OpenClaw Security Guide: Risks and Hardening

February 2, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** OpenClaw is a genuinely impressive AI agent — it can manage your email, book flights, write code, and automate workflows through WhatsApp and Telegram. It's also a security nightmare. Researchers have found 42,000+ exposed instances, critical authentication bypasses, prompt injection attacks that extract credentials in minutes, and a skill marketplace with zero moderation where 26% of analyzed skills contained vulnerabilities. The localhost auth bypass has been patched, but the architectural problems — broad permissions, unsolved prompt injection, unaudited plugins — remain. If you understand VPS deployments, network isolation, and credential rotation, you can experiment carefully on dedicated hardware with throwaway accounts. If any of that sounds unfamiliar, wait for enterprise-grade alternatives that are months away. Do not connect OpenClaw to financial accounts, health data, or client systems under any circumstances.

📚 **More on this topic:** OpenClaw Setup Guide · Best Models for OpenClaw · OpenClaw vs Commercial Agents · Local LLMs vs ChatGPT

Disclaimer: This guide is educational. It documents publicly known security issues and community-recommended mitigations for OpenClaw. Following these steps reduces risk but does not eliminate it. No setup is "perfectly secure" — OpenClaw's own documentation says as much. You assume all risk when running agentic AI software. This is not a certification of safety.

OpenClaw (formerly Clawdbot, then Moltbot) is the fastest-growing open-source project in GitHub history. Over 145,000 stars. 20,000+ forks. Cloudflare's stock jumped 20% because OpenClaw recommends their tunneling service. Andrej Karpathy praised it publicly.

It's also been called "an absolute nightmare" by Cisco's AI security research team, a "lethal trifecta" of risk by Palo Alto Networks, and "info stealer malware in disguise" by Google's VP of security engineering.

Both things are true at once. This guide helps you understand the risks before you decide whether to run it.

What OpenClaw Actually Does

Strip away the hype and OpenClaw is three things:

1. **A local gateway** — a service running on your machine that maintains WebSocket connections to messaging platforms (WhatsApp, Telegram, Signal, iMessage). Your conversation history and credentials stay local.

2. **An LLM backend** — typically Claude (via Anthropic's API), but also GPT-4, or local models through Ollama. You own the agent layer. You rent the intelligence.

3. **A skill/plugin system** — 50+ bundled skills plus ClawHub, a community marketplace. Skills give the agent hands: browser automation, file system access, shell commands, calendar integration, email management.

The result is an AI assistant you message on WhatsApp that can triage your inbox, draft emails in your voice, book flights, commit code to your repos, monitor prices, plan meals, and remember everything across sessions.

The tagline is "AI that actually does things." That's the value proposition and the core risk in five words.

What "local-first" actually means

Local-first means the gateway runs on your machine. Your data stays on your machine. But unless you're running a local model through Ollama, your prompts and conversations still travel to Anthropic or OpenAI's APIs for inference. You control the agent layer. The intelligence comes from the cloud.

The Security Model

OpenClaw's security model is opt-in, not built-in. The project's own documentation states: **"There is no 'perfectly secure' setup."**

Here's how the security architecture works in practice:

Component	Default State	Risk
Gateway authentication	Localhost trust patched; loopback WebSocket hardened (v2026.3.2)	Improved, but misconfiguration still possible

Component	Default State	Risk
Credential storage	SecretRef support across 64 targets (v2026.3.2); plaintext fallback still exists	Better — unresolved refs now fail fast on active surfaces
Skill execution	All downloaded code treated as trusted	Supply chain attacks trivial
Sandboxing	Opt-in, not default. New installs default to `messaging` tool profile (v2026.3.2)	Narrower initial scope, but agent still has host access unless you configure otherwise
Prompt injection defense	Model-dependent	No reliable solution exists
Network exposure	Depends on your setup	Misconfiguration = public exposure
Filesystem	Symlink escape protection added (v2026.3.2)	Reduces workspace breakout risk

The fundamental tension: a useful agent needs broad permissions. Broad permissions create a massive attack surface. OpenClaw's creator Peter Steinberger has acknowledged this openly. The project has shipped meaningful security improvements — v2026.3.2 alone included Gateway loopback hardening, SecretRef credential management, symlink escape protection, and a tighter default tool profile. But the architectural bind remains.

As security researcher Jameson O'Reilly (DVULN) put it: "We've spent 20 years building security boundaries around our systems. Agents require us to tear that down. The value proposition requires punching holes through every boundary that security teams took decades to build."

