tasks typically involving human intervention. They carry out complex operations, enhancing analytical capabilities and improving efficiency across various applications. Additionally, they integrate with other platforms to facilitate seamless data exchange and ensure interoperability within an organization's technology ecosystem. While these capabilities offer significant advantages, they also bring operational risks if resource consumption is not effectively managed. If computing power is not controlled correctly, AI systems may overuse processing capacity, diminishing overall system performance. Unregulated resource usage can result in unexpected slowdowns, higher infrastructure costs, and potential service outages that disrupt essential business functions. Without proper oversight, these inefficiencies can accumulate over time, burdening IT teams and increasing operational costs and complexity.



What is Unbounded Consumption

Unbounded consumption occurs when AI systems utilize more resources than intended due to a lack of proper technical controls. When resource consumption is not managed appropriately, AI-driven applications can suffer from performance degradation, resulting in slow response times and operational instability. The financial impact of unregulated resource usage can be significant. Increased processing demand can lead to higher infrastructure costs, putting strain on the organization's budget. Overusing system resources can also result in service throttling or unexpected billing spikes, particularly in cloud environments that employ usage-based pricing models. Without proper governance, these costs can escalate quickly, creating financial strain that outweighs the benefits of AI-driven automation.

Security risks also increase when AI systems operate without defined resource constraints. Attackers can take advantage of vulnerabilities in system design by submitting resource-intensive queries that overwhelm the system's processing capacity. These attacks can lead to denial-of-service situations, preventing legitimate users from accessing critical functions. Adversaries may also manipulate AI models by providing inputs designed to consume excessive memory or computational power, further degrading performance and stability. Organizations can sustain AI-driven solutions' reliability, scalability, and security by enforcing structured resource management strategies.



Why Unbounded Consumption Matters to Organizations

Unbounded consumption can lead to operational inefficiencies beyond system slowdowns or increased costs. When AI systems lack defined resource constraints, they may produce unpredictable performance issues that affect decision-making, disrupt workflows, and overwhelm IT support teams. As organizations adopt AI technologies more widely, ensuring these solutions enhance productivity rather than create new challenges necessitating constant intervention is essential.

A significant issue is the unpredictability of system performance. AI applications developed without resource constraints can behave inconsistently, often using more processing power or memory than anticipated. This unpredictability makes it challenging to maintain stable operations, as fluctuations in resource demand can lead to slowdowns or outages. As a result, IT teams find themselves in a continuous cycle of troubleshooting and adjustments, which hinders their ability to focus on long-term improvements or innovation.

Another significant impact is the strain on interconnected systems. AI does not operate in isolation; it integrates with databases, cloud platforms, and internal networks. When one component consumes excessive resources, it can degrade performance across the entire environment. Systems that support multiple applications may struggle under this load, affecting AI functions and critical business operations. Organizations risk creating bottlenecks without structured limits that weaken their overall technology infrastructure.

A proactive approach to AI resource governance is essential to tackling unbounded consumption. Organizations should establish clear guidelines that define acceptable resource usage and implement monitoring tools to detect overconsumption before it leads to disruptions.

Examples of Unbounded Consumption in AI Systems

- 1 **AI Overloading Cloud Resources:** AI applications operating in cloud environments without clearly defined resource limits can excessively use compute and storage resources. For instance, an AI-powered data processing system that continuously ingests and analyzes large datasets without constraints can significantly increase infrastructure costs.
- 2 **Unregulated API Calls in AI Chatbots:** Chatbots designed for customer service that do not have API rate limits can overwhelm backend systems with requests during peak usage times. When a chatbot responds to thousands of simultaneous queries, it can strain external data sources, resulting in delays, lower response quality, or even service outages.
- 3 **AI-Powered Fraud Detection Gone Wrong:** An AI fraud detection system that analyzes financial transactions in real time can consume excessive processing power if it is not optimized for efficiency. If the system scrutinizes every transaction with the highest level of detail instead of using risk-based prioritization, it may slow down banking operations and delay legitimate transactions.

Strategies for Controlling AI Resource Consumption

Set Resource Usage Limits: Establishing clear thresholds for system operations is essential in preventing excessive consumption. This creates boundaries that protect infrastructure from overload and effectively control costs, ensuring that AI remains a viable and manageable resource.

- Implement API rate limiting to limit the number of calls made per user or session.
- Set compute and memory limits to ensure resource allocation matches expected workloads.
- Utilize dynamic scaling policies to manage demand within set capacity, enabling the system to adapt to changing needs while maintaining safe limits.

Monitor and Analyze Usage Patterns: Ongoing monitoring of AI system behavior enables the early detection of anomalies, allowing for proactive adjustments to maintain efficiency and stability.

- Implement monitoring tools to gather real-time metrics on API calls, memory usage, and computational performance.
- Set alerts to notify administrators of unusual spikes in resource usage, ensuring potential issues are addressed before they escalate into significant disruptions.
- Utilize analytics to identify trends and optimize configurations for greater efficiency.

Implement Request Validation: Ensure user input matches system capabilities to avoid excessive resource demands. This filtering process prevents requests that could overload the system and protects against accidental and malicious overuse.

- Evaluate incoming requests for complexity and reject those that exceed predefined thresholds to prevent resource-intensive queries from overwhelming the system.
- Implement input size restrictions to minimize the risk of processing overly large queries and ensure the system can efficiently manage requests.
- Use request queues to regulate AI workload processing, ensuring tasks are handled in a controlled sequence rather than overwhelming system resources.

Conduct Stress Testing: Consistently assessing system performance during high-demand situations is a proactive approach to uncover vulnerabilities that may result in excessive resource usage.

