INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-24 18:57 UTC

Malicious Packages in AI Skills Marketplace Bypass Vetting, Deliver Infostealers to Agent Pipelines

THREAT CAMPAIGN | HIGH | CVSS 5.0

SCC Item ID	SCC-CAM-2026-0557
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	5.0
Affected Products	OpenClaw AI agent platform; ClawHub skills marketplace (all versions pulling unvetted third-party skills)
Published	2026-06-24T12:56:49
Discovery Source	Rss

Executive Summary

Attackers distributed over 800 malicious packages through ClawHub, the third-party skills marketplace for the OpenClaw AI agent platform, bypassing platform security vetting and delivering infostealers directly into AI agent pipelines. Snyk's ToxicSkills research identified 1,467 malicious payloads and prompt injection vulnerabilities in 36% of analyzed skills; a separate Silverfort disclosure revealed a ranking manipulation vulnerability that surfaced malicious packages as top results, increasing installation volume. Organizations running OpenClaw with third-party ClawHub skills face credential theft, session token compromise, and potential lateral movement into any downstream system the agent is authorized to access.

Technical Analysis

Malicious packages distributed via ClawHub, the third-party skills registry for the OpenClaw AI agent platform, bypassed platform vetting controls and delivered infostealer payloads to consuming agent pipelines. Snyk's ToxicSkills research reported 1,467 malicious payloads across ClawHub skills and identified prompt injection vulnerabilities in 36% of analyzed skills. A separate Silverfort disclosure describes a vulnerability in ClawHub's ranking algorithm that allowed attackers to manipulate skill rankings, surfacing malicious packages as top-result recommendations and significantly increasing installation volume. The attack is structurally analogous to npm/PyPI supply chain compromise (T1195.001, T1195.002) but targets AI agent ecosystems where skills may execute with elevated trust and reduced sandboxing. Infostealer payloads (T1555, T1552.001) targeting agent runtime contexts can capture credentials, API tokens, and session data that grant access to downstream

systems the agent is authorized to interact with. Exfiltration paths map to T1567 and T1071. Additional technique coverage includes T1059 (script execution within pipeline), T1554 (compromise of software supply chain artifacts), and T1566 (initial delivery vector). Relevant weaknesses: CWE-693 (Protection Mechanism Failure), CWE-494 (Download of Code Without Integrity Check), CWE-829 (Inclusion of Functionality from Untrusted Control Sphere), CWE-1357 (Reliance on Insufficiently Trustworthy Component). No CVE identifier has been assigned to this campaign. No vendor patch status is confirmed in source material as of this writing; OpenClaw and ClawHub remediation guidance should be obtained directly from vendor advisories.

