STRIKEREADY

SRB2501: High Priority

OpenClaw's Autonomous AI as an Attack Surface

Release Date: February 03, 2026

Executive Summary:

OpenClaw (formerly Clawdbot and Moltbot) is an open-source, locally hosted agentic AI system that grants large language models direct shell access, browser automation, and cross-platform orchestration for system-level execution. While its rapid evolution highlights the breakthrough potential of autonomous AI, its meteoric rise has outpaced the maturity of its security controls, introducing significant risk when deployed without rigorous security measures.

As of February 2026, a high-severity vulnerability has been identified in OpenClaw versions prior to 2026.1.29. The Control UI fails to validate gatewayUrl parameters, allowing attackers to execute a "One-Click" Remote Code Execution (RCE) attack. By tricking a user into clicking a crafted link, an attacker can force the agent to 'phone home' to a malicious server, silently exfiltrating authentication tokens and plaintext API keys, effectively granting the attacker full administrative control over the host machine. The OpenClaw ecosystem relies on modular "Skills" (plugins) that execute with high-level system privileges. Adversaries have exploited the project's identity fragmentation to flood registries with malicious skills. Over 230 unverified skills have been identified as delivery vehicles for persistent malware (such as ScreenConnect RATs) and credential stealers that target local environments and developer workflows.

Threat Description:

Vulnerability Details

One-Click Remote Code Execution (CVE-2026-25253)

A high-severity vulnerability in OpenClaw versions prior to 2026.1.29 allows attackers to hijack an agent through a single malicious link. The application automatically connects to attacker-controlled gateways via URL parameters, leaking authentication tokens and integrated API keys in plaintext. Stolen tokens enable attackers to reconnect to the victim's local agent gateway and execute arbitrary shell commands, resulting in full host compromise.

Allowlist Bypass via Environment Poisoning (CVE-2026-22708)

A critical weakness in allowlist-based command execution enables attackers to bypass security controls using built-in shell commands. Because built-ins such as export, set, or alias execute within the shell session rather than as external binaries, they are often excluded from command validation logic. Attackers can exploit this gap through indirect prompt injection to poison the execution environment, for example, by modifying the PATH variable. Subsequent execution of "allowlisted" commands (e.g., git) then resolves to attacker-controlled binaries. OpenClaw's Skills ecosystem and auto-run workflows are vulnerable to this technique, allowing trusted tools to be silently hijacked despite restrictive command policies.

Supply-Chain Compromise via Malicious VS Code Extension

OpenClaw relies on a modular Skills ecosystem that executes with high-level system privileges. Adversaries have exploited identity fragmentation and weak verification controls to distribute malicious or trojanized skills. A notable example is the Trojanized "ClawdBot Agent" VS Code extension, which executes automatically on IDE startup and deploys a legitimate remote management client pre-configured to attacker infrastructure, alongside multiple fallback payloads. These supply-chain attacks enable persistent access, credential theft, and remote administration without exploiting OpenClaw core binaries.

AI-Specific Threat Vectors: Prompt Injection and Memory Poisoning

OpenClaw's autonomous operation and persistent memory introduce novel attack vectors. Attackers can embed hidden instructions in web pages or messages that the agent processes, causing unintended actions such as data exfiltration or file modification. Additionally, poisoning the agent's long-term memory can permanently alter behavior, weaken safety controls, and establish stealthy persistence without traditional malware indicators.

Indicators of Compromise (IOCs):

OpenClaw users and security teams should actively hunt for:

Command-and-Control (C2) Domains

- `meeting.bulletmailer[.]ne` (Port 8041)
- `clawdbot.getintwopc[.]site`
- `darkgptprivate[.]com`

IP Addresses

- `179.43.176[.]32`
- `178.16.54[.]253`
- (Seychelles-hosted infrastructure)

Network Indicators

- Outbound WebSocket connections on TCP port 18789 to non-local IP addresses
- Unexpected external connections initiated immediately after clicking OpenClaw-related URLs

Malicious Files (SHA-256)