- Simulate scenarios involving heavy resource usage to evaluate system resilience. Test how the AI responds under extreme conditions and identify weak points that require reinforcement.
- Test rate-limiting mechanisms to ensure they effectively prevent unlimited consumption, verifying that these controls can manage surges in demand without failure.
- Utilize stress testing results to refine system limits and scaling strategies, applying insights from these tests to enhance configurations and policies.

Design Systems for Efficient Resource Use: Improving AI architecture and workflows to reduce resource consumption is a strategic approach that enhances the architecture by minimizing waste and ensuring that the system operates within its capacity while still delivering high-quality results.

- Utilize model compression techniques like quantization, pruning, and knowledge distillation to lower computational requirements.
- Utilize caching mechanisms to store frequently used results, which will help reduce redundant processing.
- Develop workflows that emphasize efficient operations for routine tasks, ensuring that everyday functions use minimal resources while reserving more intensive processing for exceptional cases.

Ensuring AI Sustainability

Effectively managing AI resource consumption is a technical requirement and a strategic necessity for organizations aiming to scale AI responsibly. Without appropriate constraints, AI systems may become inefficient, unpredictable, and challenging to integrate into broader operations. Resource management should not be seen as a limit to innovation; instead, it is essential for ensuring that AI remains sustainable and beneficial in the long term. Organizations that proactively manage resource consumption will be better equipped to maximize the benefits of AI while maintaining control over performance, costs, and security. Organizations can develop robust and practical solutions by designing AI systems with efficiency in mind, ensuring they deliver real value without unnecessary risk.





Chapter 12 - Securing AI Is a Continuous Commitment

AI Security: Not a One-and-Done Effort

Securing artificial intelligence is not just a one-time task that ends when a model is deployed; it is an ongoing process that requires continuous assessment, adaptation, and refinement. Techniques that attackers use to exploit AI's vulnerabilities will continue to evolve. Organizations that succeed in implementing AI security will not be the ones that treat security as merely a compliance exercise. Success comes from integrating proactive defenses and effective oversight into AI management, ensuring systems remain secure against evolving threats.

This article has covered the OWASP Top 10 for Large Language Model (LLM) Applications, detailing the most critical security risks facing AI today. While AI security introduces new challenges, the fundamental principles of cybersecurity remain unchanged. The more things change, the more they stay the same. AI security is not a separate discipline—it is an extension of everything we know about securing software, data, and infrastructure. Organizations do not need to abandon traditional security models; they need to evolve them.

Although AI security brings new challenges, the core principles of cybersecurity remain the same. In many ways, the more things change, the more they stay the same. AI security is not a separate field; it builds upon our existing knowledge of securing software, data, and infrastructure. Organizations don't need to discard traditional security models; instead, they should adapt and evolve them. The following are key areas to consider while developing your strategy to secure AI systems.

ENSURING A SECURE AI SUPPLY CHAIN

AI systems do not operate independently; they rely on open-source libraries, APIs, and external data sources. Each component in this ecosystem introduces potential risks. If even one compromised dependency is integrated into an AI system, it can become an attack vector, jeopardizing an organization's security posture.

Securing the AI supply chain requires strict verification processes like those outlined in NIST SP 800-161 Rev. 1, Cybersecurity Supply Chain Risk Management (C-SCRM). Organizations must implement policies that:

- **Require transparency** from AI model providers, including documentation of training data sources and security testing.
- **Enforce Software Bill of Materials (SBOM)** requirements for software components to track dependencies and assess risks.
- **Continuous monitoring and validation of AI systems should be conducted** to ensure they have not been manipulated post-deployment.

AI supply chain security should be managed with the same level of rigor as other critical systems. Each component—data sources, models, and third-party integrations—can introduce potential vulnerabilities. Organizations must implement strict vetting processes, track the provenance of their assets, and engage in continuous monitoring to prevent compromised or manipulated AI components from jeopardizing security.

INTEGRATING SECURITY INTO DESIGN AND BUILD PROCESSES

One of the most effective ways to secure AI is to incorporate security into the development process right from the beginning rather than considering it only as an afterthought. Adequate software security and systems design focus on “shifting left” in the Software Development Lifecycle (SDLC), ensuring that secure design principles, threat modeling, and vulnerability management are integrated. AI should follow this same approach.

Frameworks such as NIST SP 800-160 Volume 2, “Developing Cyber-Resilient Systems,” offer guidance on integrating security into system design. Such an approach is particularly important for AI security, where attacks like data poisoning, adversarial manipulation, and prompt injection can compromise the integrity of models. Instead of concentrating solely on detecting issues after deployment, organizations should design resilient AI systems right from the start.

A secure AI development process ensures that model training, validation, and deployment occur within a structured, well-governed security framework. This involves securing training datasets, implementing adversarial testing, and applying continuous monitoring to detect unexpected model behaviors. Organizations that adopt cyber-resilient AI development will be in a much better position to mitigate evolving threats.

ESTABLISHING SECURE AI DEVELOPMENT PIPELINES

Just as DevSecOps transformed traditional software security by integrating security into the entire development lifecycle, AI security must follow a similar path. Secure AI development pipelines ensure that models are created, tested, and deployed within a robust security framework.

Guidance from NIST SP 800-218, the Secure Software Development Framework (SSDF), emphasizes best practices for integrating security into software development pipelines. These best practices include:

- **Version control and integrity checks** to prevent unauthorized modifications to AI models, training datasets, and system components.
- **Security Testing** to identify vulnerabilities in system components, model configurations, and the mechanisms through which AI generates outputs.
- **Strict access controls** to prevent unauthorized model retraining or fine-tuning that could introduce unauthorized changes and unexpected results.

AI security should not depend solely on manual oversight. Automated tools must be integrated into AI pipelines to identify anomalies, adversarial inputs, and unexpected model behavior. These mechanisms must be continuously updated as attackers devise new methods to exploit AI-driven systems.

