CS-112 · AI Security

Model Poisoning

Corrupting an AI system through its training data, its supply chain, or its ongoing learning.

Executive Summary

Model poisoning is the deliberate corruption of an AI system so that it learns the wrong things or hides a malicious behavior. Attackers can taint the data a model trains on, tamper with a model or component pulled from an outside source, or feed harmful examples into a system that keeps learning from user activity. The result can be quiet errors, biased decisions, or a hidden trigger that makes the model misbehave on command.

What It Is

Model poisoning is an integrity attack on the machine learning lifecycle rather than on a running server. The most common form is data poisoning, where an attacker inserts crafted or mislabeled examples into a training set so the finished model absorbs a flaw. A related form is the backdoor, where the model behaves normally almost all the time but produces an attacker-chosen output when a specific hidden trigger appears in the input. Poisoning can also enter through the AI supply chain when an organization downloads a pretrained model, an adapter, or a dataset from an outside source that has been tampered with. Systems that keep learning from live user interactions face a further risk, because a coordinated stream of malicious input can gradually bend the model over time.

Why It Matters

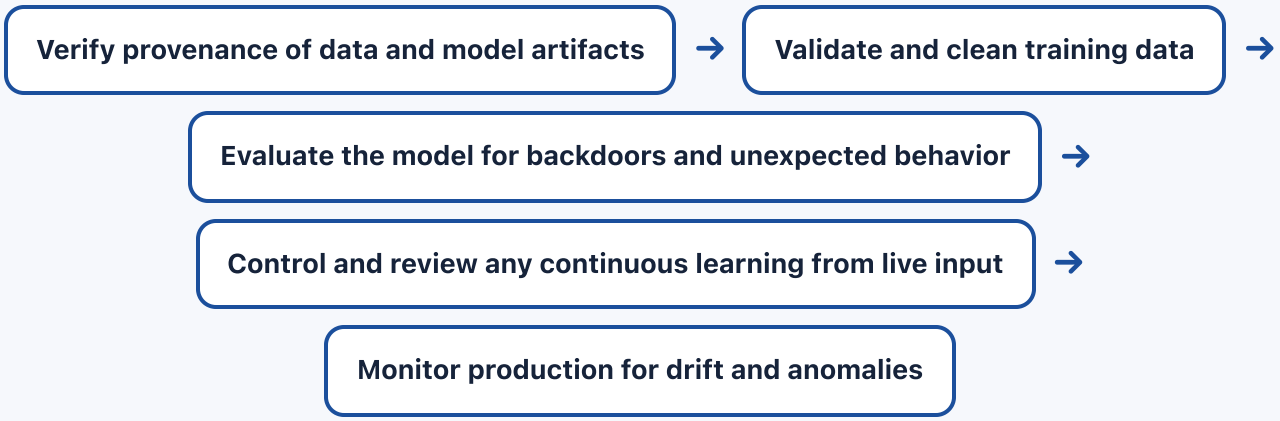
A poisoned model can fail in ways that are hard to see and hard to trace. Unlike a crash, a subtle bias or a rare backdoor may pass every routine test and only surface in production, where it can distort fraud checks, content moderation, credit decisions, or safety controls. Because so many teams build on outside models and public datasets, one tainted upstream artifact can spread

into many downstream products. For governance, risk, and compliance professionals, poisoning is a supply chain and data integrity problem that touches vendor management, model documentation, and the ability to trust an automated decision, all of which regulators and auditors increasingly expect organizations to demonstrate.

