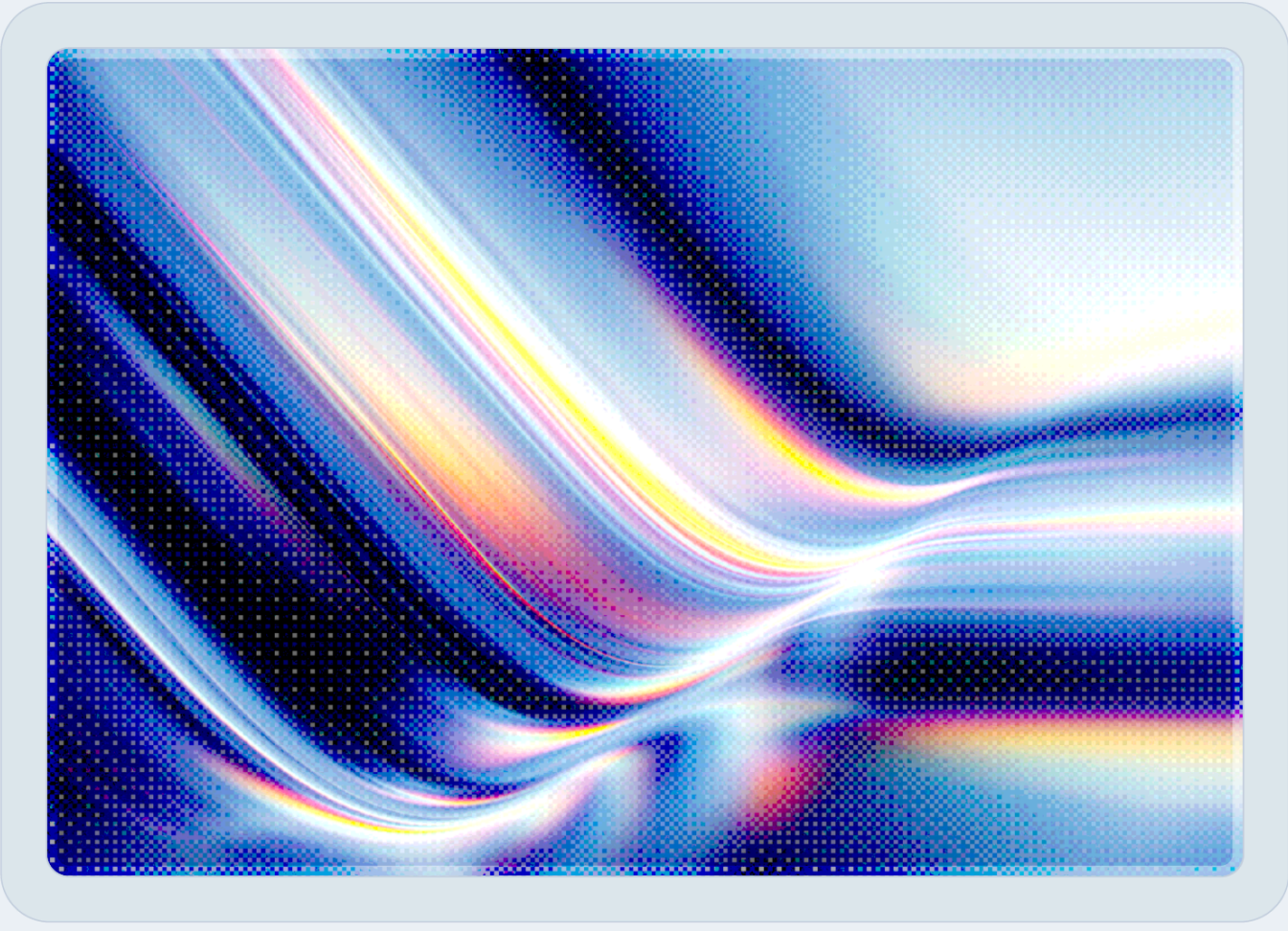
CSAI Foundation | Cloud Security Alliance

Claw Chain: Four CVEs Enable Full AI Agent Compromise

Critical Vulnerability Chain in OpenClaw Exposes 245,000 AI Agent Deployments

2026-05-17

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

- Cyera researchers disclosed four vulnerabilities in OpenClaw—collectively dubbed "Claw Chain"—that can be chained from a single foothold to achieve credential theft, privilege escalation, and persistent backdoor placement on the host system [1].
- The most severe flaw, CVE-2026-44112, carries a CVSS score of 9.6 (Critical) and exploits a time-of-check/time-of-use race condition in the OpenShell sandbox to redirect write operations outside the intended isolation boundary [2].
- As of May 2026, Shodan and ZoomEye scans identified approximately 65,000 and 180,000 publicly accessible OpenClaw instances, respectively—an exposure surface of roughly 245,000 servers reachable without any prerequisite internal access [3].
- All four vulnerabilities were responsibly disclosed to the OpenClaw maintainers in April 2026 and are fully patched in OpenClaw version 2026.4.22, released April 23, 2026 [1][2].
- Organizations running any OpenClaw version prior to 2026.4.22 should treat all credentials, API keys, and secrets reachable by their OpenClaw process as potentially compromised and initiate rotation. Organizations with evidence of sandbox access by untrusted code should treat that rotation as urgent.

