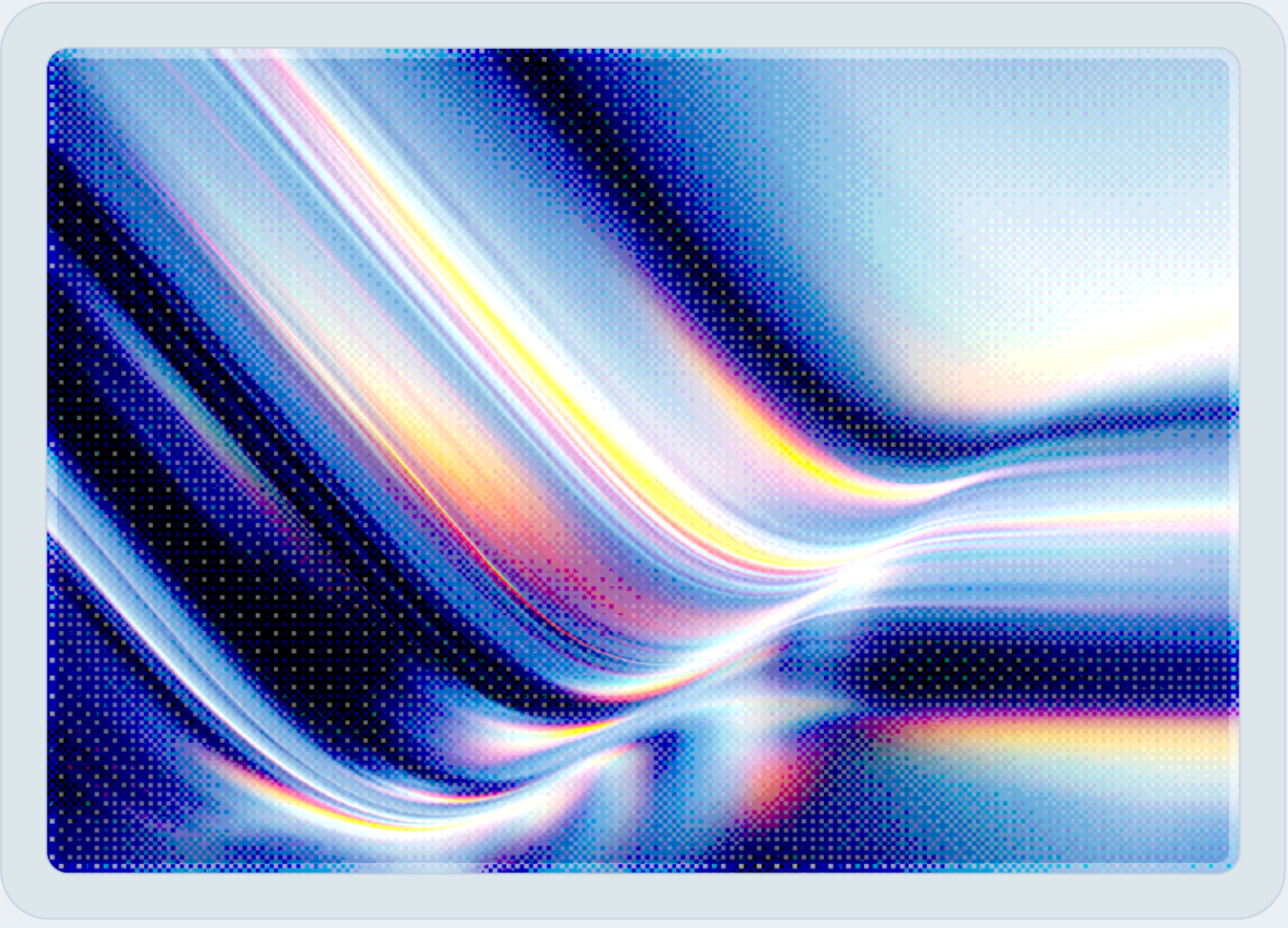
CSAI Foundation | Cloud Security Alliance

Poisoned Pipelines: Malicious AI Model and Skill Repositories

How Open AI Marketplaces — from Hugging Face to ClawHub — Are Becoming Malware Distribution Infrastructure

2026-05-10

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Open AI model and agent skill repositories have become viable malware distribution channels, with attackers exploiting the implicit trust practitioners place in community-curated platforms.
- Python's pickle serialization format, the dominant storage mechanism for ML model weights, allows arbitrary code execution at load time — a vulnerability class that has been repeatedly exploited in the wild on Hugging Face since at least March 2024 [4].
- In February 2026, the Koi Security research team discovered 341 malicious skills in the ClawHub agent skill registry — a supply chain compromise that distributed Atomic Stealer malware targeting cryptocurrency tools and trading infrastructure [1].
- PickleScan, the primary open-source tool for detecting malicious pickle payloads, was found to carry three zero-day bypass vulnerabilities in December 2025 (CVE-2025-10155, CVE-2025-10156, CVE-2025-10157), each rated CVSS 4.0 9.3 — meaning organizations relying solely on PickleScan have had months of exposure they may not have detected [2].
- The safetensors format provides a safer serialization alternative that has passed independent security audit with no critical arbitrary code execution vulnerabilities, but adoption remains incomplete across the practitioner community.
- Organizations must treat AI model and skill downloads with the same supply chain scrutiny applied to third-party code packages: verified provenance, hash integrity, sandboxed loading, and preference for formats that do not permit code execution.

