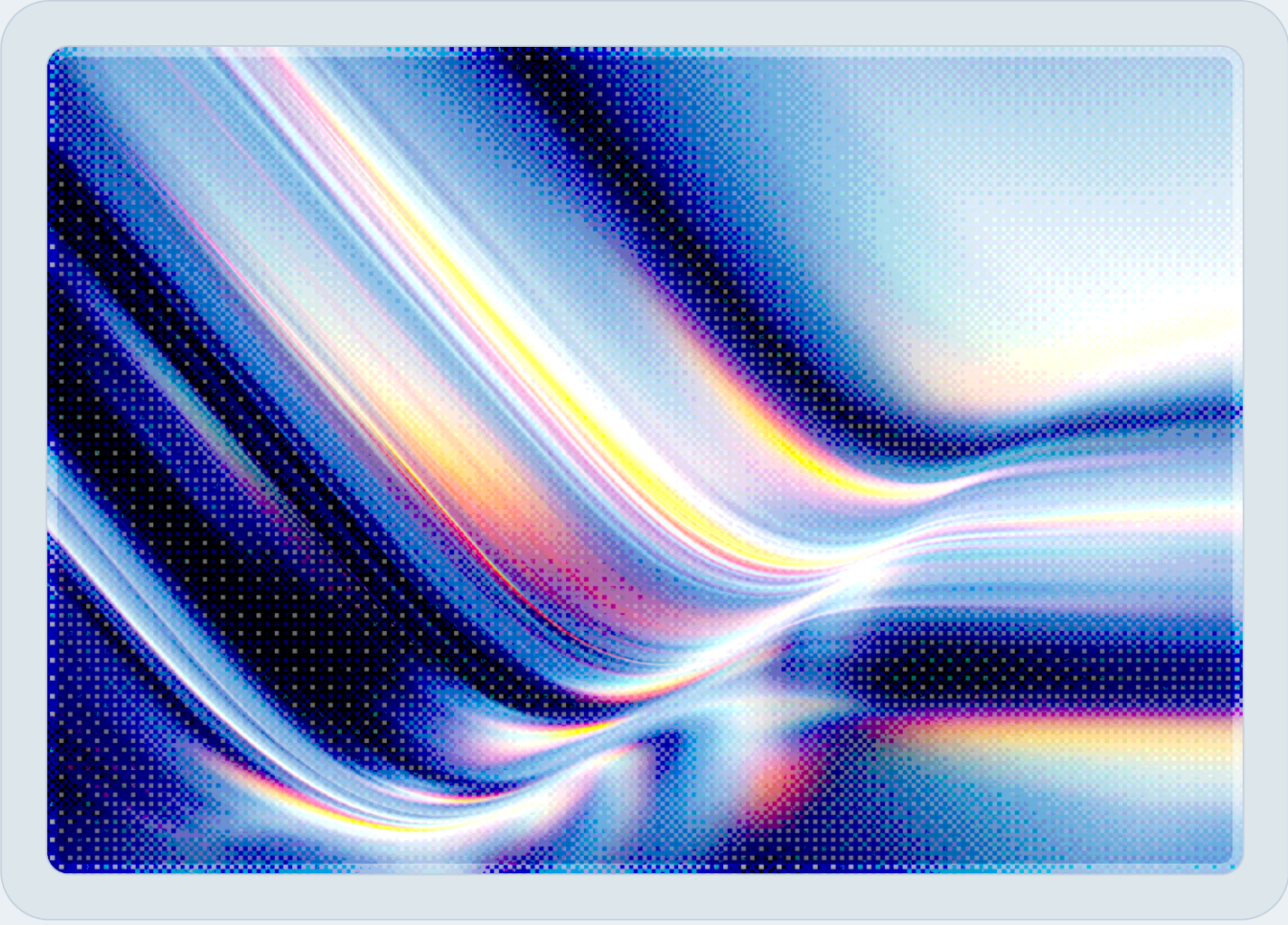
CSAI Foundation | Cloud Security Alliance

AI Hub Supply Chain Weaponization

Hugging Face and ClawHub Exploited as Malware Distribution Vectors

2026-05-04

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Acronis TRU and multiple independent security teams have confirmed that both Hugging Face and ClawHub — the skill marketplace for the OpenClaw AI agent framework — are being actively exploited as malware distribution channels. Across two Hugging Face campaigns and the ClawHub-focused ClawHavoc operation, threat actors staged multi-step infection chains delivering infostealers, trojans, cryptominers, and remote access tools targeting Windows, macOS, Linux, and Android systems [1][2].