ADDRESSING THE RISKS OF GENERATIVE AI

As AI systems advance in capability, they become more vulnerable to misuse, manipulation, and unintended consequences. Generative AI poses challenges that extend beyond traditional security concerns as it not only processes inputs but also creates new content.

The release of NIST SP 800-218A, Secure Software Development Practices for Generative AI and Dual-Use Foundation Models, offers security recommendations designed explicitly for generative AI. Areas covered under this document include:

- Controlling training data integrity to prevent data poisoning and bias injection.
- Implementing strong output filtering to prevent unauthorized, misleading, or harmful AI responses.
- Apply security guardrails to restrict unauthorized fine-tuning or prompt engineering attacks.

As AI-generated content becomes increasingly integrated into enterprise decision-making, organizations must establish governance frameworks to ensure human oversight, ethical constraints, and the development of AI models that prioritize security.

Future Areas for Research and Advancement

Although this article has addressed the most pressing AI security risks, several areas still require ongoing research and development. Here are some suggested areas for future research:

ENHANCING AI EXPLAINABILITY AND TRANSPARENCY

A challenge in AI security and AI in general is understanding how models make their decisions. AI functions as a black box, complicating efforts to audit decision-making, manipulation, or adversarial interference. Future research should concentrate on creating mechanisms for AI explainability, enabling organizations to:

- Trace AI decision-making processes to understand how decisions are made and to identify potential manipulations.
- Develop tools that detect when adversarial inputs have influenced an AI system.

IMPROVING ADVERSARIAL AI DETECTION AND DEFENSE

Adversarial attacks on AI models are growing increasingly sophisticated. Organizations must invest in AI-driven security monitoring to detect subtle manipulations, adversarial inputs, and unauthorized behavioral changes in real time. This includes:

- Deploying dynamic adversarial testing frameworks to continuously stress-test AI models.
- Developing proactive defense mechanisms that prevent AI exploitation before attackers can leverage vulnerabilities.

SECURING AI MODEL UPDATES

AI models can be updated and improved over time. However, this introduces risks, as updates can be exploited to inject vulnerabilities into AI systems. Future research should focus on:

- Developing validation mechanisms for AI model updates.
- Implementing safeguards that prevent unauthorized modifications to deployed AI systems.

ADDRESSING THE RISKS OF SYNTHETIC DATA

Synthetic data is used to train AI models without depending on real-world data, which may contain sensitive information. However, the use of synthetic data introduces new risks, such as the possibility of embedding unintended decision-making properties or revealing proprietary model behaviors. Further research is needed to address these concerns.

- Develop security validation techniques for synthetic datasets.
- Determine how synthetic data impacts AI model generalization and robustness.

AI-POWERED THREAT DETECTION FOR CYBERSECURITY

AI is increasingly utilized for cyber defense; however, research is necessary to ensure that AI-driven threat detection systems are high-performing. Key areas of focus include:

- Developing AI models that detect cyber threats in real time.
- Enhancing the ability of AI-driven defense systems to adapt to evolving threats.

Chapter 13 – Supplemental – Review of NIST AI 100-2e2025

While this article focuses on securing Large Language Model (LLM) applications through the lens of the OWASP Top 10 for LLMs, it is important to recognize that broader adversarial threats to AI systems extend beyond the scenarios outlined in OWASP. To complement this article, chapter 13 summarizes key considerations drawn from the National Institute of

Standards and Technology's (NIST) publication Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (NIST AI 100-2e2025). This document can be referenced on the NIST website at the following URL: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf>

Understanding Predictive AI and Generative AI

Securing AI systems begins with recognizing that not all AI operates similarly. Predictive AI analyzes patterns in existing data to predict a label, category, number, or decision. It does not create new content — its role is to forecast or classify based on what it has learned.

In contrast, Generative AI creates new content based on learned patterns from its training data. It generates text, images, code, audio, or other outputs beyond direct predictions.

Understanding this distinction is critical for designing effective AI security strategies. Predictive AI systems are primarily attacked by corrupting the data used for prediction, through techniques such as evasion attacks, poisoning attacks, or privacy inference. Generative AI systems, by contrast, are attacked by manipulating what they generate. For example, a chatbot can be tricked into leaking or misusing its generation capabilities through prompt injection. This distinction between predictive and generative models shapes the structure of adversarial threats, defense strategies, and security assurance requirements.

EXAMPLES OF PREDICTIVE AI INCLUDE:

- A fraud detection system that predicts whether a transaction is fraudulent.
- An email filter determines whether a message is spam.
- A medical imaging system identifying signs of cancer.

Main goal: Analyze → Predict → Classify

EXAMPLES OF GENERATIVE AI INCLUDE:

- A chatbot writing natural language responses.
- An image generator creating artwork from a text prompt.
- A coding assistant generates functions based on user input.

Main goal: Analyze → Create → Generate

Aspect	Predictive AI	Generative AI
Purpose	Predicts an outcome based on input data	Generate new content based on learned patterns
Action	Classifies, forecasts, or estimates	Creates text, images, code, or audio
Example Outputs	Fraud risk score, medical diagnosis, spam classification	Chatbot conversations, generated artwork, and AI-written code
Training Data Use	Learns to associate input with a specific label or outcome	Learns patterns to produce realistic new outputs
Security Focus	Protecting the integrity of predictions and preventing model manipulation (e.g., poisoning, evasion)	Preventing misuse of generation, prompt injection, and information leakage
Examples	Credit scoring models, spam filters, and malware detectors	ChatGPT, DALL-E, GitHub Copilot

Table 1: Predictive AI vs. Generative AI

Broader Adversarial Threats Across AI System Types

Adversarial machine learning (AML) risks are not confined to generative AI systems. These systems face targeted threats designed to degrade model performance, mislead outcomes, or extract sensitive information.