Background

OpenClaw is an open-source autonomous AI agent platform that connects large language models directly to filesystems, SaaS applications, credential stores, messaging services, and execution environments. The project launched in November 2025 under the name Clawdbot, created by Austrian developer Peter Steinberger, and attracted 9,000 GitHub stars within its first 24 hours [4]. After two rapid renames—to Moltbook and then to OpenClaw in late January 2026—the project surpassed 214,000 GitHub stars by February 2026, an adoption rate among the fastest recorded for an open-source AI agent platform [4]. By early 2026, Steinberger had reduced his direct involvement with the project, which transitioned to community governance under an independent foundation.

The platform's design positions it as a local gateway process that routes incoming messages from supported communication channels—including WhatsApp, Telegram, Discord, Signal, Slack, Microsoft Teams, iMessage, Matrix, Google Chat, and others—through an LLM-powered agent capable of

executing commands, managing files, browsing the web, and handling email [4]. Enterprise deployments commonly integrate OpenClaw with Model Context Protocol (MCP) servers, extending the agent's reach to external services such as Google Calendar, Notion, Jira, Confluence, Stripe, and GitHub. This architectural posture—where the agent acts as an authenticated intermediary with broad access to sensitive services—makes OpenClaw deployments functionally equivalent to a privileged service account, and any compromise of the platform grants an attacker the same access as the credentials it holds.

The Claw Chain findings represent the first coordinated disclosure of vulnerabilities that move the threat from theoretical to demonstrably exploitable, with a documented attack chain that requires only a single point of entry into the agent's execution environment.

Security Analysis

The Four Vulnerabilities

The Claw Chain disclosure covers four distinct vulnerabilities spanning OpenClaw's sandbox isolation layer, command validation pipeline, and identity model. While each carries its own CVSS rating and exploitable impact, Cyera's research demonstrates that together they constitute an end-to-end attack chain requiring no special network position or elevated host privileges to initiate.

CVE-2026-44112 (CVSS 9.6 — Critical) is a time-of-check/time-of-use (TOCTOU) race condition in the write path of OpenClaw's OpenShell managed sandbox backend. The flaw allows an attacker who has obtained code execution inside the sandbox to race the filesystem validation step, redirecting write operations to targets outside the intended mount root before the sandbox enforces its boundary check. The practical consequence is that an attacker can overwrite host configuration files, inject content into persistent execution paths, or plant backdoors that survive OpenClaw restarts [1][2].

CVE-2026-44113 (CVSS 7.7 — High) mirrors CVE-2026-44112 in the read path. The same race-condition pattern allows an attacker to substitute a validated file path with a symbolic link pointing to an arbitrary location on the host before the read operation completes. Files, credentials, and internal artifacts that the sandbox was never intended to expose become accessible through this substitution, including secrets stored adjacent to—but outside—the agent's declared working directories [1][2].

CVE-2026-44115 (CVSS 8.8 — High) arises from an incomplete allowlist in OpenClaw's command validation layer. The platform uses an allowlist to restrict which commands agents may issue, but the validation logic does not account for shell expansion tokens embedded within here document (heredoc)

bodies. An attacker can craft a command that passes allowlist inspection at validation time yet expands to execute unapproved shell expressions at runtime, surfacing environment variables including API keys, OAuth tokens, and other credentials that OpenClaw holds in process memory [1][2].

CVE-2026-44118 (CVSS 7.8 — High) is an improper access control vulnerability in OpenClaw's MCP loopback transport. The platform accepts a client-controlled flag called `senderIsOwner` that signals whether the caller is authorized for owner-level tools, but validates this flag against the caller's assertion rather than against the authenticated session. A locally executing process with a valid bearer token can set `senderIsOwner` to true without challenge, elevating itself to owner-level privileges and thereby gaining control of gateway configuration, cron scheduling, and execution environment management [1][2].