- The ClawHavoc campaign poisoned the OpenClaw skill marketplace with 1,184 malicious skill packages attributed to 12 author accounts, with a single uploader identified as "hightower6eu" responsible for 677 packages in what appears to have been an automated operation [9][11]. Snyk's independent ToxicSkills audit of 3,984 skills from ClawHub and the companion registry skills.sh found that 13.4% (534 skills) contained critical-level security issues and 36.82% (1,467 skills) contained at least one security flaw of any kind — with prompt injection among the most prevalent categories — while 76 confirmed active malicious payloads were identified [10].

- On Hugging Face, ReversingLabs documented nullifAI — a technique using malformed Pickle files compressed with 7z rather than the ZIP format expected by PyTorch — to embed reverse-shell payloads that pass undetected through Hugging Face's automated scanning infrastructure [3][4]. Separately, CVE-2026-25874 (CVSS 9.8) exposes LeRobot installations to unauthenticated remote code execution through unsafe pickle deserialization over an unauthenticated gRPC endpoint, and the flaw remained unpatched as of this writing [5][6].

- Palo Alto Unit 42's Model Namespace Reuse research demonstrates a structural vulnerability in AI hub trust models: deleted Hugging Face namespaces are not permanently reserved, allowing attackers to re-register abandoned identities and intercept pipelines that resolve models by name without pinning a specific version [7]. This attack class has been demonstrated against both Google Vertex AI and Microsoft Azure AI Foundry.

- Both platforms share a common enabling condition: low barriers to content publication, no mandatory code signing or security review, and a shared trust assumption — embedded in developer workflows and automated CI/CD pipelines alike — that named resources from established AI platforms can be consumed without independent content verification [7].

Background

The supply chain attack pattern that security practitioners spent years documenting across npm, PyPI, and RubyGems has now extended into AI-native infrastructure. Hugging Face, which as of early 2026 hosts well over one million open-weight models [12] and serves as the dominant public model repository for AI research and commercial development, and ClawHub, the skill marketplace for the rapidly growing OpenClaw AI agent framework, have both become active targets for threat actors who recognized that the implicit trust users extend to these platforms has outpaced any security controls protecting them.

Hugging Face occupies a unique and structurally important position in the AI supply chain. Where npm distributes code, Hugging Face distributes model weights: binary artifacts that, when loaded, execute arbitrary code through Python's pickle deserialization mechanism. This means that the threat model for consuming a model from Hugging Face is meaningfully different from downloading a library — yet the dominant pattern in public repositories and tutorials — loading models with a single `from_pretrained()` call — implicitly treats platform availability as a proxy for safety. Hugging Face introduced automated scanning partnerships and model card warnings in response to earlier incidents, but the nullifAI technique demonstrated that those controls can be circumvented by attackers who understand the assumptions baked into the scanner's behavior.

ClawHub occupies an analogous position in the emerging AI agent ecosystem. OpenClaw allows users to extend their AI assistant's capabilities through installable skill packages — analogous to browser extensions or npm packages for AI agents — and ClawHub is the primary public registry for discovering and distributing those skills. At the time of the ClawHavoc disclosures in early February 2026, the only prerequisite for publishing a skill was a GitHub account at least one week old [8]. There was no automated static analysis, no code signing requirement, and no sandbox execution. Skills published under credible-looking names with professionally written documentation faced no meaningful friction before reaching users.

