What are malicious AI packages and models?

Malicious AI packages and models are poisoned or impersonating components in the AI software supply chain: fake or trojanised packages that pose as AI SDKs and tools, and machine-learning models on public hubs that run code the moment you load them. Two shifts make this worse. Attackers now use AI agents to generate and publish malicious packages at scale, and common model formats can execute code on load. The defenses are to verify what you install and load, prefer safe model formats, and scan both code and model files, including metadata, before use.

HOW IT WORKS

01 AI agents mass-producing malicious packages

The newest shift is attackers using AI agents to generate and publish malicious packages at scale. In a 2026 RubyGems campaign, thousands of malicious packages were uploaded, and investigators linked the activity to AI agents driving the work at machine speed, with some packages referencing an AI provider's tooling in their names.

The agent origin left fingerprints: large numbers of near-identical packages, generated or disposable names with timestamp suffixes, repeated model-signature words, and placeholder author names. The payloads stole registry API tokens and scraped and exfiltrated data. Notably, some hid malicious content in package metadata, such as descriptions, author fields, and documentation-build directives, rather than in obvious code, to slip past scanners. The scale is the weapon: an agent swarm can flood a registry faster than defenders triage it.

02 Poisoned models on public hubs

A model is code as well as weights. Common serialization formats such as Python pickle can execute arbitrary code when a model is loaded, so a malicious model on a public hub runs on your machine the instant you load it, before you ever run inference.

Researchers have repeatedly found malicious models on public hubs that abuse this. The safer path is a format that does not execute code, such

HOW TO DEFEND

- Verify exact names and provenance before installing a package or downloading a model, and pin and lock dependencies so a lookalike cannot be added silently.
- Prefer safe model formats such as safetensors, and never load an untrusted pickle.
- Scan packages and model files, including metadata, in CI, not just the obvious source code.
- Restrict what documentation and build steps can run, since some payloads hide in doc-generation and post-install hooks.
- Protect registry and hub tokens so one stolen credential cannot cascade into more malicious uploads.

SOURCES

[1] OWASP Top 10 for LLM Applications (Supply Chain)

[2] Adversarial Threat Landscape for AI Systems

[3] Hub Security and safetensors

as safetensors, plus scanning models before you load them and treating an unknown model like any untrusted download.

03 Impersonating AI vendors

Attackers also register lookalike packages that pose as an official AI provider's library, so a mistyped install, or one an assistant suggests, pulls malware instead of the real SDK. This is typosquatting and slopsquatting aimed at the fast-moving AI dependency list, where teams add new AI libraries quickly and often trust the first result.

Find out whether a poisoned AI package or model could reach your systems.

securelayer7.net/learn/ai-security/malicious-ai-packages-and-models

Open online