Known Vulnerabilities

These are documented, publicly disclosed issues. Some have been patched. The architectural ones haven't, because they can't be — they're inherent to how agentic AI works.

Localhost Authentication Bypass

Discovered by: Jameson O'Reilly, DVULN **Status:** Patched

The gateway's authentication logic trusted all localhost connections by default. If you ran OpenClaw behind a reverse proxy (a common deployment pattern), proxy traffic was treated as local — no authentication required. Full access to credentials, conversation history, and command execution.

O'Reilly scanned the internet and found **hundreds of exposed instances**. Of those examined manually, at least eight were completely open: API keys visible, Telegram bot tokens exposed, one instance had Signal configured on a public server.

A broader scan by security firm **Slowmist** found that an authentication bypass made **several hundred API keys and private conversation histories** accessible to anyone who knew where to look.

A separate analysis found **42,000+ publicly exposed instances**, with 5,194 actively verified as vulnerable. Of those, **93.4% had critical authentication bypass vulnerabilities** enabling unauthenticated access to the gateway control plane, with potential for remote code execution.

Prompt Injection via Email/Messages

Demonstrated by: Matt Vukoule **Status:** Unsolved (architectural)

Vukoule sent a single crafted email to an OpenClaw instance with email integration enabled. Using prompt injection — hidden instructions embedded in the email content — he extracted a private key and gained control **in under 5 minutes**.

This works because LLMs cannot reliably distinguish instructions from content. If OpenClaw reads an incoming WhatsApp message, email, or any external text that contains hidden instructions, the model may follow them. It could forward your credentials, execute a shell command, or exfiltrate data — and you'd never see it happen.

No one has solved prompt injection. OpenClaw's docs recommend using Anthropic's Claude Opus 4.5 because it's "quite good at recognizing prompt injections," but "quite good" is not "reliable." Smaller or cheaper models are significantly more vulnerable. The Qwen 3.5 family (Feb-Mar 2026) shows improved instruction following and reasoning at smaller sizes — the 9B beats GPT-OSS-120B on reasoning benchmarks — but improved reasoning doesn't mean improved injection resistance. No local model should be trusted to catch every injection.

ClawHub Supply Chain Attacks

Demonstrated by: Jameson O'Reilly, DVULN **Status:** Partially addressed, fundamental risk remains

O'Reilly uploaded a benign skill to ClawHub (OpenClaw's plugin marketplace), artificially inflated the download count to 4,000, and watched developers from **seven different countries** install it within hours. The skill did nothing malicious — but it could have.

ClawHub had **zero moderation process** at launch. The developer notes literally stated: “All downloaded code will be treated as trusted code.”

A broader analysis of 31,000 agent skills found that **26% contained at least one vulnerability**. The most severe findings included active data exfiltration — skills that silently send your data to external servers via curl commands, with no user notification.

Every plugin runs with whatever permissions you’ve granted the agent. One malicious update to a popular skill and your personal AI assistant becomes an exfiltration tool.

Cross-Site WebSocket Hijacking

Status: Patched

The gateway server didn’t validate WebSocket origin headers, meaning any website could establish a connection. A malicious webpage could retrieve an auth token, open a WebSocket, disable sandboxing, and achieve remote code execution — all from a browser tab.

Google Account Suspensions (February 2026)

Status: Ongoing risk for OAuth-based provider connections

Starting February 12-14, 2026, Google suspended accounts of users who connected their Google AI Ultra subscriptions ($250/month) to OpenClaw via OAuth tokens. Google DeepMind engineer Varun Mohan stated they’d seen a “massive increase in malicious usage of the Antigravity backend” that degraded service for other users.

The core issue: OpenClaw routed requests through private OAuth tokens instead of the standard per-token API. Users on flat-rate subscriptions consumed far more compute than Google’s pricing model assumed. Google classified this as ToS-violating automated access.

Some users reported losing access beyond just AI services — Gmail, Workspace, YouTube, and account histories going back years. No advance warning, no refunds. OpenClaw creator Peter Steinberger called it “pretty draconian” and announced plans to remove Antigravity OAuth support.

What this means for you: Do not connect flat-rate AI subscriptions (Google AI Ultra, Anthropic Pro) to OpenClaw via OAuth. Use per-token API keys instead — Anthropic and OpenAI both offer standard API access that’s designed for programmatic use. The OAuth shortcut saves money until the provider notices, and then you lose the account.