The Acronis TRU analysis found that malicious ClawHub skills used Hugging Face repositories as staging infrastructure for final payloads [1], indicating the two platforms functioned as complementary components of a single infection chain rather than independently targeted systems. This cross-platform architecture exploits the trust relationship users and security tools extend to both services simultaneously.

Security Analysis

ClawHavoc: Poisoning the OpenClaw Skill Marketplace

The scale and cross-platform architecture of ClawHavoc — 1,184 malicious packages across 12 coordinated accounts, with payload staging on Hugging Face — make it among the most extensively documented supply chain operations targeting an AI agent ecosystem to date. Antiy CERT's analysis identified these 1,184 malicious skill packages within ClawHub's historical repository [9]; the figure reflects a comprehensive scan of the registry's full upload history rather than only packages live at a single point in time. The OpenClaw Hub's own initial disclosure identified 341 malicious skills [14]; follow-on scanning of the full registry expanded that count to over 824 confirmed malicious skills in a registry that had grown to more than 10,700 total entries by mid-February 2026 [8]. A single account tracked as "hightower6eu" was responsible for 677 of the identified packages in what security researchers attributed to an automated toolchain rather than manual effort, with a second account, "sakaen736jih," accounting for an additional 390 packages [9][11].

The attack methodology combined technical deception with social engineering. Malicious skills were published under legitimate-sounding names — productivity tools, cryptocurrency trackers, coding assistants — accompanied by professional README documentation and SKILL.md files that instructed users to copy-paste terminal commands or download "helper tools" from external sites [8][9][11]. This ClickFix-style social engineering pattern exploits the tendency of technically sophisticated users to follow structured setup instructions without scrutinizing their content. Beyond direct user manipulation, several malicious skills embedded hidden prompt injection payloads in their descriptions, instructing the AI agent itself to execute commands on behalf of users without their explicit knowledge [1][10].

The payloads delivered through ClawHub targeted multiple platforms. SecurityWeek's reporting attributes the broader campaign to attacks across Windows, macOS, Linux, and Android systems [2], though technical disclosures on Android-specific payloads were not available at time of writing. On macOS, users received the Atomic macOS Stealer (AMOS), a credential-harvesting infostealer distributed commercially as malware-as-a-service and previously documented in campaigns targeting macOS developers [1][2]. Windows users received keyloggers and trojans designed to harvest stored credentials, browser session cookies, and cryptocurrency wallet files. The use of AMOS — a commercial MaaS product — indicates financially motivated operators seeking to maximize credential yield across developer workstations [1][2]. Whether the 12 accounts represent a single coordinated actor or a loosely affiliated operation has not been publicly confirmed.

The Snyk ToxicSkills audit provides essential context for evaluating ClawHavoc's scope relative to the broader ecosystem. Scanning 3,984 skills from ClawHub and the companion registry skills.sh, Snyk found 534 skills (13.4%) with critical-level security issues, 1,467 skills (36.82% of the audited corpus) containing at least one security flaw of any kind — with prompt injection among the most prevalent categories — and 76 confirmed active malicious payloads [10]. These figures predate the formal disclosure of ClawHavoc and indicate that the campaign identified by security researchers represents a subset of a broader, ongoing abuse of the skill publishing model.

nullifAI and the Limits of Model Hub Scanning

The nullifAI technique — documented by ReversingLabs in February 2025 — reveals a category-level limitation in signature-based scanning: when the scanner cannot parse the container format, it cannot evaluate the content [3][4]. The platform relies on automated scanning that analyzes PyTorch model files by inspecting their Pickle content for known malicious opcodes. The technique circumvents this by packaging the model as a 7z archive rather than the ZIP format that PyTorch's `torch.load()` function and Hugging Face's scanner expect. Because the scanner cannot parse the 7z format, it fails to flag the file as unsafe. Meanwhile, a custom loader can still extract and execute the Pickle payload, which in the discovered models consisted of a platform-aware reverse shell connecting back to a hardcoded IP address. ReversingLabs characterized the identified models as proof-of-concept rather than a confirmed active campaign, and Hugging Face removed them within 24 hours of notification [3]. The technique demonstrates that scanner-based defenses can be defeated by an attacker who understands how the scanner operates — an adversarial dynamic that is likely to persist unless the ecosystem migrates away from arbitrary-execution serialization formats such as pickle, a transition that safetensors adoption has begun but not completed.