EVASION ATTACKS

Here, adversaries craft inputs designed to be misclassified by the model without meaningfully altering the underlying semantics of the input.

- **Mechanism:** The attacker perturbs inputs in ways imperceptible to humans but highly impactful to the model's decision boundary.
- **Examples:**
 - Slight modifications to an image of a stop sign, such as adding stickers or minor distortions, causing an autonomous vehicle classifier to mislabel it as a speed limit sign.
 - Carefully crafted emails are designed to avoid spam detection while maintaining malicious intent.
 - Network traffic modified to bypass intrusion detection systems without altering attack payloads.

Evasion attacks are possible under certain attacker knowledge scenarios:

- **White box:** Attackers know the model parameters and exploit gradients to craft adversarial examples.
- **Black box:** Attackers query the model and use output feedback to approximate decision boundaries, enabling evasion even without internal access.

Evasion attacks reveal a critical flaw, predictive models may behave unpredictably when faced with small, adversarial, crafted inputs — a risk that remains even after traditional model validation and deployment reviews.

POISONING ATTACKS

Poisoning attacks occur during the training phase. Attackers introduce maliciously crafted samples into the training dataset to corrupt the model.

Poisoning attacks are categorized into:

- **Availability Attacks:** Aim to degrade the model's overall accuracy, forcing it to make more errors across general inputs.
- **Integrity Attacks:** Insert specific triggers into the training data, causing the model to behave normally under typical inputs but misclassify or misbehave when the trigger is present.

Examples:

- In a facial recognition system, images containing subtle patterns (e.g., a specific pair of sunglasses) are inserted during training so that the model falsely accepts unauthorized users wearing the trigger.
- Poisoning credit scoring models by injecting fabricated applicant records with anomalous features that skew creditworthiness predictions.

Poisoning is particularly dangerous in environments where training data is sourced externally, from crowdsourced platforms, open repositories, or third-party providers — all of which are increasingly common in large-scale machine learning.

PRIVACY ATTACKS

Privacy attacks exploit the information retained in trained models to infer properties or recover aspects of the training data. Unlike poisoning or evasion attacks, which target the model's operational integrity, privacy attacks compromise the confidentiality of underlying datasets.

Privacy attacks include:

- **Membership Inference Attacks:** The attacker determines whether a specific record was used during model training.
Even models trained on anonymized data can leak membership information, exposing individuals or sensitive data.
- **Property Inference Attacks:** The attacker learns statistical properties of the training data.
Inferring demographic attributes of a population used to train a healthcare model, even if those properties were not part of the model's explicit output.

- **Model Inversion Attacks:** The attacker reconstructs input features or entire records by systematically querying the model and analyzing its outputs.

For instance, reconstructing approximate images of training subjects in a facial recognition model based only on model access.

Privacy attacks present a unique threat in environments where models are shared externally (e.g., via APIs or public services) or deployed in adversarial settings. They show the model can become an information leakage channel even when raw data is protected.

Operational risk management must expand beyond perimeter defenses and software vulnerabilities to include systematic adversarial resilience testing for all deployed AI systems, predictive and generative.

Organizations should treat AI models as active attack surfaces and design security architectures that anticipate targeted manipulation and exploitation at all phases of the model lifecycle.

Taxonomy by Attacker Goals, Capabilities, and Knowledge

Unlike traditional cybersecurity taxonomies, which categorize threats by system vulnerabilities or attack vectors, adversarial machine learning (AML) requires fundamentally different framing. NIST AI 100-2e2025 organizes adversarial attacks across three dimensions: attacker goals, capabilities, and knowledge.

This taxonomy reflects the reality that in AML scenarios, successful attacks often exploit the model's statistical properties, training dependencies, or operational dynamics rather than conventional software flaws. Systematic threat modeling across these three axes improves the ability to anticipate, detect, and mitigate AI-specific risks.

ATTACKER GOALS

Three primary goals adversaries may pursue when attacking AI systems:

- **Availability Attacks:** The adversary seeks to degrade the model's overall performance, causing it to produce unreliable, nonsensical, or unusable outputs.

Example Scenarios:

- Overwhelming a fraud detection model with carefully crafted transactions that evade detection while causing false positives in legitimate activities.
- Causing an autonomous vehicle's object detection system to misinterpret surroundings, forcing operational shutdowns.

- **Integrity Attacks:** The adversary manipulates model behavior to produce specific, incorrect outputs aligned with their objectives.

Example Scenarios:

- Causing a security system to misidentify unauthorized individuals as trusted users.
- Manipulating a predictive maintenance model to suppress early warnings of critical equipment failures.

- **Privacy Attacks:** The adversary aims to extract sensitive information from the model or the data used to train it.

Example Scenarios:

- Determining whether a specific individual's data was used in training a healthcare model.
- Reconstructing private attributes or detailed inputs from model outputs.



ATTACKER CAPABILITIES

An adversary's capabilities define what they can control or influence within the AI system's lifecycle. NIST identifies several critical capability vectors:

- **Training Data Control:** The ability to inject, modify, or influence data used during model training.

Risk Implications:

- Enables poisoning attacks, backdoor insertion, and bias manipulation.
- Particularly relevant in environments relying on external, crowdsourced, or open datasets.

- **Model Control:** The ability to modify model architecture, training code, hyperparameters, or even internal weights after deployment.

Risk Implications:

- Facilitates direct backdoor implantation, functionality sabotage, or embedding of persistent vulnerabilities.
- May occur during supply chain compromises or insider threat scenarios.

- **Test Data Control:** Modifying or crafting inputs at inference time, directly targeting deployed systems.

Risk Implications:

- Enables evasion attacks and adversarial input crafting without needing internal model access.
- Common in scenarios where models are exposed through public APIs, web services, or user-facing applications.

- **Query Access:** The ability to submit inputs to the model and observe outputs without modifying internal systems.

