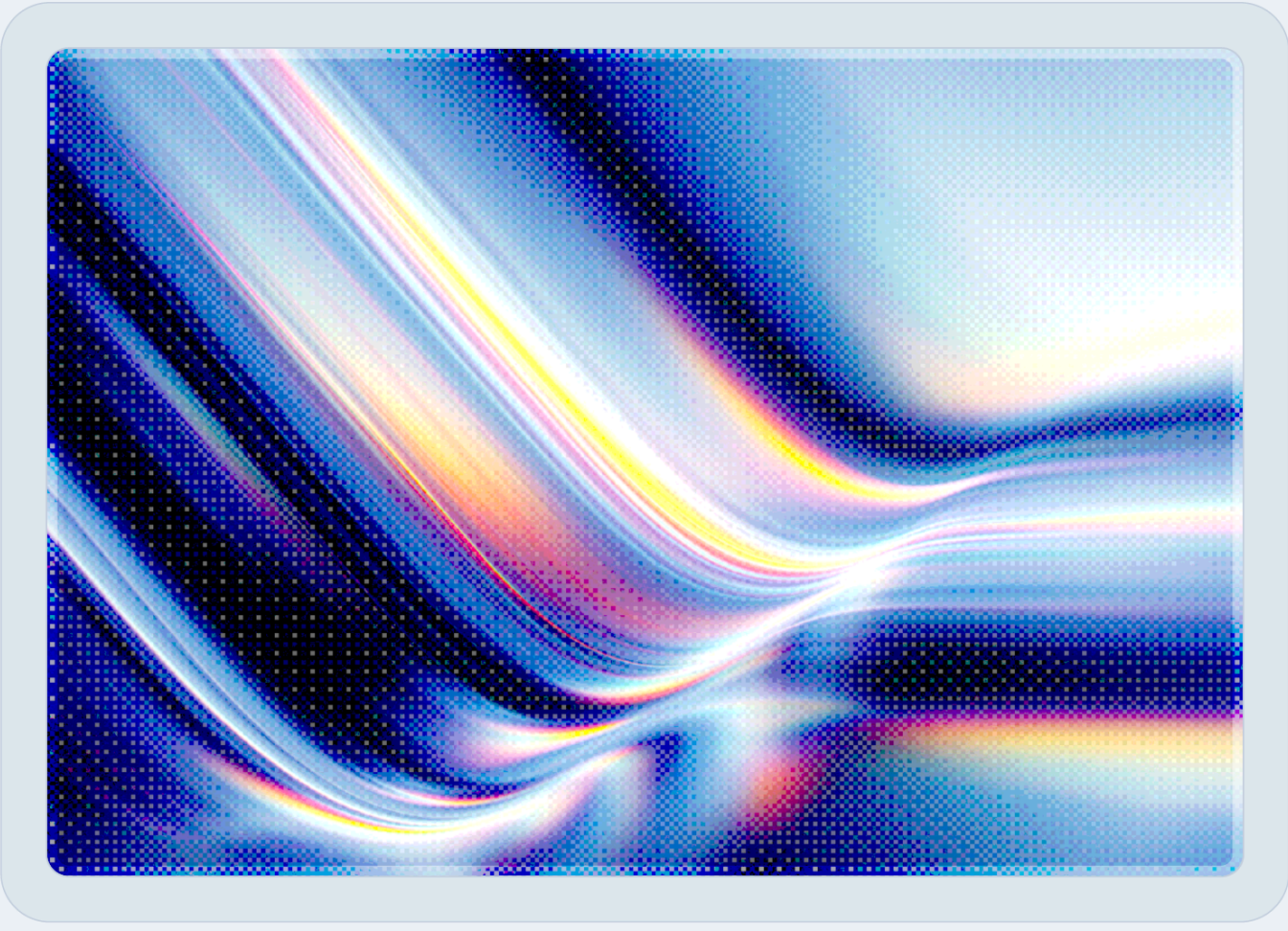
CSAI Foundation | Cloud Security Alliance