Background

The growth of open AI repositories over the past three years has changed how organizations integrate machine learning into production systems. Platforms such as Hugging Face have become the de facto distribution layer for pretrained model weights, fine-tuned variants, embedding models, and tokenizers. As of 2026, Hugging Face hosts hundreds of thousands of model repositories, with practitioners routinely downloading weights as a first step in any new AI deployment. The platform's model card

system and community review processes create a credible appearance of curation, but the fundamental trust model depends on individual contributors acting in good faith — a proposition that threat actors have learned to exploit.

The emergence of agentic AI systems has opened a parallel distribution surface. Platforms such as ClawHub serve as registries for the modular skills and tool-call definitions that autonomous AI agents use to interact with external services. In architecture, ClawHub resembles an npm registry or PyPI index for AI agents: developers publish packaged capabilities — search tools, API integrations, specialized reasoning routines — and orchestration frameworks pull them at runtime. The ClawHavoc campaign, documented in February 2026, demonstrated that this newer category of AI component store carries the same supply chain exposure as software package registries, with an additional wrinkle: poisoned agent skills can instruct an AI system to perform malicious actions directly, rather than merely executing a payload in the host operating system [1].

Both platforms share a structural characteristic that security practitioners should recognize as foundational to the risk: they function as what this analysis calls "trustless model marketplaces" — not in the cryptographic sense of blockchain trustlessness, but in the operational sense that content is accepted and distributed without binding verification of author identity, code review by the platform, or runtime attestation of behavioral safety. The absence of mandatory signing, comprehensive behavioral analysis, or enforced format restrictions means that the repository's reputation becomes the primary trust signal, and that reputation can be abused.

The attack surface is not theoretical. Security researchers at JFrog documented approximately 100 malicious AI and ML model files on Hugging Face in March 2024, using Python's pickle serialization format to embed reverse shell payloads that executed silently when a model was loaded into memory. One documented case involved a model file from the account "baller423" that connected to an external IP address upon loading, giving attackers interactive shell access to the data scientist's workstation [4]. NullBulge, a threat actor that emerged in April 2024 and later conducted a breach of Disney's internal Slack infrastructure, distributed early-stage malware through Hugging Face by packaging malicious payloads inside game modification files presented as community content [5]. These incidents share a pattern: the platform's openness, which is its value proposition, is the same property that makes it suitable attack infrastructure.

Security Analysis

The Pickle Problem

The technical core of the model repository attack surface is Python's `pickle` serialization module. Pickle is the default storage format for PyTorch model weights — the `.pth` and `.bin` files that practitioners download and load daily — because it handles arbitrary Python objects efficiently without requiring a schema. The security cost of this flexibility is severe: pickle deserialization will execute any Python code embedded in a file at load time, before any inspection of the loaded object is possible. An attacker who can distribute a pickle file can therefore achieve arbitrary code execution on any system that loads it, with no further exploitation required [6].

The scale of pickle adoption in the ecosystem makes this a systemic concern rather than an edge case. Roughly 95 percent of the malicious models JFrog identified in March 2024 used PyTorch's pickle-based format [4]. The attack technique requires only that an attacker embed malicious logic in a model's `__reduce__` method — a standard Python mechanism for controlling deserialization behavior that pickle's security documentation explicitly warns against using with untrusted data, yet which the ML ecosystem has normalized as part of how models are stored and loaded.

PickleScan, an open-source tool integrated into Hugging Face's scanning pipeline, was developed to detect dangerous pickle payloads by checking for the presence of functions on a security blocklist. The tool represents a genuine defensive contribution, but its blocklist approach has proven brittle under adversarial pressure. JFrog's December 2025 disclosure revealed three zero-day bypasses, each rated CVSS 4.0 9.3: CVE-2025-10155 allows evasion by renaming files with alternative extensions (`.bin` , `.pt`) that PickleScan did not scan by default; CVE-2025-10156 exploits CRC error handling in ZIP archives to conceal malicious content; and CVE-2025-10157 uses subclassed module paths to invoke dangerous functions without matching the blocklist patterns [2]. Organizations relying on PickleScan as a primary defense should treat their posture as unvalidated until they have confirmed they are running a patched version and have audited model inventories loaded during the vulnerability window.