How It Works

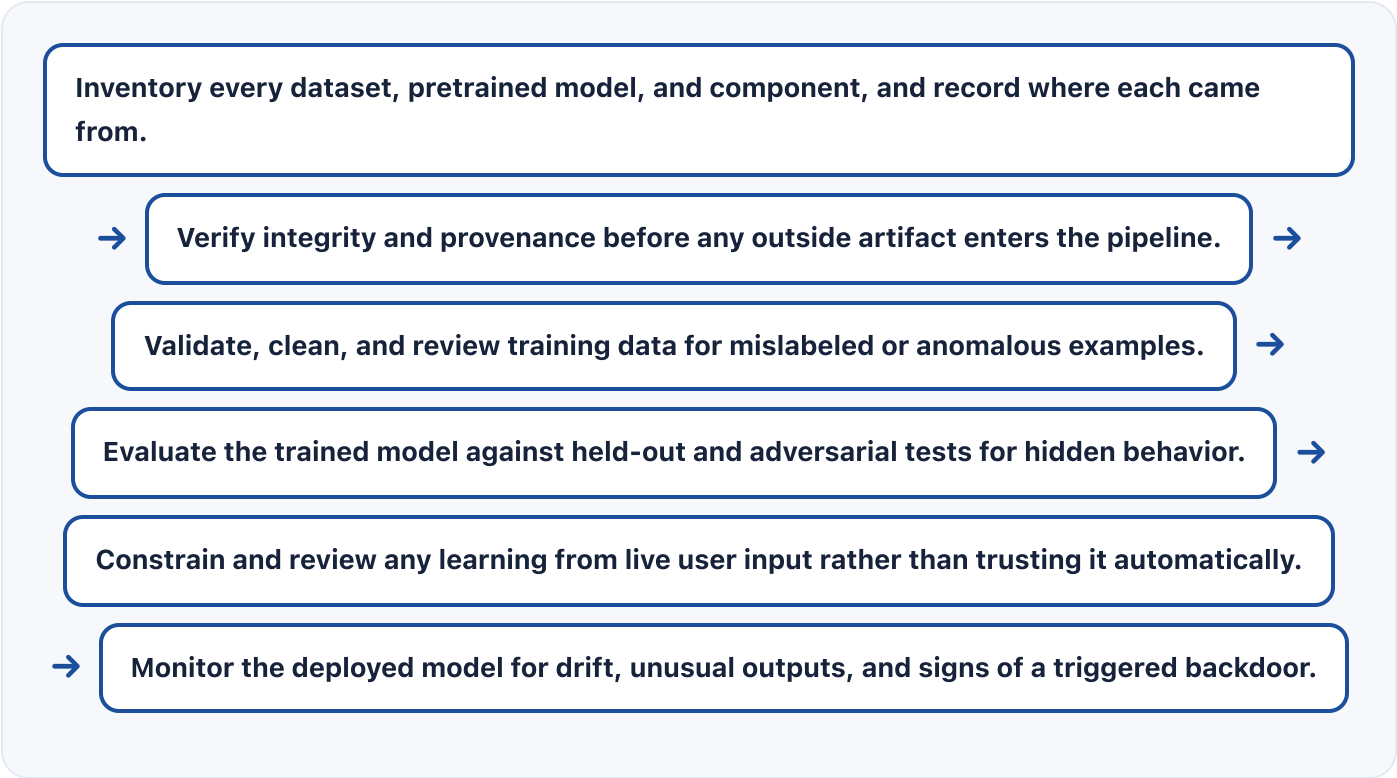
The attacker's goal is to influence what the model learns or to embed behavior that survives into deployment. In a data poisoning attack, tainted examples are mixed into the training set, often in small enough amounts to avoid notice, so the model generalizes a wrong pattern. In a backdoor attack, the poisoned examples teach the model to react to a specific trigger, giving the attacker a covert switch. In a supply chain attack, the compromise is baked into a model or component before the victim ever trains anything, so the flaw arrives fully formed. Defense centers on knowing where every dataset and model came from, verifying integrity before use, validating and cleaning training data, evaluating models for unexpected behavior, and watching production for drift or anomalies. This reference explains the risk and its mitigations and does not provide instructions for building a poisoning attack.

Architecture Diagram



Poisoning enters through data, the supply chain, or continuous learning; provenance, validation, and monitoring reduce the risk at each stage.