The Claw Chain Attack Sequence

The power of these four vulnerabilities lies not in their individual severity but in how naturally they compose into an end-to-end attack. Cyera's analysis demonstrates a four-stage chain that begins from any foothold achievable through OpenClaw's normal operating surface—a malicious plugin, a prompt injection payload in untrusted external input, or a compromised MCP server—and terminates in full, persistent compromise of the host [1].

In the first stage, an attacker establishes code execution within the OpenShell sandbox. OpenClaw's plugin ecosystem and its acceptance of structured external inputs (webhook payloads, forwarded messages, MCP tool responses) provide multiple candidate entry points for this initial foothold. Deployments that do not extensively vet third-party plugins may treat sandbox containment as a sufficient control, given that it is a core assumption of OpenClaw's security model.

In the second stage, the attacker exploits CVE-2026-44113 and CVE-2026-44115 concurrently. The symlink-swap read flaw exposes secrets stored in files adjacent to the sandbox boundary, while the heredoc shell expansion flaw extracts credentials held in environment variables. Together, these two vulnerabilities give the attacker a working inventory of the API keys, tokens, and service credentials that the OpenClaw process has been granted by its operator.

In the third stage, the attacker exploits CVE-2026-44118 to elevate from sandbox-user level to owner level within the agent runtime. This pivot grants access to gateway configuration, the ability to modify cron-scheduled agent tasks, and control over the execution environment settings that govern all subsequent agent behavior.

In the fourth and final stage, the attacker weaponizes CVE-2026-44112 to write outside the sandbox boundary and plant persistent backdoors on the host. Because this write capability survives process restarts, the attacker's foothold persists even after the OpenClaw process is cycled—and because the compromised credentials harvested in stage two are typically long-lived OAuth tokens or API keys rather than session-bound artifacts, the stolen material remains useful indefinitely unless rotated.

Scale and Exposure

The combination of OpenClaw's rapid adoption and its design as a network-accessible service creates an unusually large exposure surface. Scan data from Shodan and ZoomEye as of May 2026 identifies approximately 245,000 publicly accessible OpenClaw instances [3]. Many of these appear to be personal or small-team deployments running on consumer-grade home networks or low-cost VPS instances; scan data does not confirm whether authentication controls are present at the perimeter, but the pattern is consistent with the platform's original consumer-facing positioning.

Enterprise deployments present a different but potentially more consequential risk profile. An OpenClaw instance integrated with corporate SaaS platforms, internal wikis, or ticketing systems via MCP may hold credentials that grant access far beyond the agent's own function—including OAuth tokens authorized by employees who may have limited visibility into how those tokens are subsequently used by the agent platform. Cyera's analysis treats this credential exposure as a first-order concern distinct from the technical severity of the vulnerabilities themselves [1].

Recommendations

Immediate Actions

Organizations should upgrade all OpenClaw deployments to version 2026.4.22 without delay. The April 23, 2026, release addresses all four vulnerabilities through GitHub Security Advisories GHSA-5h3g-6xhh-rg6p, GHSA-wppj-c6mr-83jj, GHSA-r6xh-pqhr-v4xh, and GHSA-x3h8-jrgh-p8jx [1]. In the updated release, the MCP loopback runtime issues separate bearer tokens for owner and non-owner sessions, and the `senderIsOwner` privilege determination is derived exclusively from which token authenticated the request rather than from a client-supplied header.

Alongside patching, organizations should treat all credentials, API keys, OAuth tokens, and environment variables accessible to any pre-patch OpenClaw process as potentially compromised and initiate rotation. This rotation should extend to any service account or delegated authorization that an

OpenClaw instance held—including integrations authorized through individual employee accounts—because the heredoc expansion flaw (CVE-2026-44115) could have surfaced these credentials to an attacker with sandbox access at any prior point.

Short-Term Mitigations

After patching and credential rotation, organizations should audit their OpenClaw deployment surface to identify any instances exposed to the public internet without authentication controls. Where public accessibility is not operationally required, instances should be placed behind network-layer access controls, a reverse proxy with authentication enforcement, or a VPN. Shodan queries targeting OpenClaw's default service port and banner characteristics may assist in identifying unknown external-facing deployments within organizational IP ranges.

Plugin provenance should be reviewed for all active OpenClaw deployments. Because the Claw Chain's first stage requires code execution within the sandbox—obtainable via a malicious plugin, a prompt injection payload, or a compromised MCP server—organizations should inventory installed plugins, verify each against its upstream source, and remove any that cannot be verified. Third-party plugins from community repositories that have not undergone security review should be treated with the same scrutiny applied to unvetted software packages in production environments.

Audit logs from OpenClaw deployments should be reviewed for anomalous patterns that may indicate prior exploitation. Indicators of compromise include unexpected file access outside the agent's configured working directories, environment variable reads executed through heredoc expansion, owner-level API calls originating from non-owner session contexts, and any filesystem modifications in paths the agent would not reach under normal operation.