Agent Skill Registries: The Expanding Surface

The ClawHavoc campaign extended this supply chain risk from static model weights to dynamic agent capabilities. ClawHub functions as the official skill registry for the OpenClaw agent framework, allowing agent orchestration systems to discover and load capabilities at runtime in a pattern analogous to

package managers in software development. The Koi Security team's analysis in February 2026 identified 341 malicious skills among 2,857 total entries in the registry — roughly 12 percent of all available skills at the time of discovery [1].

The attack payload in ClawHavoc was Atomic Stealer, a credential-harvesting malware family that targets cryptocurrency wallets, browser-stored passwords, and session tokens. The category selection was not random: skills targeting cryptocurrency trading bots and Polymarket integrations were among the most heavily poisoned, indicating that the attacker prioritized categories where victims would have high-value financial credentials accessible through the agent's runtime environment [1].

The structural risk here extends beyond any specific campaign. Agent skills, unlike static model weights, are designed to execute code and interact with external services as part of their normal function. This means that a malicious skill does not need to exploit a deserialization vulnerability — it simply needs to be a functional skill that also performs malicious actions. The line between "a skill that calls an API" and "a skill that calls an API and exfiltrates credentials" is invisible to any static analysis tool that does not understand the full intent graph of what the agent is being asked to do. Production-grade platform-level behavioral sandboxing for agent skills does not yet exist at major public registries, leaving the intent-graph problem unaddressed at the infrastructure layer.

Trust Chains and Namespace Attacks

Beyond direct payload injection, AI model repositories face a category of attack that exploits the trust implied by naming conventions. Namespace reuse attacks — documented by Palo Alto Networks Unit 42 — involve registering model names that closely resemble widely-used legitimate models, or reactivating namespaces vacated by legitimate maintainers, and publishing poisoned variants under those familiar identities [7]. The pattern mirrors typosquatting and dependency confusion attacks in software package ecosystems, and it is effective for the same reason: platform defaults do not require practitioners to verify cryptographic integrity or confirm that the download path matches a canonical source, making it straightforward to download the first plausible match by name without additional validation.

The trust problem is compounded by the social proof mechanisms these platforms employ. Star counts, download statistics, and community discussion threads create the appearance of community validation. Attackers who invest in creating plausible community activity around a malicious model — by accumulating downloads through bot activity or by forking legitimate popular models with poisoned weights — can manufacture credibility that is difficult for practitioners to distinguish from the real thing using default platform signals alone.

The Safetensors Opportunity

The safetensors format, developed by Hugging Face as an alternative to pickle for storing model weights, addresses the deserialization code-execution risk by design. Safetensors uses a simple header-plus-raw-buffer format that contains no executable code path; loading a safetensors file does not trigger arbitrary code execution through the deserialization process itself. A 2023 joint security audit commissioned by Hugging Face, EleutherAI, and Stability AI and conducted by Trail of Bits found no critical arbitrary code execution vulnerabilities in the format [3]. Hugging Face now offers automatic safetensors conversion for many repositories and displays format warnings for pickle-only models.

Adoption, however, has been uneven. A significant portion of established repositories continue to distribute only pickle-format weights, reflecting that PyTorch's default serialization format remains pickle-based, and many fine-tuning and training pipelines produce pickle output by default as a result. Organizations that have not explicitly enforced a safetensors-only policy in their model loading pipelines are likely still loading pickle-format weights in production without realizing it.

Recommendations

Immediate Actions

Organizations should audit their AI model inventories and deployment pipelines immediately, identifying every location where model weights are loaded from external sources and confirming the serialization format in use. Where pickle-format models are in active use, those models should be quarantined pending rescan with an up-to-date PickleScan build that includes patches for CVE-2025-10155, CVE-2025-10156, and CVE-2025-10157. Model loading in development and CI/CD environments should be conducted in network-isolated containers that can observe but not act on any outbound connections that a malicious payload might attempt to establish.

For organizations using OpenClaw or any other agent framework that consumes skills from ClawHub or similar registries, new skill imports should be suspended pending a full inventory review of currently deployed skills. Until the registry provides verified provenance or behavioral sandboxing for skill packages, organizations should treat third-party agent skills as untrusted code and evaluate them accordingly — including reviewing the full source of any skill before execution.

Short-Term Mitigations

The most durable mitigation for the pickle deserialization attack class is to eliminate pickle-format model loading from production pipelines entirely. Organizations should enforce a policy requiring safetensors format for any newly acquired model weights and should prioritize converting existing pickle-format inventories. Hugging Face's automatic conversion tooling can accelerate this process for models already hosted on the platform. Where conversion is not immediately feasible, model loading should be instrumented to detect anomalous outbound network connections, process spawning, or file system writes that would indicate malicious payload execution.