Visual Workflow



Common Attacks

- Data poisoning that inserts tainted or mislabeled examples into a training set
- Backdoor attacks that make the model misbehave only when a hidden trigger appears
- Supply chain compromise of a downloaded pretrained model, adapter, or dataset
- Feedback abuse that bends a continuously learning system through coordinated malicious input
- Label manipulation that quietly skews classification toward an attacker's goal

Common Mistakes

- Trusting public datasets and third-party models without verifying their source or integrity
- Retraining on live user data with no filtering or review
- Testing only for accuracy and never for hidden or triggered behavior
- Keeping no record of what data and components went into a given model version
- Assuming poisoning is theoretical and leaving no monitoring to detect it

Best Practices

- Track provenance for every dataset and model artifact across the lifecycle
- Verify integrity of outside models and data before use, and prefer trusted sources
- Validate and clean training data, and review anomalous or unexpected examples
- Evaluate models with adversarial and held-out tests, not accuracy alone
- Gate and review continuous learning so live input cannot silently retrain the model
- Monitor production for drift and unusual behavior, and keep versioned records to roll back

Quick Checklist

- [x] Provenance recorded for all datasets and model components
- [x] Integrity of outside models and data verified before use
- [x] Training data validated and screened for anomalies
- [x] Model evaluated for backdoors and unexpected triggered behavior
- [x] Continuous learning from live input is controlled and reviewed
- [x] Production monitoring and versioned rollback are in place

Recommended Tools

Data validation and quality tooling
Screens training data for anomalies, mislabeling, and outliers

Model and data provenance tracking
Records the source and lineage of every artifact in the pipeline

Model evaluation and red-team testing
Probes trained models for hidden or triggered behavior

Drift and anomaly monitoring
Watches deployed models for behavior that shifts over time

Industry Standards

NIST AI Risk Management Framework (AI RMF, AI 100-1)

MITRE ATLAS
Catalogs poisoning and supply chain

Frames data integrity and model risk under Govern, Map, Measure, and Manage

techniques against machine learning systems

OWASP Top 10 for LLM Applications
Highlights training data poisoning and supply chain risk for model applications

Career Relevance

Model poisoning is central to AI security engineers who defend the machine learning pipeline and to AI risk managers who must account for data integrity and supply chain exposure. AI governance analysts document dataset provenance and model lineage as part of trustworthy AI programs, and GRC analysts fold poisoning into vendor risk reviews and model audits. Understanding it demonstrates that a candidate can protect not just the running application but the process that produced the model, a skill in demand across the roles AI-Governance-Jobs.com serves.

Interview Questions

- What is the difference between data poisoning and a model backdoor?
- How can poisoning enter through the AI supply chain rather than through training?
- Why is testing for accuracy alone insufficient to catch a poisoned model?
- What controls reduce the risk when a system keeps learning from live user input?
- How does data and model provenance help you respond after poisoning is suspected?

Related Certifications

OWASP resources for LLM and ML security

ISACA or ISC2 AI security offerings (as available)

CompTIA Security+ (for the security foundations)

Further Reading

- NIST AI Risk Management Framework
- MITRE ATLAS

- OWASP Top 10 for LLM Applications

Key Takeaways

- Model poisoning corrupts what a model learns through data, the supply chain, or continuous learning.
- Backdoors make a model misbehave only on a hidden trigger, so they can pass routine tests.
- Outside models and public datasets can carry poisoning into many downstream products.
- Provenance, data validation, adversarial evaluation, and monitoring are the core defenses.
- It is a data integrity and supply chain risk that AI governance and GRC roles must manage.

FAQ

How is model poisoning different from prompt injection?

How much poisoned data does it take to matter?

We only use a pretrained model from a vendor, are we safe?

Related Careers

RELATED ROLES

Ai Security Engineer | Ai Risk Manager | Ai Governance Analyst | Grc Analyst

RELATED CERTIFICATIONS

OWASP resources for LLM and ML security

ISACA or ISC2 AI security offerings (as available)

CompTIA Security+ (for the security foundations)

CURRENT OPENINGS

Live roles are on the job board.

Browse all jobs

GUIDES & SALARY

Career guides | Role roadmaps | Salary data | Career tools

SUGGESTED LEARNING PATH

1. Ground the basics with CS-001 Cybersecurity
2. Study this sheet: **Model Poisoning**
3. Go deeper: Prompt Injection
4. Go deeper: OWASP Top 10 for LLM Applications
5. Validate it: work toward **OWASP resources for LLM and ML security**
6. Find the role: browse current openings

RELATED SHEETS

CS-111 Prompt Injection | CS-116 OWASP Top 10 for LLM Applications | CS-115 Secure AI Adoption | CS-001 Cybersecurity | CS-104 NIST Cybersecurity Framework (CSF)