- `Code.exe`
- `DWrite.dll`
- `e20b920c7af988aa215c95bbaa365d005dd673544ab7e3577b60fecf11dcdea2`
- `d1e0c26774cb8beabaf64f119652719f673fb530368d5b2166178191ad5fcbea`

File Paths

- `%TEMP%\Lightshot\` (staging directory)
- `C:\Program Files (x86)\ScreenConnect Client (083e4d30c7ea44f7)\`

Persistence Indicators

- New Windows services or startup entries referencing `Code.exe` in `%TEMP%`
- Scheduled tasks or cron jobs created without explicit user approval

Behavioral Indicators

- Unexpected use of shell built-ins (`export, alias, unset`) during agent sessions
- Environment variable drift affecting `PATH, LD_PRELOAD,` or `PYTHONPATH`
- Execution of trusted commands from temporary directories (`/tmp, /var/tmp, %TEMP%`)

Mitigation and Remediation:

Organizations should treat OpenClaw as high-risk software and respond accordingly:

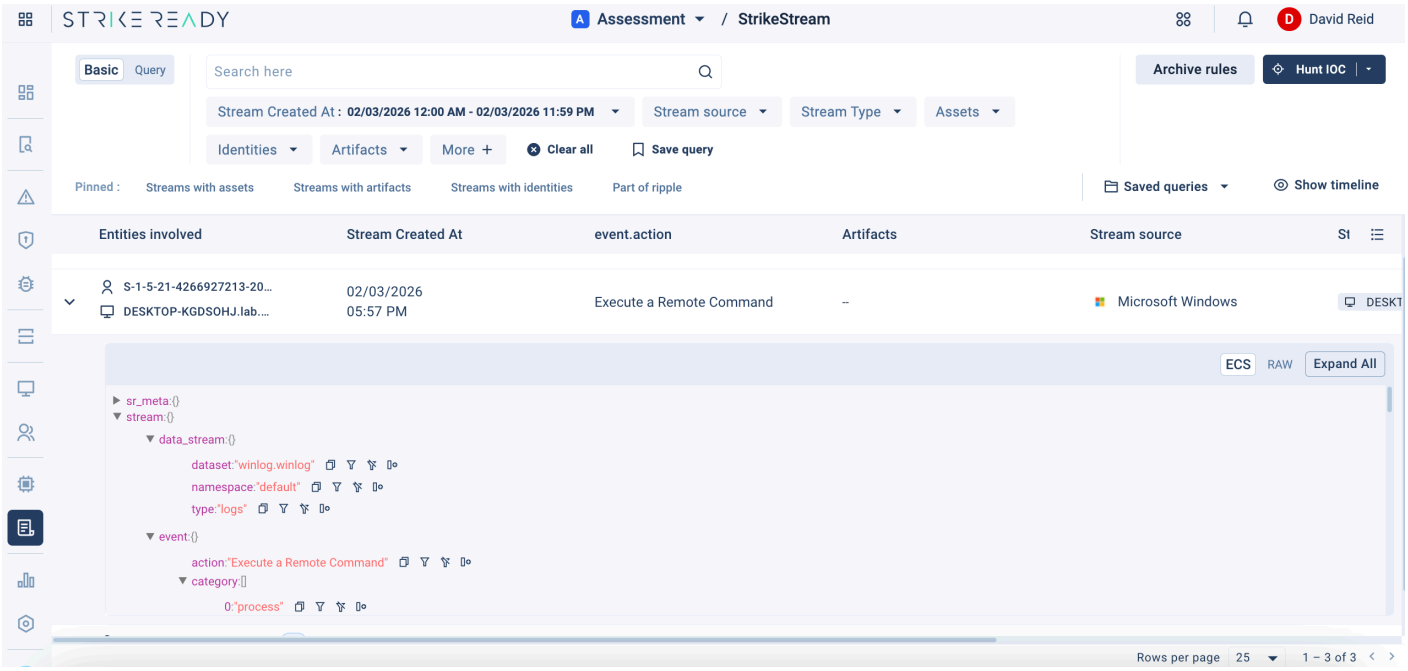
- Upgrade to OpenClaw v2026.1.29 or later, which fixes CVE-2026-25253 (One-Click RCE) and enforces gateway URL validation. Apply all official patches and security updates immediately.
- Disable unattended execution for browser actions, shell commands, email access, and skill execution. All state-changing actions must require explicit user approval.
- Run OpenClaw inside a dedicated VM or container with no access to host credentials, read-only filesystem where possible, and developer SSH keys or cloud credentials.
- Treat all Skills as untrusted code with malicious potential. Disable their auto-installation and regularly audit installed Skills, removing unverified or outdated plugins.
- Disable browser automation by default unless strictly required. Restrict automation to domain allowlists; block email, OAuth, and file upload access.
- Rotate all API keys (OpenAI, Anthropic, Slack, GitHub) and authentication tokens if running versions prior to v2026.1.29. Avoid storing sensitive credentials in agent-accessible files.
- Remove unverified VS Code extensions or third-party plugins. Only install extensions from official sources.
- Audit and sanitize persistent memory files (~/.openclaw/MEMORY.md) before use. Disable automatic ingestion of untrusted web content or email data without review.
- Restrict TCP port 18789 to localhost and monitor outbound WebSocket traffic to prevent unauthorized access.