Action Checklist

1. Step 1: Containment, Immediately audit all installed ClawHub skills across OpenClaw deployments. Suspend or disable any third-party skills installed from ClawHub that were not explicitly vetted by your organization. Prioritize skills installed within the last 90 days. Reference CIS 2.3: Address Unauthorized Software, treat unvetted skills as unauthorized software pending review.
2. Step 2: Detection, Review agent pipeline execution logs for anomalous outbound connections, credential access patterns, or unexpected data transfers attributable to skills processes. Look for T1552.001 indicators (credential file access), T1567/T1071 indicators (unusual HTTPS or DNS exfiltration from agent processes), and T1059 indicators (unexpected script execution spawned by skill components). Cross-reference installed skill hashes against the Snyk ToxicSkills and Silverfort disclosures. Reference NIST AU-6: Audit Record Review, Analysis, and Reporting. Apply D3 System File Analysis (SFA) to agent runtime directories for evidence of infostealer staging.
3. Step 3: Eradication, Remove all ClawHub-sourced skills that cannot be positively verified as clean through source review or vendor confirmation. Do not rely on ClawHub ranking or vetting signals as a trust indicator given the confirmed ranking manipulation vulnerability. Reference CIS 2.1: Establish and Maintain a Software Inventory, ensure every installed skill is inventoried with provenance. Apply NIST CM controls: treat skills as software components subject to change management and integrity verification (CWE-494 remediation: require cryptographic integrity checks before installation). Apply D3 Credential Rotation (CRO) for any credentials or tokens accessible to agent pipelines running affected skills.
4. Step 4: Recovery, After removing suspect skills, rotate all credentials, API keys, and session tokens that were accessible to affected agent pipelines. Audit downstream systems the agent was authorized to interact with for signs of unauthorized access. Re-enable only positively verified skills. Validate that agent pipelines resume operation without anomalous behavior. Monitor with NIST SI-4 (System Monitoring) controls for 30 days post-remediation. Apply D3 Local Account Monitoring (LAM) to accounts used by agent service identities.
5. Step 5: Post-Incident, This incident exposes a structural gap in AI agent supply chain governance. Implement a formal approval process for all third-party skills before installation, analogous to software composition analysis (SCA) in traditional development pipelines. Reference CIS 7.1: Establish and Maintain a Vulnerability Management Process, extend scope explicitly to AI agent skills and plugin registries. Reference NIST AC-6 (Least Privilege): review and restrict the permissions granted to agent pipeline service accounts. Evaluate D3 User Account Permissions (UAP) to scope agent runtime privileges to minimum required access. Document findings and update your AI system procurement and third-party risk policies.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to legal, privacy, and senior leadership if forensic review of OpenClaw pipeline logs or downstream system access logs confirms that the agent service identity accessed, read, or transmitted data containing PII, PHI, financial records, authentication credentials, or API keys to any external destination, as this triggers breach notification obligations under GDPR, CCPA, HIPAA, or applicable state law depending on data classification.
Recovery Notes	After eradicating all unverified ClawHub skills and rotating credentials, re-admit only skills with a confirmed SHA-256 hash matching a source retrieved directly from the skill author's official repository — not from ClawHub, given the confirmed ranking manipulation vulnerability that allowed malicious packages to surface as trusted top results. Monitor all re-enabled agent pipelines for 30 days using DNS query logs and outbound connection logs, specifically watching for the exfiltration channel patterns (anomalous HTTPS POSTs or DNS TXT queries from the OpenClaw process) identified during detection. Validate that downstream systems accessed by the agent show no continued unauthorized activity after credential rotation by reviewing their access logs weekly for the same 30-day window.
Forensic Artifacts	OpenClaw agent pipeline execution logs (~/.openclaw/logs/ or platform-equivalent): contain skill invocation records, child process spawns, and outbound API call destinations — the primary source for identifying which malicious ClawHub skill executed and what actions it took within the pipeline | File system artifacts in agent skill directories (~/.openclaw/skills//): include the skill package contents, any secondary payloads staged by the infostealer, and install timestamps that establish when a ToxicSkills-family package entered the environment | Environment variable dumps from the OpenClaw agent process (/proc//environ on Linux): capture the exact set of credentials, API keys, and tokens that were accessible in the agent runtime at time of compromise — directly reflects what the infostealer could have exfiltrated | Outbound network capture (pcap) from the OpenClaw agent process PID: reveals the specific exfiltration channels used by the infostealer payload, including C2 domains, DNS tunneling patterns, or HTTPS POST destinations not matching legitimate LLM provider endpoints (openai.com, anthropic.com) | Downstream system access logs for the agent service account identity covering the 90-day ClawHub install window: establish whether the compromised agent pipeline was used to pivot into internal APIs, data stores, or SaaS platforms the agent was authorized to access, scoping the full blast radius beyond the OpenClaw host itself

Per-Action IR Details

Step 1: Containment — Immediately audit all installed ClawHub skills across OpenClaw deployments. Suspend or disable any third-party skills installed from ClawHub that were not explicitly vetted by your organization. Prioritize skills installed within the last 90 days. Reference CIS 2.3: Address Unauthorized Software — treat unvetted skills as unauthorized software pending review.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: CIS 2.3 (Establish and Maintain a Software Inventory) — Address Unauthorized Software, CIS 2.1 (Establish and Maintain a Software Inventory) — Establish and Maintain a Software Inventory, NIST AC-3 (Access Enforcement)

Compensating: On each OpenClaw host, enumerate installed skills via the OpenClaw CLI or config directory (e.g., `find ~/.openclaw/skills/ -type d -mtime -90` on Linux or equivalent Windows path under `%APPDATA%\OpenClaw\skills\`). Cross-reference directory listing against your organization's approved-skills list in a spreadsheet or text file. Disable unapproved skills by removing or renaming their manifest files and restarting the agent service — no SIEM required.