Hugging Face's model scanning partnership with Protect AI provides broader context for the scale of the problem. As of their six-month joint report, Protect AI had scanned 4.47 million unique model versions and identified 352,000 unsafe or suspicious issues across 51,700 models [12]. The majority of these issues trace to the use of Python's pickle format for model serialization — a format that is, by design, capable of executing arbitrary Python code during deserialization. The nullifAI technique illustrates a recurring pattern in scanner evasion: attackers who understand a detection system's parsing assumptions can engineer artifacts that satisfy the scanner while delivering malicious content.

CVE-2026-25874 and the gRPC Attack Surface

Hugging Face's LeRobot framework, designed to support robotic AI applications, carries a critical unpatched vulnerability that extends the pickle deserialization risk from model loading into runtime infrastructure. CVE-2026-25874 (CVSS 9.8) affects the asynchronous inference PolicyServer

component, which offloads policy computation to a GPU-backed server via a gRPC endpoint [5][6]. The server uses Python's `pickle.loads()` to deserialize incoming requests across multiple RPC endpoints, and it binds using `add_insecure_port()` with no Transport Layer Security and no authentication requirement. Any network-accessible host running the PolicyServer can be exploited by an unauthenticated attacker to execute arbitrary code, with potential impact extending to connected robotic systems.

A particularly significant detail in the disclosure is that the vulnerable source code contained `# nosec` annotations directly adjacent to the `pickle.loads()` calls — comments placed to suppress warnings from Bandit and similar Python security linters that correctly flagged the deserialization calls during development [5]. The presence of these annotations indicates the warnings were suppressed during development rather than overlooked. Whether this reflects a deliberate risk acceptance decision or a workflow artifact, the effect was that a critical deserialization flaw in a publicly distributed framework escaped pre-release static analysis. CSA's own analysis of CVE-2026-25874 is available through CSA Labs [13].

Model Namespace Reuse: The Structural Trust Gap

The Model Namespace Reuse vulnerability, documented by Palo Alto Networks Unit 42, identifies a supply chain risk that is independent of any single model file's content [7]. When a Hugging Face user or organization deletes their account, the associated namespace is not permanently retired. Any actor can subsequently register the same namespace and begin publishing models under the original account's identity. Pipelines and code that reference models by name — using patterns like `model = AutoModel.from_pretrained("OriginalOrg/ModelName")` without a version pin or cryptographic hash — will silently resolve to the attacker-controlled replacement.

Unit 42's analysis found thousands of public GitHub repositories containing unversioned model references that would be susceptible to this technique [7]. Their researchers demonstrated the attack class against both Google Vertex AI's Model Garden and Microsoft Azure AI Foundry, both of which allow direct deployment of Hugging Face models by name. Google has since implemented daily scans to identify orphaned model references, but the underlying architectural gap — that name-based trust without content verification is structurally equivalent to trusting an unverified download URL — persists across the ecosystem wherever models are consumed by reference rather than by verified content hash.

Recommendations

Immediate Actions

Organizations using OpenClaw should audit all installed ClawHub skills against the lists of confirmed malicious accounts published by Antiy CERT and OpenClaw Hub [9][14]. Any environment that installed skills from accounts identified as part of the ClawHavoc campaign between early February 2026 and the date of this writing should be treated as potentially compromised. This means rotating credentials stored in browser profiles, credential managers, environment variables, and configuration files on any host where an affected skill was installed, and reviewing outbound network connections for communication with known AMOS command-and-control infrastructure.