Additional CVEs

OpenClaw deployments using `mcp-remote` are exposed to **CVE-2025-6514**, a command-injection RCE vulnerability. Without isolation, a successful injection means full host compromise.

Essential Security Measures

If you're going to run OpenClaw despite the risks, these measures reduce (but do not eliminate) your attack surface.

Run on Dedicated Hardware or a VM

Do not run OpenClaw on your daily-driver machine. Use a dedicated Mac Mini, a spare laptop, or a virtual machine. If the agent is compromised, the blast radius stays contained.

At minimum, run it in a hardened Docker container:

- Non-root user inside the container
- Read-only filesystem
- Drop all unnecessary Linux capabilities
- Strict, explicit volume mounts (don't mount your entire home directory)
- Restrict outbound network access to only required domains

Use Throwaway Accounts for Testing

Create dedicated email addresses, messaging accounts, and API keys specifically for OpenClaw. Do not connect your primary accounts while you're evaluating it.

If you decide to connect real accounts later, you'll do so with full awareness of what you're exposing.

Set Up Cloudflare Tunnel

If OpenClaw needs to communicate with the outside world (webhooks, messaging platform callbacks), **do not expose your home network directly**. Cloudflare Tunnel creates a secure bridge between your local machine and the internet without opening ports on your router.

This is why Cloudflare's stock moved — the OpenClaw community adopted tunnels almost universally. It's the right call. Direct port forwarding with an AI agent that has shell access is asking for trouble.

Rotate Credentials Regularly

OpenClaw historically stored credentials in plaintext local config files. v2026.3.2 introduced SecretRef support across 64 credential targets with `openclaw secrets` planning/apply/audit workflows — use it. Unresolved refs now fail fast on active surfaces, which is a real improvement. But assume credentials can still be compromised. Rotate API keys on a schedule — weekly if you're actively using it, immediately if anything looks suspicious.

Use a credential management service (like Composio Managed Auth) to route integrations so the agent never handles raw tokens directly.

Network Isolation

Restrict the agent's network access to only the domains it actually needs. Block everything else at the firewall level. If OpenClaw only needs to reach the Anthropic API and your email provider, those should be the only allowed outbound connections.

Disable Unnecessary Skills

Every enabled skill is an attack surface. Start with the minimum set you actually need. Disable browser automation, shell commands, and file system access unless you have a specific use case that requires them.

The high-risk tools to limit or allowlist:

- `exec` (shell command execution)
- `browser` (web browsing/automation)
- `web_fetch` / `web_search` (outbound HTTP)
- `file_read` / `file_write` (filesystem access)

Use `ollama launch openclaw` for Local Setup

If you're running Ollama locally, `ollama launch openclaw` (v0.17.0+) handles the connection between Ollama and OpenClaw automatically. This is safer than manual configuration because it sets up the local API connection without exposing Ollama to the network — no need to bind to `0.0.0.0`. As of Ollama v0.17.7, it also supports context compaction for long agent sessions. Update Ollama to v0.17.7 before using it: `curl -fsSL https://ollama.com/install.sh | sh`.

Keep It Updated

The project is actively patched — v2026.3.2 (March 2026) added Gateway loopback WebSocket hardening, symlink escape protection, SecretRef credential management across 64 targets, and plugin route deduplication prevention. Run the latest version. Check the changelog before updating — v2026.3.2 has breaking changes (default tool profile switched to `messaging`, ACP dispatch on by default, plugin SDK removed `registerHttpHandler`).

For Ollama, v0.17.6+ is required for Qwen 3.5 tool calling to work correctly. Earlier versions routed Qwen 3.5 through the wrong parsing pipeline, which could produce malformed tool calls — a reliability issue with security implications if your agent is executing commands based on those calls.

What NOT to Connect

Some integrations carry unacceptable risk regardless of your hardening measures:

Financial accounts. Banking, brokerage, cryptocurrency wallets, payment processors. A prompt injection that initiates a transfer is irreversible.

Health data. Medical records, health apps, insurance portals. HIPAA violations aside, this data is permanently sensitive and cannot be “un-leaked.”

Client communications. If you handle other people’s data professionally — legal, medical, financial, consulting — connecting those systems to an agent with known architectural vulnerabilities is a liability time bomb.

Production infrastructure. SSH keys to servers, cloud console access, CI/CD pipelines. An agent with shell access and your AWS credentials is one prompt injection away from a very bad day.

Password managers. Do not give OpenClaw access to your vault. The agent doesn’t need every credential you own to be useful.