Claw Chain: Four Chained CVEs Compromise AI Agents

Sandbox Escape, Credential Theft, and Persistent Backdoors in OpenClaw

2026-05-18

AI-assisted Rapid Research



CSAI

cloud security alliance®

Key Takeaways

- Four vulnerabilities in OpenClaw's sandbox and MCP loopback runtime — CVE-2026-44112, CVE-2026-44113, CVE-2026-44115, and CVE-2026-44118 — can be chained by an attacker to progress from sandbox code execution to host-level persistence without triggering conventional security alerts.
- The highest-severity flaw, CVE-2026-44112, carries a CVSS score of 9.6 (Critical) and allows an attacker to redirect filesystem writes outside OpenClaw's sandbox boundary, enabling configuration tampering and backdoor installation on the underlying host.
- Shodan and ZoomEye scans conducted in May 2026 identified approximately 245,000 publicly accessible OpenClaw instances — roughly 65,000 via Shodan and 180,000 via ZoomEye — creating an unusually wide attack surface for what is ostensibly an agent-local sandbox component [1][2].
- All four vulnerabilities were disclosed to OpenClaw maintainers on April 22, 2026, and patches were released the following day. Organizations running any OpenClaw release prior to version 2026.4.22 should treat their deployments as unpatched and act accordingly.
- The Claw Chain attack pattern is notable for what it does not look like: each exploitation step mimics normal agent behavior, making detection difficult using conventional host-level monitoring controls alone, without monitoring purpose-built for agentic AI behaviors.

Background

OpenClaw is an open-source autonomous AI agent platform with broad adoption in enterprise and developer communities. It provides a managed sandbox environment — OpenShell — designed to isolate agent-initiated code execution from the underlying host filesystem, and exposes a Model Context Protocol (MCP) loopback runtime that allows local processes to interact with the agent's gateway and scheduling subsystems. The premise of this architecture is that the sandbox acts as a hard boundary: agent actions remain contained even if the agent's reasoning or inputs are compromised.

The "Claw Chain" research, disclosed publicly on May 18, 2026 by Cyera's security research team, challenges that premise directly [1][3]. By chaining four distinct vulnerabilities spanning the sandbox, its command validation layer, and the MCP loopback runtime, an attacker with an initial foothold inside the

sandbox can systematically work outward: stealing credentials and sensitive files, escalating to owner-level control over the agent gateway, and ultimately planting a persistent backdoor on the host. The four vulnerabilities are tracked as CVE-2026-44112, CVE-2026-44113, CVE-2026-44115, and CVE-2026-44118, with associated GitHub Security Advisories GHSA-5h3g-6xhh-rg6p, GHSA-wppj-c6mr-83jj, GHSA-r6xh-pqhr-v4xh, and GHSA-x3h8-jrgh-p8jx respectively [1].

The discovery arrives against a backdrop of an already-documented threat landscape for OpenClaw. Prior research identified authentication weaknesses across publicly accessible instances, malicious skill distribution through the ClawHub marketplace, and companion infrastructure vulnerabilities including the earlier CVE-2026-25253 one-click RCE chain. Claw Chain builds on this history: it does not require an external network connection or a publicly exposed endpoint to trigger. Rather, it begins inside the sandbox itself, turning the agent's own privileges and trust relationships into the attacker's primary tool.

Security Analysis

The Four Vulnerabilities

The Claw Chain exploit chain relies on four individually significant vulnerabilities that together form a complete attack progression from sandboxed code execution to persistent host compromise.

CVE-2026-44112 (CVSS 9.6, Critical) is a time-of-check/time-of-use (TOCTOU) race condition in OpenClaw's OpenShell sandbox. The vulnerability affects filesystem write operations: when OpenShell validates a file path before permitting a write, there exists a brief window during which an attacker-controlled process can substitute the validated path with a symbolic link pointing to an arbitrary location outside the sandbox's mount root. Exploiting this race allows writes to proceed to host filesystem locations — including configuration directories and startup paths — that the sandbox was intended to isolate [1][2][3]. The consequence is the ability to plant files, including backdoors, on the host outside of any sandbox constraint.