Risk Implications:

- Supports black-box evasion attacks, model extraction efforts, and inference-based privacy attacks.
- Even restricted query access can leak valuable information about model behavior and training data.

Understanding attacker capabilities helps defenders prioritize controls.

ATTACKER KNOWLEDGE

An adversary's knowledge about the model heavily influences the attack techniques available to them:

- **White-Box Attacks:** The adversary has complete visibility into the model's architecture, parameters, training data, and loss functions.

Capabilities:

- Crafting precise adversarial examples using gradient-based methods.
- Designing targeted poisoning attacks by exploiting the importance of known features.
- Extracting sensitive data with high fidelity through model inversion techniques.

White-box attacks typically assume an insider threat or a compromised development environment.

- **Black-box attacks:** The adversary can only observe model outputs given specific inputs and has no access to internal details.

Capabilities:

- Performing black-box evasion attacks using query-based exploration.
- Training substitute (surrogate) models by collecting input-output pairs and approximating decision boundaries.

Despite more limited access, black-box attacks are efficient in real-world deployments where models are exposed through APIs or user interfaces.

- **Gray-Box Attacks:** The adversary has partial information, such as knowledge of the model family, training data distribution, or limited access to internals.

Capabilities:

- Fine-tuning attacks by exploiting assumptions about model behavior.
- Launching informed evasion or extraction efforts with better success rates than pure black-box settings.

OPERATIONAL IMPLICATIONS

This taxonomy allows defenders to model adversarial threats systematically by asking three critical questions:

- 1 What is the adversary trying to achieve? (Goal: Availability, Integrity, Privacy)
- 2 What system components can the adversary access or control? (Capabilities: Training data, model internals, inputs, query access)
- 3 How much does the adversary know about the model? (Knowledge: White-box, Black-box, Gray-box)

Understanding these factors enables organizations to:

- Prioritize hardening of the most exposed system elements.
- Deploy layered defenses aligned to realistic attacker profiles.
- Identify where traditional cybersecurity controls must be augmented to address model-specific threats.

Incorporating attacker goals, capabilities, and knowledge into AI security planning moves defenses from reactive patchwork toward proactive, threat-informed resilience.

Supply Chain Vulnerabilities

Supply chain vulnerabilities in AI systems extend beyond traditional software risks. The AI development and deployment lifecycle introduces unique risks tied to model

sourcing, data provenance, and third-party component integration. Threats can be introduced at any stage, often without immediate detection.

PRE-TRAINED MODELS

Many organizations adopt pre-trained models sourced from external repositories, research labs, or commercial vendors to accelerate AI deployment. However, the use of externally sourced models carries significant security and operational risks:

- **Model Poisoning and Backdoors:** Pre-trained models can be maliciously modified to include backdoors, hidden triggers that cause them to behave incorrectly under specific conditions. These triggers can remain dormant during standard validation but activate under adversarial inputs.
- **Embedded Biases and Agenda-Driven Outputs:** Even absent explicit backdoors, pre-trained models can encode biases, manipulations, or preferences that reflect the agendas of their creators. Recent public investigations have raised concerns around models such as DeepSeek, where outputs aligned with Chinese Communist Party (CCP) narratives. Such biases can subtly influence downstream AI behavior, compromising integrity, objectivity, and mission alignment.

- **Opaque Training Processes:** Models trained without transparency around data sources, labeling practices, or training methodologies introduce unknown risks. Without visibility into the original data or model objectives, it becomes difficult for downstream users to assess the trustworthiness of the models they are integrating.

Adopting pre-trained models without rigorous evaluation creates a critical supply chain risk vector that traditional IT security controls do not mitigate.

FINE-TUNING DATA

Even when organizations perform their own fine-tuning on base models, the datasets used for adaptation introduce another layer of supply chain vulnerability:

- **Data Poisoning:** Attackers can inject malicious or biased examples into fine-tuning datasets, shaping model behavior during adaptation.

- **Indirect Influence:** Poisoned fine-tuning data may not cause immediate misbehavior but can subtly degrade system performance, introduce biases, or prime the model for exploitation under specific conditions.

MODEL ARTIFACTS AND COMPONENTS

Beyond the models and data themselves, the supporting components of the AI system — libraries, toolchains, pipelines, and model serialization formats — also represent critical supply chain risks:

- **Malicious Libraries and Dependencies:** Attackers can compromise software libraries used in AI training and deployment pipelines, embedding vulnerabilities or backdoors that persist across models built using those tools.

- **Tampered Model Files:** Serialized model artifacts can be modified post-training to include malicious payloads, without altering high-level model functionality during superficial testing.
- **Data Processing Pipeline Risks:** Attackers can compromise preprocessing or feature extraction pipelines to systematically bias input handling, degrade model performance, or introduce covert behaviors.

RISK IMPLICATIONS

Supply chain compromises are dangerous because they fundamentally undermine assumptions of system trustworthiness. Once embedded, malicious influences can:

- Evade detection during routine testing.
- Remain dormant until activated by attacker-controlled triggers.

- Manipulate outputs in ways that are difficult to attribute.
- Amplify biases or narratives counter to organizational missions.

MITIGATION APPROACHES

NIST and broader best practices emphasize that securing the AI supply chain requires:

- **Provenance Tracking:** Maintaining complete records of model origin, training sources, fine-tuning datasets, and development environments.
- **Vetting and Testing:** Conducting evaluations of external models for hidden triggers, anomalous behaviors, and bias profiles before adoption.
- **Integrity Verification:** Employing cryptographic signing, secure model distribution practices, and reproducibility analysis to ensure artifacts have not been tampered with.