Evidence: Before suspending or disabling any skill, capture: (1) a full directory listing with timestamps (`ls -laR ~/.openclaw/skills/` or `dir /s /tc %APPDATA%\OpenClaw\skills\`), (2) a snapshot of currently running OpenClaw agent processes and their loaded modules (`ps aux | grep openclaw` or `Get-Process | Where-Object {$_.Name -like '*openclaw*'}`), and (3) active outbound network connections from agent processes (`ss -tp` or `Get-NetTCPConnection | Where-Object {$_.OwningProcess -in (Get-Process openclaw).Id}`). These are volatile and will be lost once skills are disabled or the agent is restarted.

Step 2: Detection — Review agent pipeline execution logs for anomalous outbound connections, credential access patterns, or unexpected data transfers attributable to skills processes. Look for T1552.001 indicators (credential file access), T1567/T1071 indicators (unusual HTTPS or DNS exfiltration from agent processes), and T1059 indicators (unexpected script execution spawned by skill components). Cross-reference installed skill hashes against the Snyk ToxicSkills and Silverfort disclosures. Reference NIST AU-6: Audit Record Review, Analysis, and Reporting. Apply D3-SFA (System File Analysis) to agent runtime directories for evidence of infostealer staging.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation)

Compensating: Without a SIEM, parse OpenClaw pipeline execution logs manually: grep for outbound connections not matching expected LLM API endpoints (e.g., `grep -E '(https?://|dns:)' ~/.openclaw/logs/pipeline.log | grep -v 'openai.com\|anthropic.com\|openclaw.io'`). Use `sha256sum` (Linux) or `Get-FileHash` (PowerShell) on each installed skill package and compare hashes against Snyk ToxicSkills and Silverfort published IOC lists. Use Wireshark or `tcpdump -i any -w agent_capture.pcap host $(hostname)` during a controlled agent execution to capture live traffic for offline analysis.

Evidence: Capture before analysis modifies any state: (1) OpenClaw pipeline execution logs in full (`~/.openclaw/logs/` or equivalent), including skill invocation records that would show which skill component spawned child processes or made outbound calls; (2) a live `tcpdump` or Wireshark capture of DNS queries and HTTPS connections from the agent process PID to detect exfiltration channels used by ToxicSkills-family infostealers; (3) file system access audit entries showing reads against credential stores (e.g., `~/.aws/credentials`, `~/.ssh/`, browser credential databases, or environment variable snapshots) by the OpenClaw process; (4) SHA-256 hashes of all installed skill package archives before any are removed.

Step 3: Eradication — Remove all ClawHub-sourced skills that cannot be positively verified as clean through source review or vendor confirmation. Do not rely on ClawHub ranking or vetting signals as a trust indicator given the confirmed ranking manipulation vulnerability. Reference CIS 2.1: Establish and Maintain a Software Inventory — ensure every installed skill is inventoried with provenance. Apply NIST CM controls: treat skills as software components subject to change management and integrity verification (CWE-494 remediation: require cryptographic integrity checks before installation). Apply D3-CRO (Credential Rotation) for any credentials or tokens accessible to agent pipelines running affected skills.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 2.1 (Establish and Maintain a Software Inventory) — Establish and Maintain a Software Inventory, CIS 2.2 (Establish and Maintain a Software Inventory) — Ensure Authorized Software is Currently Supported, CIS 2.3 (Establish and Maintain a Software Inventory) — Address Unauthorized Software

Compensating: Perform a verified removal: for each unvetted ClawHub skill, delete the skill directory and confirm removal (`rm -rf ~/.openclaw/skills//` then verify with `ls`). After removal, run `find / -name '*.py' -newer /tmp/baseline_timestamp -path '*/openclaw/*' 2>/dev/null` to detect any skill-dropped files outside the skills directory. Use ClamAV (`clamscan -r ~/.openclaw/`) to scan remaining skill files and the agent working directory for infostealer signatures. Document each removed skill with its install timestamp, hash, and removal date in a plain-text incident log.