CVE-2026-44113 (CVSS 7.7, High) is the read-side counterpart to CVE-2026-44112. The same TOCTOU pattern applies to read operations: an attacker can swap a validated file path with a symbolic link to a sensitive location outside the allowed mount root before the read completes. This enables exfiltration of host system files, OpenClaw configuration artifacts, environment variables, authentication tokens, and conversation histories that the agent process legitimately holds access to — but which should never be readable from within the sandbox [1][2].

CVE-2026-44115 (CVSS 8.8, High) targets OpenClaw's command execution allowlist. The platform validates submitted commands against an allowlist before permitting them to run. However, the validation does not account for shell environment variable expansion inside heredoc bodies. An attacker can embed shell expansion tokens within a heredoc block, causing sensitive values — including API keys, bearer tokens, and credentials present in the agent's environment — to be expanded and returned through a command that passes the allowlist check at validation time. The result is credential exfiltration through what appears, to the validation layer, to be a safe and permitted command [1][2][4].

CVE-2026-44118 (CVSS 7.8, High) is an improper access control vulnerability in the MCP loopback runtime. OpenClaw's MCP gateway accepts a client-controlled flag, `senderIsOwner`, and acts on it without cross-referencing the authenticated session to confirm the claim. A local process holding any valid bearer token can set this flag to true, immediately gaining owner-level access over the gateway's configuration, scheduling, and execution management interfaces. From owner position, an attacker can reconfigure agent behavior, redirect outputs, and establish persistence that survives agent restarts [1][2].

The Attack Chain

These four vulnerabilities do not need to be exploited independently. Cyera researchers demonstrated a sequential attack progression — referred to as "Claw Chain" — in which each step exploits a different vulnerability and uses its output to enable the next [1][3].

The chain begins with an initial foothold: a malicious plugin, a compromised external input, or a prompt injection that achieves code execution within the OpenClaw sandbox. This initial access does not require any of the four CVEs; it exploits the agent's own execution capabilities, which are intentionally broad. From the sandbox, the attacker applies CVE-2026-44113 and CVE-2026-44115 to harvest credentials and sensitive files from the host and agent environment. Armed with a valid bearer token obtained through this exfiltration step, the attacker then exploits CVE-2026-44118 to assert owner-level control over the MCP loopback gateway. Owner access enables persistent reconfiguration of the agent's behavior and lays the groundwork for the final step: exploiting CVE-2026-44112 to redirect a sandbox write operation to the host filesystem, installing a backdoor that persists beyond the current agent session.

What makes this progression operationally significant is how well it blends with legitimate agent activity. As Cyera's researchers noted, "each step looks like normal agent behaviour to traditional security controls" [1][5]. The agent possesses legitimate authority to read files, expand environment variables, call the MCP gateway, and write within its permitted scope. Claw Chain exploits the margins of that authority — the race window, the validation gap, the unvalidated flag — while producing actions that fall within the visible envelope of normal operation.

Exposure and Context

The Claw Chain vulnerabilities affect all OpenClaw releases prior to version 2026.4.22. The scale of exposure identified by Cyera is consequential: approximately 245,000 publicly accessible OpenClaw instances — roughly 65,000 identified via Shodan and 180,000 via ZoomEye — were found via internet scanning in May 2026 [1][2]. Enterprise environments using OpenClaw for automated development, data analysis, or workflow automation are at particular risk because deployed agents often require broad access to internal systems, credentials, and data stores to perform their functions. Financial services, healthcare, and legal organizations processing sensitive data through OpenClaw agents face compounded risk given the volume and sensitivity of the information within reach of a compromised agent process [2].