- **Ongoing Behavioral Monitoring:** Implementing operational controls that continuously monitor deployed model behavior against expected outputs, looking for indicators of compromise or deviation.
- **Limiting Trust:** Minimizing reliance on opaque external models when mission-critical integrity, privacy, or resilience are at stake. Where feasible, retraining or reinforcement training with trusted datasets can help mitigate inherited risks.

Organizations can align their security programs with the realities of modern AI supply chain threats by treating pre-trained models, fine-tuning datasets, and AI components as high-risk assets.

Direct Versus Indirect Prompt Injection

Prompt injection attacks can occur through two distinct mechanisms, each targeting different aspects of how models process inputs.

- **Direct Prompt Injection:** In direct attacks, the adversary submits malicious input explicitly crafted to manipulate model behavior. These prompts are designed to override, subvert, or confuse system instructions embedded within an AI system.

Examples:

- Crafting a jailbreak prompt that disables content moderation controls.
- Inserting commands that cause the model to reveal internal configurations or sensitive data.

Direct prompt injection exploits the system's trust in user-provided input, targeting prompt structure and context alignment vulnerabilities.

- **Indirect Prompt Injection:** Indirect attacks occur when an adversary manipulates external content sources that the model ingests during normal operation. In this case, the malicious instructions are embedded within documents, websites, emails, or other retrieved data, rather than being directly input by the user.

Examples:

- Embedding adversarial instructions into a web page that an agent retrieves and processes.
- Inserting manipulated text into a third-party document that the AI summarizes or analyzes.

Indirect prompt injection exploits the model's handling of external or retrieved content, targeting the trust boundary between system inputs and model instructions.

DEFENSIVE CONSIDERATIONS

Effective defense requires addressing both attack pathways:

- **Direct Injection Defenses:** Organizations must ensure that user inputs cannot manipulate system instructions or alter model behavior beyond intended boundaries.
- **Indirect Injection Defenses:** Systems that retrieve or integrate external content must apply secure retrieval practices, including input sanitization at ingestion, content validation before processing, and strict isolation between retrieved data and operational prompts.



Securing LLM Agents and Tool Use

Agents may browse the Internet, interact with APIs, manage files, execute code, or control external systems. These capabilities enable powerful automation but also significantly expand the attack surface.

There are several specific risks associated with agents:

- **Unintended Command Execution:** Agents that interpret outputs as executable actions are vulnerable to adversarial prompts or injected content that triggers unauthorized or harmful commands. Attackers can manipulate workflows to achieve malicious objectives if model outputs are trusted without strict validation.
- **Data Exfiltration and Information Leakage:** Agents empowered to interact with external systems can be manipulated to retrieve, leak, or exfiltrate sensitive information. This risk is amplified when agents access internal systems or privileged data repositories.

- **Privilege Escalation and Lateral Movement:** Attackers can design inputs that cause agents to invoke privileged APIs or access restricted resources, effectively escalating privileges within the system or pivoting toward broader enterprise compromise.

Without strong safeguards, attackers can weaponize agentic LLM systems, moving beyond prompt injections to achieve active exploitation.

Challenges in Evaluating and Sustaining Defenses

Defending AI systems against adversarial attacks requires more than implementing static controls or relying on isolated evaluations. Many defenses that appear effective under initial testing fail when subjected to advanced adversaries.

Because the threat landscape is dynamic, organizations cannot treat AI security evaluations as one-time exercises. Defenses must be viewed as iterative, requiring:

- Continual reassessment against evolving threat models.
- Rigorous red-teaming and stress-testing under both white-box and black-box adversary simulations.

- Periodic model hardening based on updated attack techniques and environmental changes.
- Operational monitoring to detect shifts in model behavior that may indicate exploitation or emerging vulnerabilities.

Building AI systems resilient to adversarial threats requires sustained investment in testing that informs system defense updates.

Trade-Offs Between Robustness, Accuracy, and Privacy

Building resilient AI systems requires more than applying defensive techniques; it demands a clear understanding of the operational trade-offs each defense introduces. Organizations cannot assume that security improvements

come without cost. Each defensive approach must be evaluated in the context of mission priorities, acceptable risk thresholds, and operational requirements.

SECURITY DESIGN IMPLICATIONS

Security designs must balance these trade-offs thoughtfully. Protecting AI systems requires maximizing defenses and aligning those defenses with the intended operational environment. Decisions must consider:

- How much accuracy loss is acceptable in favor of greater robustness?
- How much computational overhead can be absorbed in production environments?
- How critical is data confidentiality relative to model performance?

There is no universal solution. Trade-off management must be an explicit part of AI system design, acquisition, deployment, and governance processes.

Selecting appropriate defensive strategies requires a deliberate, risk-informed approach. This ensures that security controls strengthen mission success rather than inadvertently degrading system effectiveness or resilience.

Multimodal System Considerations

These systems combine diverse data types to enhance capabilities and introduce new and complex adversarial attack surfaces. Areas of concern include:

- **Exploitation of Vulnerabilities Within a Single Modality:** Attackers can target the weakest modality within a multimodal system to compromise overall system integrity.

Examples:

- Subtly manipulating an input image so that it corrupts the model's interpretation or downstream decision-making.
- Embedding adversarial perturbations in audio data that, when transcribed or analyzed, inject malicious content into text-based outputs.

Because multimodal systems must aggregate and process data from multiple sources, weaknesses in one pathway can cascade across others, undermining the entire system.

- **Cross-Domain Adversarial Attacks:** Attackers can design cross-modal attack chains where a malicious input in one domain affects processing and output in another.

Examples:

- Poisoning an image captioning model by embedding adversarial triggers in the image that manipulate the generated textual description.
- Crafting metadata within structured data that alters outputs without blatant manipulation at the input layer, once processed alongside textual prompts.