Provenance controls should be implemented for all AI model downloads. This means pinning model downloads to specific commit hashes rather than to floating branch references, verifying SHA-256 checksums against values obtained from a source separate from the download path, and maintaining a software bill of materials equivalent — an AI Model Bill of Materials — that records every model in use, its version, its source repository, and its last validation date. The same discipline applied to software dependencies should be applied to AI model dependencies.

For agent skill consumption, organizations should evaluate whether the operational need for third-party registry skills justifies the supply chain exposure. Internal skill libraries maintained under the same development and review controls as production code provide a more defensible posture than direct registry consumption, particularly for agentic systems that operate with broad tool access or sensitive credential stores.

Strategic Considerations

At the ecosystem level, the current state of AI model repositories reflects an early-stage supply chain that has not yet developed the trust infrastructure that software package ecosystems built over decades in response to analogous attacks. Code signing, reproducible builds, and auditable provenance have become established goals — and increasingly standard in high-assurance software ecosystems — over decades of supply chain incidents in the software industry; they remain largely absent for AI model weights and agent skills. The industry should treat building this infrastructure as a shared priority rather than a platform-by-platform decision.

Security teams should engage with their AI tool vendors and open-source framework maintainers to advocate for safetensors-by-default policies, mandatory signing requirements, and behavioral sandboxing for agent skill execution. Where organizations participate in AI platform governance, they should push for platform-level commitments to mandatory provenance disclosure, persistent author identity verification, and transparent incident reporting when malicious content is discovered and removed. Where malicious content has been removed from open repositories, the absence of a

corresponding public security advisory makes it difficult for affected practitioners to assess whether they downloaded content during the exposure window — a gap that community advocates and enterprise adopters should work to close.

CSA Resource Alignment

This research note connects directly to several CSA frameworks and initiatives. The AI Controls Matrix (AICM) [8] addresses AI supply chain security as a control domain spanning all provider roles — model providers, application providers, and orchestrated service providers — and explicitly calls for controls governing model integrity, provenance verification, and training data protection. Organizations aligning their AI deployment practices to AICM should map the model repository risk surface to the supply chain and model security control domains and assess gap closure against the threats documented here.

The MAESTRO threat modeling framework, developed by the CSA AI Safety Initiative, identifies supply chain compromise as a Layer 1 (foundational model) threat that propagates through every layer of an agentic AI stack. The ClawHavoc campaign is a concrete instantiation of a MAESTRO-described threat path: a poisoned agent skill at Layer 3 (agent capabilities) delivers a credential harvester that targets assets at Layer 5 (external integrations and financial services). Organizations applying MAESTRO to their agentic systems should include third-party skill registries explicitly in their threat models and assess the blast radius of a compromised skill given the agent's granted permissions.

The CSA STAR program provides an assessment and certification framework through which AI platform and repository operators can demonstrate security controls to their user communities. The absence of STAR-for-AI attestation by major open AI repository operators is a gap that the community and enterprise adopters should actively pressure to close. Mandatory disclosure of security scan coverage, incident statistics, and format safety policies — structured as STAR-compatible attestations — would give practitioners the basis for a risk-informed decision about which repositories to consume from and under what controls.

CSA's Zero Trust guidance is also directly applicable: the implicit trust that practitioners extend to downloads from reputable-seeming AI repositories is structurally equivalent to the implicit trust in network perimeters that Zero Trust was designed to eliminate. The default posture of Zero Trust — no implicit trust based on source reputation, always verify — applies as directly to a model weight downloaded from a popular repository as it does to a network packet arriving at an internal service.

References

[1] Koi Security. "ClawHavoc: 341 Malicious Clawed Skills Found by the Bot They Were Targeting." Koi Security Blog, February 2026.

[2] JFrog Security Research. "PyTorch Users at Risk: Unveiling 3 Zero-Day PickleScan Vulnerabilities." JFrog Blog, December 2025.

[3] Hugging Face. "SafeTensors Security Audit." Hugging Face Blog, 2023.

[4] JFrog Security Research. "Data Scientists Targeted by Malicious Hugging Face ML Models with Silent Backdoor." JFrog Blog, March 2024.

[5] SentinelOne. "NullBulge: Threat Actor Masquerades as Hacktivist Group Rebelling Against AI." SentinelOne Labs, 2024.

[6] Python Software Foundation. "pickle — Python Object Serialization." Python Documentation. (Security warning: "The pickle module is not secure. Only unpickle data you trust.")

[7] Palo Alto Networks Unit 42. "Model Namespace Reuse: Anatomy of an AI Supply Chain Attack." Palo Alto Networks, 2024.

[8] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.0." Cloud Security Alliance, 2025.