It is worth noting that Claw Chain extends the already-documented threat landscape for OpenClaw. Earlier research established authentication weaknesses in publicly reachable OpenClaw instances, that the ClawHub skill marketplace has carried malicious plugins, and that companion infrastructure such as Moltbook has suffered its own credential exposure incidents. Claw Chain represents a qualitatively different threat: unlike attacks that exploit missing authentication or rogue plugins, it targets the integrity of OpenClaw's core sandboxing mechanism — the architectural control that is supposed to make other risks manageable.

Recommendations

Immediate Actions

Organizations running OpenClaw should treat the patch boundary at version 2026.4.22 as a hard line. Any instance running an earlier release should be considered vulnerable to all four CVEs in the Claw Chain and should be updated or isolated before resuming production use. Given that Cyera publicly disclosed the full technical details of the attack chain on May 18, 2026, organizations should not assume a substantial grace period before exploitation attempts begin.

Beyond patching, organizations should audit which users and processes have access to OpenClaw's MCP loopback interface. CVE-2026-44118's exploitation requires only a valid bearer token and local access — not elevated privileges — so any process running in the same host context as the OpenClaw agent should be considered a potential source of privilege escalation if that process is compromised. Restricting MCP

loopback access to trusted processes and enforcing session-bound authentication are immediate configuration hardening steps that reduce the exploitability of this vulnerability even before a patch is applied.

Short-Term Mitigations

The TOCTOU race conditions in CVE-2026-44112 and CVE-2026-44113 arise from a fundamental design pattern in OpenShell: path validation and file I/O are non-atomic operations. Organizations should review whether filesystem activity logging on the host can detect the pattern of rapid path substitution that characterizes TOCTOU exploitation. Monitoring for unexpected symbolic link creation in or near the OpenShell mount root, particularly during active agent task execution, provides a behavioral signal that does not depend on signature-based detection.

For CVE-2026-44115, organizations can reduce exposure by auditing the environment variables available to the OpenClaw process. Sensitive credentials that can be removed from the agent's environment without breaking its function should be removed; credential injection at task invocation time — rather than persistent environment presence — limits the window during which heredoc expansion can expose them. Secrets management integration, if not already deployed, is a higher-order mitigation that addresses multiple OpenClaw credential exposure vectors simultaneously.

Organizations that have not already done so should evaluate their deployment posture against the broader OpenClaw threat landscape documented in prior research: default authentication settings, internet exposure of control panels, and plugin provenance. Claw Chain does not change that landscape but amplifies its consequences: a sandbox escape capability transforms any of those existing footholds into a path to full host compromise.

Strategic Considerations

The Claw Chain disclosure illustrates a structural challenge in agentic AI security: sandboxing is necessary but not sufficient as an isolation guarantee when the sandbox runtime is complex software with its own vulnerability surface. The TOCTOU vulnerabilities in OpenShell are not exotic; race conditions in filesystem-level isolation are a well-understood class of bugs, and their appearance in an AI agent sandbox underscores that the security properties claimed by agent platforms must be evaluated with the same rigor applied to any system software.

Organizations building AI agent programs should define a minimum security baseline for agent runtimes that includes: (1) documented, independently verified isolation boundaries, (2) vendor commitment to timely vulnerability disclosure and patching, (3) separation of agent execution from sensitive credential stores, and (4) behavioral monitoring capability that can detect deviation from normal agent activity

patterns. OpenClaw's twenty-four-hour patch turnaround following Cyera's April 22 disclosure demonstrates responsive vulnerability management on the maintainers' part, but the existence of these vulnerabilities in a platform processing sensitive enterprise data reflects how security properties of agent runtimes have not yet received the sustained scrutiny applied to more established system software.