SECURITY DESIGN IMPLICATIONS

Effective AI security for multimodal systems requires:

- **End-to-End Threat Modeling:** Organizations must perform threat modeling that spans all input modalities, identifying how vulnerabilities in one domain can propagate through the system.
- **Modality-Specific Hardening:** Each data type — text, image, audio, video, structured data — must be individually hardened against adversarial manipulation before integration.
- **“Fusion Point Security”:** The points where multimodal data is aggregated or fused must be treated as critical trust boundaries, protected with input validation, anomaly detection, and contextual consistency checks.
- **Continuous Cross-Modality Testing:** Security testing must include adversarial scenarios that combine inputs across modalities, reflecting the fundamental ways attackers may exploit complex system interactions.

OPERATIONAL RISK

Multimodal systems increase complexity and operational risk. Failing to secure all supported data pathways creates opportunities for attackers to compromise system behavior through indirect or less scrutinized channels. A single compromised modality — if unchecked — can create cascading security failures across otherwise secure components.

Building resilient multimodal AI systems demands integrated security strategies that recognize the interdependencies between different data types and the adversarial opportunities those interdependencies create.

Expanding Privacy Attack Awareness

Advanced adversaries can exploit the statistical properties of trained models to extract sensitive information, even when conventional security controls are in place.

These privacy attacks do not rely on traditional IT system vulnerabilities. Instead, they target how machine learning models memorize, generalize, and expose information through their learned representations and outputs. Key areas that need to be addressed include:

- **Membership Inference Attacks:** In membership inference, the adversary aims to determine whether a specific data record was used during model training.
- **Property Inference Attacks:** Property inference attacks allow an adversary to learn aggregate statistical properties of the training dataset, even if those properties were not explicitly intended for exposure.
- **Model Inversion Attacks:** Model inversion techniques attempt to reconstruct features or entire data records based solely on model outputs and prediction behaviors.

DEFENSE CONSIDERATIONS

Protecting against privacy attacks requires architectural and operational controls that go beyond standard access management:

- **Regular Privacy Testing:** Privacy risk assessments must be integrated into model evaluation workflows, including formal testing for membership inference susceptibility, property leakage, and inversion vulnerabilities.
- **Application of Privacy-Enhancing Technologies:** In sensitive environments, differential privacy must be incorporated during training and inference phases. Differential privacy introduces controlled noise to prevent attackers from distinguishing individual records based on model outputs while maintaining functional performance.

- **Post-Deployment Monitoring:** Even after deployment, models should be monitored for anomalous output behaviors that may indicate active privacy probing by adversaries.

STRATEGIC IMPLICATIONS

Privacy attacks represent a distinct and evolving threat class in AI security—one that traditional defenses cannot fully address. As machine learning models increasingly operate on sensitive datasets, organizations must treat model privacy protection as a key component of AI risk management, not an

optional enhancement. Building trust in AI systems requires continuous attention to how information may be leaked, not only through infrastructure compromise but also through the model's behavior.

Sustaining Secure and Trustworthy AI Operations

Securing AI systems requires addressing a threat landscape that is broader, deeper, and more dynamic than what traditional cybersecurity frameworks account for. While the OWASP Top 10 for LLM Applications provides a strong foundation for mitigating many risks specific to large language models, it captures only part of the operational reality agencies and organizations now face.

As outlined in NIST AI 100-2e2025, adversarial threats extend across other AI system types — predictive and generative — and exploit vulnerabilities in model design across training pipelines, supply chains, deployed agents, and multimodal environments. These threats leverage model properties, not just external system weaknesses, requiring a shift in how organizations define and defend their AI attack surfaces.

Organizations must treat AI models as dynamic, high-risk assets embedded in increasingly complex operational ecosystems. Future AI security programs must evolve to systematically account for:

- Adversarial manipulation targeting both predictions and generations.
- Supply chain risks arising from opaque model sourcing and fine-tuning processes.
- Cross-modal exploitation strategies that bridge data domains.
- Privacy breaches that emerge from model behaviors, not infrastructure gaps.
- The need for continual reassessment, not static defense postures.

Risk management strategies must embed the disciplines outlined in this article and the supporting source materials as core business/mission requirements, not as reactive post-deployment fixes. Adversarial threats to AI systems will continue to evolve alongside attacker capabilities. Building and sustaining resilience is not a one-time effort but an iterative, ongoing operational requirement that must be planned, budgeted, and continuously refined as part of the AI system lifecycle.

Final Thoughts

Securing AI goes beyond just defending against cyber threats; it involves ensuring that AI remains trustworthy, reliable, and ethical. It should enhance decision-making without introducing new vulnerabilities. The next phase of AI security will not be determined by whether organizations acknowledge the risks but by how effectively they respond to them. If organizations treat AI security as an afterthought, they could face unpredictable failures, loss of trust, and severe mission impacts. Conversely, those who integrate security into every stage of AI development, deployment, and monitoring will be better able to adapt and respond to changes in the threat landscape over time.

The challenge is clear: incorporate AI security into all your processes and verify the improvements. If AI is to be a fundamental component of enterprise technology, its security must be given the same level of rigor as any other mission-critical system.



Glossary of Terms

Adversarial Attacks – Techniques used to manipulate AI models by introducing misleading inputs, often to deceive or exploit weaknesses in decision-making processes.

AI Autonomy – The degree to which an AI system can make decisions and take actions independently, potentially leading to excessive agency if not properly constrained.

AI Governance – The policies, frameworks, and practices that guide the responsible development, deployment, and oversight of artificial intelligence systems.

AI Hallucination – When an AI model generates outputs that appear credible but are factually incorrect, often due to incomplete or biased training data.

AI Security – The measures and practices designed to protect AI systems from attacks, manipulation, and vulnerabilities that could compromise their integrity, confidentiality, and availability.

Artificial Intelligence (AI) – A field of computer science focused on creating systems that can perform tasks typically requiring human intelligence, such as language processing, decision-making, and pattern recognition.