Evidence: Before removing any skill package, preserve: (1) a forensic copy of the full skill directory (`cp -a ~/.openclaw/skills// /evidence/skills-backup//`) to support later malware analysis; (2) any staged payload files or secondary scripts dropped by the skill into temp directories (`/tmp`, `%TEMP%`, or the agent's working directory); (3) environment variable dumps from the agent process (`/proc//environ` on Linux or `Get-Process openclaw | ForEach-Object { $_.StartInfo.EnvironmentVariables }`) which may capture API keys and tokens the infostealer exfiltrated; (4) a copy of the OpenClaw skills manifest or lockfile showing provenance metadata for each installed skill.

Step 4: Recovery — After removing suspect skills, rotate all credentials, API keys, and session tokens that were accessible to affected agent pipelines. Audit downstream systems the agent was authorized to interact with for signs of unauthorized access. Re-enable only positively verified skills. Validate that agent pipelines resume operation without anomalous behavior. Monitor with NIST SI-4 (System Monitoring) controls for 30 days post-remediation. Apply D3-LAM (Local Account Monitoring) to accounts used by agent service identities.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AC-2 (Account Management), NIST AC-12 (Session Termination), CIS 5.1 (Establish and Maintain an Inventory of Accounts) — Establish and Maintain an Inventory of Accounts, CIS 6.2 (Establish an Access Revoking Process) — Establish an Access Revoking Process

Compensating: Enumerate all secrets accessible to the OpenClaw agent service account by reviewing environment variables, `.env` files, and secrets mounted into the agent runtime. Rotate each credential at the issuing service (e.g., revoke and reissue AWS IAM keys via `aws iam delete-access-key` + `aws iam create-access-key`, revoke GitHub PATs via the GitHub UI, invalidate OpenAI API keys in the developer dashboard). After rotation, review downstream system access logs — for each system the agent was authorized to call, query its access logs for the agent service account identity for the 90-day window matching the ClawHub install history. Re-enable verified skills only after confirming their SHA-256 hash matches a known-good source outside ClawHub.

Evidence: Before rotating any credential, document and preserve: (1) a complete list of all credentials, tokens, and API keys accessible to the affected agent pipeline (from environment variables, config files, and secrets managers), including their last-used timestamps from the issuing service's audit log — this establishes the exfiltration scope; (2) downstream system access logs for the agent service identity covering the 90-day exposure window, capturing any access to data stores, APIs, or internal services the agent was permitted to reach; (3) the agent's session token and authentication artifacts (e.g., OAuth tokens, JWT payloads) still present in memory or on disk before invalidation.

Step 5: Post-Incident — This incident exposes a structural gap in AI agent supply chain governance. Implement a formal approval process for all third-party skills before installation, analogous to software composition analysis (SCA) in traditional development pipelines. Reference CIS 7.1: Establish and Maintain a Vulnerability Management Process — extend scope explicitly to AI agent skills and plugin registries. Reference NIST AC-6 (Least Privilege): review and restrict the permissions granted to agent pipeline service accounts. Evaluate D3-UAP (User Account Permissions) to scope agent runtime privileges to minimum required access. Document findings and update your AI system procurement and third-party risk policies.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — Establish and Maintain a Vulnerability Management Process, CIS 7.2 (Establish and Maintain a Remediation Process) — Establish and Maintain a Remediation Process, NIST AC-6 (Least Privilege), NIST AC-2 (Account Management), CIS 2.1 (Establish and Maintain a Software Inventory) — Establish and Maintain a Software Inventory

Compensating: Without a dedicated SCA platform, implement a lightweight approval gate: create a required-review checklist (source repository URL, last commit date, author identity, SHA-256 hash, manual code review sign-off) that must be completed before any ClawHub skill is installed. Store approvals in a shared Git repository as version-controlled YAML files. For ongoing monitoring, set up a cron job or scheduled task that hashes all installed skills weekly and diffs against the approved-hashes file, alerting via email if a mismatch is detected. Restrict agent

service accounts to the minimum required API scopes using the principle of least privilege, enforced at the issuing service (e.g., AWS IAM policies scoped to specific S3 buckets, not `s3:*`).