The pattern of chaining an initial low-privilege foothold through a sequence of individually-rated-High vulnerabilities to achieve Critical outcomes is itself a strategic concern. Each of the four CVEs in Claw Chain would be a manageable risk in isolation. Their combination produces a complete compromise chain. Security teams evaluating agentic AI platforms should assess not only the individual severity ratings of known vulnerabilities but the degree to which they compose — whether an attacker who gains a minor advantage can use it as a stepping stone toward full control.

CSA Resource Alignment

The Claw Chain vulnerabilities map directly to several CSA frameworks that provide guidance for organizations evaluating and operating agentic AI platforms.

CSA's MAESTRO framework for agentic AI threat modeling identifies sandbox escape and privilege escalation as core threat categories for autonomous agent systems [10]. The Claw Chain attack chain — moving from sandboxed code execution through credential theft, privilege escalation, and persistence — traverses each layer of the MAESTRO threat model in sequence. Organizations conducting MAESTRO-based threat assessments of OpenClaw deployments should treat all four CVEs as confirmed realizations of sandbox integrity threats, not merely theoretical risks.

The AI Controls Matrix (AICM) [7] provides control guidance applicable to each phase of the Claw Chain. AICM controls addressing execution environment isolation are directly relevant to CVE-2026-44112 and CVE-2026-44113, where the gap between validation and execution creates the exploitable race. Credential and secret management controls apply to CVE-2026-44115. Access control and session management controls govern the trust model that CVE-2026-44118 violates. Organizations using AICM as an audit framework should map these CVEs to the relevant control domains and assess whether their current control implementations would have detected or prevented exploitation. AICM is a superset of CCM, making it the appropriate lens for agentic AI deployments rather than CCM alone.

CSA's Agentic AI Red Teaming Guide [6] provides practical test procedures for the attack patterns demonstrated in Claw Chain. Specifically, its guidance on sandbox integrity testing, filesystem boundary validation, and privilege escalation through inter-process communication is directly applicable to

identifying TOCTOU and MCP loopback vulnerabilities in locally deployed agent runtimes. Organizations that have not yet applied the Red Teaming Guide to their OpenClaw deployments should treat the Claw Chain disclosure as a prompt to do so.

The STAR program [9] provides a mechanism for evaluating the security posture of AI agent platform providers against documented controls. OpenClaw's vulnerability history — including Claw Chain, CVE-2026-25253, and the broader exposures documented in prior research — illustrates why STAR-based or equivalent third-party security assessments of agentic AI platforms provide meaningful assurance. Organizations considering enterprise deployment of any agentic AI platform should require evidence of independent security assessment as part of vendor qualification.

CSA's AI Organizational Responsibilities publications [8] address the governance dimension that Claw Chain highlights. Unauthorized or ungoverned use of OpenClaw within an enterprise dramatically expands the blast radius of vulnerabilities like those in Claw Chain. Shadow AI deployments that are not tracked, patched, or monitored are unpatched by definition, and a 245,000-instance exposure window suggests that a significant fraction of vulnerable deployments are in exactly this category.

References

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

[2] CybersecurityNews. "OpenClaw Chain Vulnerabilities Expose 245,000 Public AI Agent Servers to Attack." CybersecurityNews, May 2026.

[3] SecurityWeek. "'Claw Chain' OpenClaw Flaws Allow Sandbox Escape, Backdoor Delivery." SecurityWeek, May 18, 2026.

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

[5] The Next Web. "Four OpenClaw flaws let attackers steal data, escalate privileges, and plant backdoors through the agent's own sandbox." The Next Web, May 2026.

[6] Cloud Security Alliance. "Agentic AI Red Teaming Guide." CSA, 2025.

[7] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA, 2025.

[8] Cloud Security Alliance. "AI Organizational Responsibilities: Core Security Responsibilities." CSA AI Safety Initiative, 2025.

[9] Cloud Security Alliance. "STAR for AI." CSA Security Trust Assurance and Risk Program, 2025.

[10] Cloud Security Alliance. "MAESTRO: Agentic AI Threat Modeling Framework – OpenClaw Analysis." CSA Blog, February 2026.