Data and Model Poisoning – An attack method in which adversaries manipulate training data or alter model weights to introduce vulnerabilities, biases, or harmful behaviors.

Data Leakage – The unintentional exposure of sensitive or proprietary information through AI-generated outputs, often caused by poor input validation or system prompt leakage.

Denial-of-Service (DoS) Attack – A type of cyber attack where adversaries overwhelm an AI system with excessive resource demands, leading to degraded performance or system failure.

Embedding Weaknesses – Vulnerabilities arising from the improper handling of vector embeddings can expose patterns in data, allow for adversarial manipulation, or lead to security breaches.

Explainability in AI – The ability to interpret and understand how an AI system makes decisions, essential for ensuring trust, transparency, and security.

Fine-Tuning – The process of adjusting a pre-trained AI model using additional training data to optimize performance for a specific task, which can introduce security risks if not adequately controlled.

Generative AI – A subset of AI that creates new content, such as text, images, or code, based on patterns learned from training data.

Human Oversight – Incorporating human review and intervention in AI decision-making processes to prevent errors, biases, or unintended consequences.

Inaccurate or Misleading Outputs – AI-generated responses that lack factual accuracy or misrepresent information, potentially leading to incorrect decision-making.

Least Privilege Principle – A security practice that restricts AI systems' access and permissions to the minimum necessary for their function, reducing the risk of unauthorized actions.

Large Language Model (LLM) – An AI model trained on vast amounts of text data to generate human-like responses, used in chatbots and automated content generation applications.

Model Drift – A gradual degradation in an AI model's accuracy over time due to changes in data patterns, requiring regular monitoring and updates.

Model Robustness – The ability of an AI system to maintain consistent performance and resist manipulation or exploitation by adversarial inputs.

NIST (National Institute of Standards and Technology) – A U.S. federal agency that develops technology, metrics, and standards to promote innovation and industrial competitiveness. In cybersecurity and AI, NIST provides authoritative frameworks, guidelines, and best practices for securing systems and managing risk.

NIST SP 800-160 – A cybersecurity framework that guides on developing cyber-resilient systems, including AI technologies.

NIST SP 800-161 – A publication focused on cybersecurity supply chain risk management (C-SCRM) to ensure secure AI components and software dependencies.

NIST SP 800-218 – A framework detailing best practices for secure software development, emphasizing AI model security and integrity.

OWASP (Open Worldwide Application Security Project)

– A nonprofit foundation that works to improve software security through community-led open-source projects, education, and resources. OWASP is known for publishing widely used security guidance, including the OWASP Top 10 lists for various technologies such as web applications and large language model (LLM) applications.

OWASP Top 10 for LLM Applications – A list of the most critical security risks facing large language model applications, providing guidelines for mitigating AI vulnerabilities.

Phishing via AI – Using AI-generated messages to deceive individuals into revealing sensitive information, often through social engineering tactics.

Prompt Injection – An attack technique where adversaries manipulate an AI's prompt inputs to alter its behavior, bypass security measures, or extract sensitive information.

Rate Limiting – A security control that restricts the number of API or system requests within a given timeframe to prevent overuse and mitigate unbounded consumption risks.

Red Teaming AI – A security practice where adversarial testing is conducted to identify weaknesses in AI systems before malicious actors can exploit them.

Resource Consumption in AI – The computational power, memory, and storage required for AI operations must be managed to prevent system slowdowns, cost overruns, or service disruptions.

Secure Software Development Framework (SSDF) – A set of best practices for integrating security into software and AI model development processes.

Software Bill of Materials (SBOM) – A detailed inventory of software components and dependencies used in AI systems, ensuring transparency and security in the supply chain.

Supply Chain Risks in AI – The security challenges associated with third-party AI models, software libraries, and data sources that could introduce vulnerabilities.

System Prompt Leakage – The unintended exposure of internal AI system instructions, allowing adversaries to manipulate model behavior or extract sensitive details.

Threat Modeling for AI – Identifying potential attack vectors, vulnerabilities, and mitigation strategies to secure AI systems.

Unbounded Consumption – A security risk where AI systems excessively use resources without limits, leading to performance degradation, increased costs, or denial-of-service conditions.

Vector Search Systems – AI-driven databases that enable efficient retrieval of embeddings, which, if not secured, can be exploited for unauthorized data extraction.

Vulnerability Management in AI – The continuous process of identifying, assessing, and mitigating security weaknesses in AI models and associated systems.

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



Dr. Darren Death

Dr. Darren Death serves as the Chief Information Security Officer, Chief Privacy Officer, and Deputy Chief Artificial Intelligence Officer at the Export-Import Bank of the United States. In this role, he leads enterprise efforts to secure information systems, protect privacy, and govern the responsible use of AI in alignment with federal mandates and mission-driven priorities.

A seasoned technology leader with over 25 years of experience designing and securing enterprise IT systems, Dr. Death has delivered complex, high-impact technology solutions across both public and private sectors. Prior to joining EXIM, he served as the Chief Information Security Officer for ASRC Federal, where he directed the cybersecurity program across a broad portfolio of federal contracts and subsidiary companies.



CyberRisk Alliance

CyberRisk Alliance provides business intelligence that helps the cybersecurity ecosystem connect, share knowledge, accelerate careers, and make smarter and faster decisions. Through its trusted information brands, network of experts, and innovative events it provides cybersecurity professionals with actionable insights and act as a powerful extension of cybersecurity marketing teams. CyberRisk Alliance brands include SC Media, the Official Cybersecurity Summits, TechExpo Top Secret, InfoSec World, Identiverse, Cybersecurity Collaboration Forum, Cybersecurity Collaborative, Security Weekly, ChannelE2E, MSSP Alert, and LaunchTech Communications.



ICIT

www.icitech.org