Flat-rate AI subscriptions via OAuth. Google suspended AI Ultra subscribers ($250/month) who connected via OAuth in February 2026 — some lost access to their entire Google account. Anthropic has also banned accounts. Use per-token API keys, not subscription OAuth tokens.

The Architectural Problem

The individual bugs get patched. The localhost bypass was fixed. The WebSocket hijacking was fixed. But the architectural problems aren't bugs — they're features working as designed.

Prompt injection is unsolved

OpenClaw connects to your email, messaging apps, and social accounts. It reads incoming content and acts on it. But LLMs process all text as potential instructions. A carefully crafted WhatsApp message with hidden directives will be processed just like any other input. The model may follow those instructions — forwarding credentials, executing commands, exfiltrating data.

This isn't an OpenClaw-specific flaw. It's intrinsic to how language models work. No one has a reliable fix. Enterprises address it by restricting what content agents can access and what actions they can take. OpenClaw's entire value proposition is doing the opposite.

Agents need broad permissions by design

An agent that can't read your email, access your calendar, or execute commands isn't useful. But an agent that can do all of those things has the same capabilities as malware. The difference is intent — and intent can be hijacked via prompt injection.

Palo Alto Networks called this the **"lethal trifecta"**: access to private data, exposure to untrusted content, and the ability to perform external communications while retaining memory. All three are core features, not bugs.

Running it safely defeats the purpose

As one security team noted: "A sandboxed assistant can't access your real email and calendar." The security-utility tradeoff is stark. The more you lock down OpenClaw, the less useful it becomes. The more useful you make it, the more exposed you are.

Who Should Wait

Most people. If you need to ask whether OpenClaw is safe to run, the answer for you is "not yet."

Specifically, wait if:

- You don't know what a reverse proxy is or why it matters
- You can't explain the difference between `localhost` and `0.0.0.0`

- You don’t have dedicated hardware to isolate it
- You handle sensitive data professionally (legal, medical, financial)
- You want to connect it to accounts you can’t afford to lose
- You don’t have a credential rotation process in place

Enterprise-grade alternatives are coming. VC-funded agentic AI companies are already shipping products with professional security practices, audit logging, and managed credential handling. Google’s Gemini integrations, for example, run inside Google’s security perimeter — the AI accesses your Gmail without your API keys ever touching an open-source plugin.

The window between “OpenClaw proves this is possible” and “professional alternatives exist” is measured in months, not years. Waiting is the rational choice for the vast majority of users.

Who can experiment carefully

If you understand VPS deployments, network isolation, Docker hardening, and credential management — and you’re willing to use throwaway accounts on dedicated hardware — OpenClaw offers a genuine glimpse of where personal AI is headed. The capabilities are real. A user asked it to make a restaurant reservation; when OpenTable didn’t have availability, OpenClaw found AI voice software, called the restaurant directly, and secured the reservation over the phone. That kind of autonomous problem-solving is new and genuinely impressive.

Just know what you’re signing up for.

Bottom Line

OpenClaw is not malware. It’s an ambitious, fast-moving open-source project built by a solo developer who open-sourced a personal tool and watched it go viral. The capabilities are real — autonomous task completion, multi-platform integration, persistent memory, creative problem-solving.

The security posture is also real — and it’s immature. 42,000+ exposed instances. Credential theft in under 5 minutes via prompt injection. A plugin marketplace where a quarter of skills have vulnerabilities. An architecture that fundamentally requires broad permissions to be useful.

If you run it:

1. **Dedicated hardware or VM** — never your daily machine
2. **Throwaway accounts** — not your real email, not your real credentials

3. **Cloudflare Tunnel** — never expose your network directly
4. **Docker with hardening** — non-root, read-only FS, restricted network
5. **Minimal skills** — disable everything you don't actively need
6. **Rotate credentials** — weekly at minimum, immediately if suspicious
7. **Never connect** financial, health, client, or production systems

If you don't run it: you're making the smart call. Enterprise alternatives with proper security are months away. The capabilities OpenClaw demonstrates are coming to products with audit logging, managed auth, and actual security teams behind them. You'll lose nothing by waiting.

OpenClaw matters because it proved the demand for AI agents that actually do things. The security story will catch up. It just hasn't yet.

This guide reflects publicly available information as of March 2026 (OpenClaw v2026.3.2, Ollama v0.17.7). The project is evolving rapidly. Check OpenClaw's security documentation and GitHub issues for the latest patches and advisories before making any deployment decisions.

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/openclaw-security-guide/

Free guides for running AI locally