Explore with StrikeReady:

Here's how users can harden their security when using OpenClaw using StrikeReady.

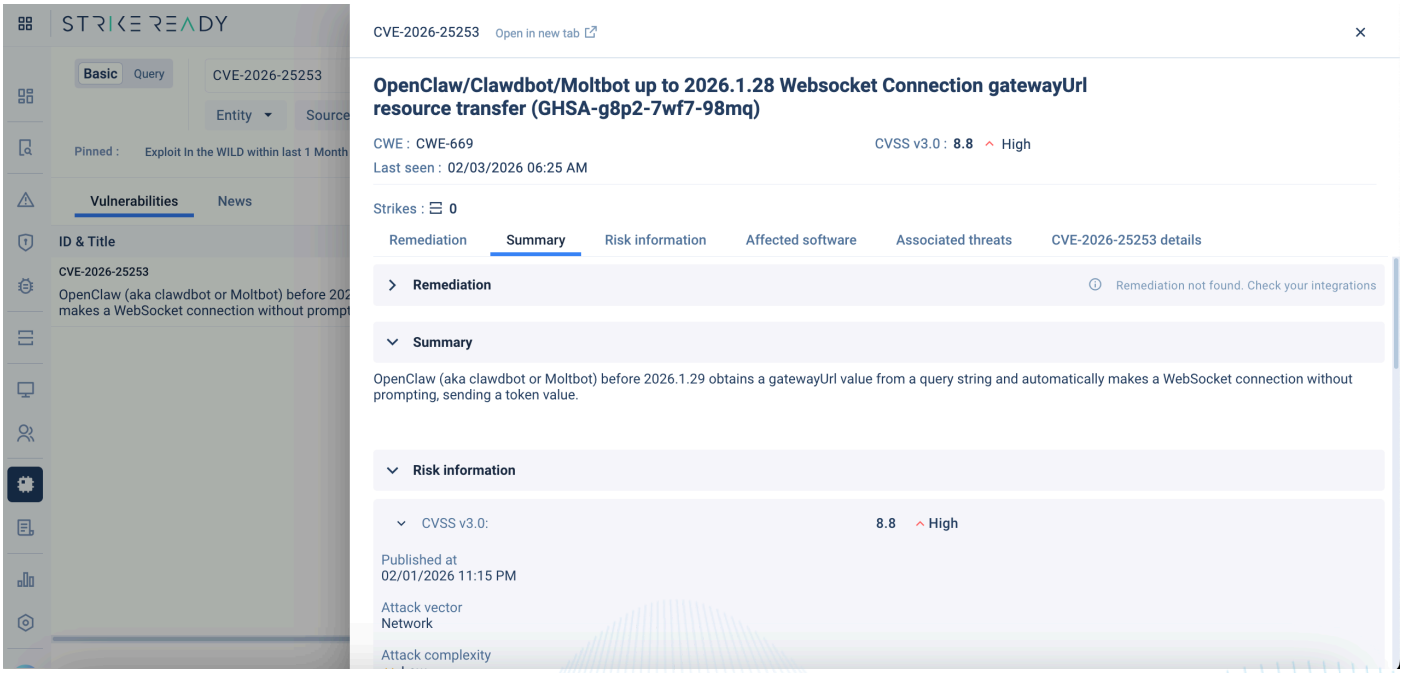
StrikeStream:

1. Login to StrikeReady product at https://portal.strikeready.app/v2/
2. Select "StrikeStream" available on the left sidebar.
3. Search for streams on Remote Command Execution.



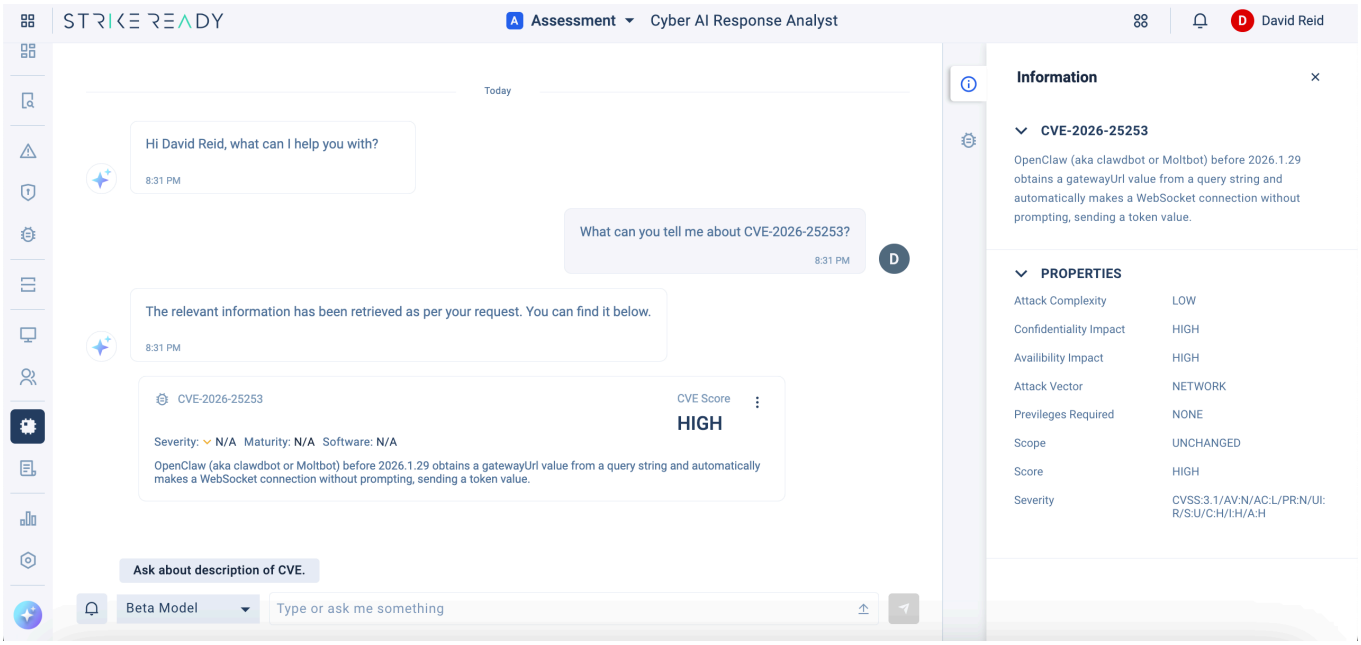
Recon:

1. Select "Recon" available on the left sidebar.
2. Search for above-cited vulnerabilities associated with OpenClaw for a detailed overview.



CARA:

1. Navigate to our security assistant, CARA (Cyber AI Response Analyst), on the bottom left corner.
2. There, you can ask CARA about OpenClaw, it's associated vulnerabilities, and more:



StrikeReady Research is working day and night to keep its customers updated as events regarding OpenClaw unfold.