Strategic Considerations

The Claw Chain findings illustrate a structural challenge that extends beyond any specific vulnerability. Agentic AI platforms in enterprise settings are often granted credentials equivalent to privileged service accounts, yet governance frameworks for such deployments remain immature in many organizations. OpenClaw, in particular, is frequently managed under practices designed for interactive user tools rather than for infrastructure-layer software with persistent credential access.

OpenClaw deployments that hold OAuth tokens for corporate SaaS applications, access keys for cloud infrastructure, or credentials for internal systems represent a class of privileged identity that many organizations may not yet have incorporated into their identity lifecycle management programs, given the recency of agentic platform adoption at enterprise scale. Organizations that operate or plan to operate agentic AI platforms should establish governance frameworks that apply the same lifecycle

controls to agent credentials as to human and service account credentials. This means defining an authorized scope of access for each deployment, requiring that credentials be issued with the minimum permission necessary to accomplish the agent's function, setting rotation schedules, and establishing a revocation path that can be exercised rapidly when a vulnerability disclosure or suspected compromise occurs. The Claw Chain disclosure is unlikely to be the last time a widely deployed agent platform's credential surface becomes the primary target of a multi-stage attack.

Enterprises evaluating agent platforms for operational use should require security assessments of the platform's sandbox isolation architecture, command validation logic, and identity model before deployment—particularly where the agent will hold access to regulated data or critical business systems. Treating agent platforms as infrastructure-layer software subject to the same procurement and security review standards applied to databases, messaging systems, and runtime environments reflects a risk posture commensurate with the access that broad-permission agent deployments hold.

CSA Resource Alignment

The Claw Chain vulnerabilities map directly to threat categories addressed by CSA's MAESTRO framework, which provides a seven-layer model for agentic AI threat analysis [5]. Under the MAESTRO framework, the sandbox escape vulnerabilities (CVE-2026-44112 and CVE-2026-44113) map to Layer 6, targeting the deployment infrastructure that is expected to enforce execution boundaries. The heredoc bypass (CVE-2026-44115) aligns with Layer 3—a failure within the agent framework's own input validation logic. The privilege escalation flaw (CVE-2026-44118) corresponds to Layer 4, exploiting a weakness in the agent communication protocol's identity and session model. The Claw Chain's multi-layer traversal is consistent with MAESTRO's prediction that attacks on agentic systems will often cross architectural layer boundaries: each vulnerability affects a distinct layer, and the chain traverses multiple layers to achieve its full impact.

CSA's AI Controls Matrix (AICM) provides complementary control guidance applicable to organizations operating OpenClaw and similar agentic platforms. AICM control domains covering access management, identity lifecycle, and runtime environment isolation are directly implicated by the Claw Chain findings. In particular, the AICM's guidance on least-privilege access provisioning for AI agents and on credential management for agentic workflows addresses the systemic conditions that amplified the Claw Chain's potential impact.

CSA's Zero Trust guidance emphasizes that no component of an agentic system should be trusted based on network location or prior authorization alone—a principle violated by OpenClaw's pre-patch `senderIsOwner` model, which accepted the caller's self-reported privilege level rather than

validating it against an independently authenticated session. Applying zero trust principles to agent runtime identity—where every privilege claim is verified against an authoritative source rather than a caller-supplied flag—would likely have mitigated CVE-2026-44118.

CSA's published analysis of applying MAESTRO to real-world agentic AI threat models demonstrates how the framework's layered approach can be integrated into automated security testing pipelines, including CI/CD integration for ongoing threat model validation [6]. Organizations that conduct adversarial testing against their OpenClaw deployments should incorporate sandbox boundary testing, command validation probing, and identity model review as explicit objectives, applying the multi-layer lens MAESTRO provides.

References

[1] Cyera Research. "Claw Chain: Cyera Research Unveil Four Chainable Vulnerabilities in OpenClaw." Cyera Blog, May 2026.

[2] The Hacker News. "Four OpenClaw Flaws Enable Data Theft, Privilege Escalation, and Persistence." The Hacker News, May 2026.

[3] Cybersecurity News. "OpenClaw Chain Vulnerabilities Expose 245,000 Public AI Agent Servers to Attack." Cybersecurity News, May 2026.

[4] GitHub. "openclaw/openclaw: Your own personal AI assistant. Any OS. Any Platform.." GitHub, 2026.

[5] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 2025.

[6] Cloud Security Alliance. "Applying MAESTRO to Real-World Agentic AI Threat Models: From Framework to CI/CD Pipeline." CSA Blog, February 2026.