Organizations running Hugging Face LeRobot should immediately assess exposure of the PolicyServer gRPC endpoint. Until an official patched release is available — fixes are planned for version 0.6.0 per published disclosures [5] — the endpoint should be firewalled to restrict access to trusted source IPs, and `--insecure-port` bindings should not be exposed to untrusted networks. Deployments in research or cloud environments where GPU-backed inference servers may be network-accessible should be treated as actively at risk until patching is complete. Additionally, all Hugging Face model references in CI/CD pipelines, model training scripts, and production inference code should be reviewed for unversioned or unhashed model identifiers; the `revision` parameter should be added to all `from_pretrained()` calls to pin to a specific commit hash rather than a branch or tag name. Any model loaded through the pickle format without a verified SHA256 hash should be treated as an unverified binary and evaluated with corresponding scrutiny.

Short-Term Mitigations

For ClawHub and AI agent skill consumption, organizations should establish an approved-skill registry — an internal allowlist of vetted skill packages with verified version hashes — before permitting AI agents to install or execute community skills in enterprise environments. Skills should be evaluated in an isolated sandbox before being approved for production use, with network egress restricted during evaluation to observe any anomalous outbound connections. Prompt injection auditing should be applied to skill descriptions and SKILL.md files before installation, as the ToxicSkills study found prompt injection among the most prevalent flaw categories across the broader skill ecosystem [10].

For Hugging Face model consumption, organizations should shift from name-based to content-verified model resolution. This means storing the expected SHA256 hash of model weights alongside model references in code and CI configuration, and failing builds or deployments when the retrieved artifact

does not match the expected hash. This control mitigates both the nullifAI-style file-substitution technique and the Model Namespace Reuse attack, since a replaced model will produce a different hash regardless of how the substitution was achieved. Private model mirrors — hosting verified copies of approved models in a controlled artifact registry — provide an additional layer of provenance control and reduce exposure to namespace takeover events on the public platform.

Automated scanning should be treated as a necessary but insufficient control for both model files and skill packages. The nullifAI technique demonstrates that scanner evasion is feasible for motivated attackers, and the ToxicSkills audit found that prompt injection payloads — which operate at the semantic layer rather than the binary layer — are not detectable by traditional signature-based scanners. Defense in depth requires combining scanner-based detection with behavioral monitoring, network egress inspection, and provenance verification.

Strategic Considerations

The same properties that made Hugging Face and ClawHub effective benign distribution platforms — broad community trust, frictionless access, and deep integration into developer workflows — are precisely what make them valuable as malware delivery channels. AI model hubs are following the same adversarial trajectory that npm and PyPI experienced as they scaled — initially low-friction publishing environments that became active targets as their trust surface grew — with the added complexity that the artifacts being distributed — model weights and agent skills — execute through mechanisms that traditional endpoint and application security tools were not designed to inspect.

Organizations building AI-powered products should designate a responsible owner for AI supply chain security equivalent to the role that library dependency governance plays in traditional software engineering. This means establishing a policy for which model providers and registries are approved for use, requiring provenance documentation for any model or skill incorporated into a production system, and including AI artifact provenance in Software Bill of Materials (SBOM) outputs. The SBOM discipline that is now standard for software libraries needs to extend to model weights, fine-tuning datasets, and agent skill packages with equal rigor.

Platform operators — both Hugging Face and the OpenClaw project — face architectural decisions about the trust model their platforms offer. Mandatory code signing for skill packages, content-addressed storage (where artifacts are retrieved by hash rather than by name), and behavioral sandboxing for model execution at upload time would each reduce the attack surface that ClawHavoc and nullifAI exploited. The trajectory of platform security in traditional software registries suggests these controls will be adopted, but the timeline is measured in years, not months, and the threat actors currently exploiting the gap are not waiting.

CSA Resource Alignment

The attack patterns documented in this research note map to multiple CSA frameworks that provide structured guidance for organizations assessing and mitigating AI supply chain risk.