Evidence: Post-incident documentation to preserve: (1) the full timeline of ClawHub skill installations across all OpenClaw deployments, reconstructed from install logs and file system timestamps, to establish the incident window and scope for any required regulatory notification; (2) the lessons-learned record specifying which skills were confirmed malicious, which were confirmed clean, and which remain unresolved — this feeds the updated approved-skills inventory; (3) before policy changes are finalized, capture the current agent service account permission sets from each integrated system as a baseline to measure least-privilege enforcement progress.

Detection Guidance

Detection should focus on agent pipeline runtime behavior and outbound network activity. Key indicators: (1) Outbound connections from agent processes to unrecognized hosts, particularly over HTTPS on non-standard ports, maps to T1071 and T1567. (2) File system access by skill processes to credential stores, token caches, browser storage directories, or environment variable files, maps to T1552.001 and T1555. (3) Script interpreter invocations (Python, PowerShell, shell) spawned as child processes of agent pipeline workers, maps to T1059. (4) Skill component hashes not matching expected values at install time, maps to CWE-494 and CWE-829. Query agent execution logs for skills installed from ClawHub between initial availability and the disclosure date. Cross-reference skill package names and hashes against the Snyk ToxicSkills disclosure (1,467 identified payloads) and Silverfort's ranking manipulation research. Apply D3 System File Analysis (SFA) to agent working directories. Apply D3 Local Account Monitoring (LAM) to service accounts used by OpenClaw agent processes. NIST AU-2 (Event Logging) and AU-6 (Audit Record Review) should be confirmed active for agent pipeline hosts. Note: no specific IOC hash list was available from source material at time of writing, consult Snyk's ToxicSkills blog and Silverfort's ClawHub disclosure directly for current IOC lists.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://snyk.io/blog/toxicskills-malicious-ai-agent-skills-clawhub/`	Snyk ToxicSkills research listing malicious ClawHub skill payloads — consult for current IOC and package name list	HIGH
URL	`https://www.silverfort.com/blog/clawhub-vulnerability-enables-attackers-to-manipulate-rankings-to-become-the-number-one-skill/`	Silverfort disclosure of ClawHub ranking manipulation vulnerability — consult for affected skill identifiers	HIGH

Framework Mappings

MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1567** — Exfiltration Over Web Service
- **T1059** — Command and Scripting Interpreter

- **T1555** — Credentials from Password Stores
- **T1071** — Application Layer Protocol
- **T1554** — Compromise Host Software Binary
- **T1566** — Phishing
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1195.002** — Compromise Software Supply Chain

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **AT-2** — Literacy Training and Awareness
- **SI-8** — Spam Protection
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **6.3** — Require MFA for Externally-Exposed Applications
- **15.1** — Establish and Maintain an Inventory of Service Providers

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC9.2** — Manages risks associated with vendors and business partners

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.001	Credentials In Files	Credential-Access
T1567	Exfiltration Over Web Service	Exfiltration
T1059	Command and Scripting Interpreter	Execution
T1555	Credentials from Password Stores	Credential-Access
T1071	Application Layer Protocol	Command-And-Control
T1554	Compromise Host Software Binary	Persistence
T1566	Phishing	Initial-Access
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1195.002	Compromise Software Supply Chain	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/cyber-risk/malicious-openclaw-skills-cl...	T3
ClawHub vulnerability puts malicious skill at #1 - Silverfort	https://www.silverfort.com/blog/clawhub-vulnerability-enables-attac...	T3
820 Malicious Skills Found in OpenClaw's ClawHub Marketplace ...	https://www.reddit.com/r/cybersecurity/comments/1rskb80/820_malicio...	T3
Why You Should Uninstall OpenClaw AI Immediately - Immersive Labs	https://www.immersivelabs.com/resources/c7-blog/openclaw-what-you-n...	T3
Snyk Finds Prompt Injection in 36%, 1467 Malicious Payloads in a ...	https://snyk.io/blog/toxicskills-malicious-ai-agent-skills-clawhub/	T3

Generated 2026-06-24 18:57 UTC by TJS Security Command Center