CSA's AI Controls Matrix (AICM), as a superset of the Cloud Controls Matrix, addresses software supply chain risk through controls requiring provenance verification, software composition analysis, and third-party component validation. The AICM's model provider domain is directly applicable to Hugging Face consumption: organizations using community models from public repositories should be applying the same controls they would apply to any unverified third-party library, including integrity verification, behavioral testing in isolation, and documented approval workflows. The ClawHub skill ecosystem falls under the AICM's application provider and orchestrated service provider domains, where controls on third-party integrations and runtime environment trust apply.

CSA's MAESTRO framework for agentic AI threat modeling identifies the skill and tool ecosystem as a primary attack surface for AI agents at Layer 1 (Model and Inference) and Layer 3 (Infrastructure). ClawHavoc represents a MAESTRO Layer 1 threat: the agent's capability set is extended with attacker-controlled instructions that execute within the agent's permission context. The prompt injection payloads embedded in malicious ClawHub skill descriptions are specifically addressed by MAESTRO's indirect instruction injection threat category, in which adversarial content delivered through a trusted channel hijacks agent behavior without the user's knowledge.

The STAR program's security assessment questionnaire provides a vehicle for organizations to evaluate AI platform providers against the baseline expectations that the Hugging Face and ClawHub incidents reveal as gaps. Security teams performing vendor risk assessments on AI model hubs and agent skill registries should include questions about upload integrity controls, namespace lifecycle policies, scanning infrastructure and its known limitations, and incident response commitments for malicious content removal.

CSA's Zero Trust guidance applies directly to the trust assumptions that both ClawHavoc and nullifAI exploit. The core Zero Trust principle — never trust, always verify — should be operationalized for AI artifact consumption as: no model or skill is trusted because it comes from a named, reputable source; trust is established through content verification (hash matching), behavioral analysis, and scoped execution that limits what an artifact can do even if it is malicious. The alternative — extending implicit trust to any artifact that passes a platform's automated screening — is a trust assumption that adversaries have now demonstrated they can defeat.

References

[1] Acronis TRU. "Poisoning the well: AI supply chain attacks on Hugging Face and OpenClaw." Acronis Threat Research Unit, 2026.

[2] SecurityWeek. "Hugging Face, ClawHub Abused for Malware Distribution." SecurityWeek, May 2026.

[3] ReversingLabs. "Malicious ML models discovered on Hugging Face platform." ReversingLabs Blog, February 2025.

[4] The Hacker News. "Malicious ML Models Found on Hugging Face Leverage Broken Pickle Format to Evade Detection." The Hacker News, February 2025.

[5] The Hacker News. "Critical CVE-2026-25874 Leaves Hugging Face LeRobot Open to Unauthenticated RCE." The Hacker News, April 2026.

[6] Resecurity. "CVE-2026-25874: Hugging Face LeRobot Unauthenticated RCE via Pickle Deserialization." Resecurity Blog, April 2026.

[7] Palo Alto Networks Unit 42. "Model Namespace Reuse: An AI Supply-Chain Attack Exploiting Model Name Trust." Palo Alto Networks, 2025.

[8] Conscia. "The OpenClaw security crisis." Conscia Blog, February 2026.

[9] Antiy Labs. "ClawHavoc: Analysis of Large-Scale Poisoning Campaign Targeting the OpenClaw Skill Market for AI Agents." Antiy CERT, February 2026.

[10] Snyk. "Snyk Finds Prompt Injection in 36%, 1467 Malicious Payloads in a ToxicSkills Study of Agent Skills Supply Chain Compromise." Snyk Security Research, February 2026.

[11] CyberPress. "ClawHavoc Poisons OpenClaw's ClawHub With 1,184 Malicious Skills." CyberPress, February 2026.

[12] Protect AI / Hugging Face. "4M Models Scanned: Protect AI + Hugging Face 6 Months In." Hugging Face Blog, April 2025.

[13] Cloud Security Alliance Labs. "LeRobot CVE-2026-25874: Unauthenticated RCE via Pickle." CSA Labs Research, April 2026.

[14] OpenClaw Hub. "ClawHavoc Incident – 341 Malicious ClawHub Skills Report." OpenClaw Hub Security Advisories, February 2026.