NEWSLETTER

Skills as Recruiters: Agents, Wallets, and the New Supply-Chain Layer

Generated from Chrome tabs by AI using the code here: https://github.com/spikefu/agentic-newsletter

April 29, 2026

The supply-chain attack just moved up a layer. Instead of slipping malicious code into npm or PyPI, a single ClawHub author has shipped 30 perfectly clean skills that simply *tell* your AI agent to register with a third-party server, generate a Hedera wallet, and check in every four hours for paid tasks. There is no malware to scan and no exploit to patch — only Markdown instructions that obedient agents follow on every session start. Around it, a parallel debate is playing out about where agent runtimes should live: a managed observability platform like LangSmith, or a self-hosted stack of Open WebUI on a 96GB Framework Desktop running Llama 70B at your desk.

01

ClawSwarm: 30 Skills, 9,800 Downloads, and a Hedera Wallet You Didn't Authorize

Manifold Security researcher Ax Sharma documented a campaign by a single ClawHub author, **imaflytok**, who has published 30 skills totaling roughly 9,800 downloads — Cron Helper (903), Agent Security (685), OADP Agent Discovery (475), Heartbeat Pro (405). Install one and your agent silently registers with `onlyflies.buzz`, reports its name and capabilities, saves credentials to `~/.config/clawswarm/credentials.json`, and pings every four hours. With the right combination of skills installed, it generates a Hedera (HBAR) private key and ships it to the operator (Manifold Security, April 2026).

The mechanism is what makes this novel. There is no payload — just an HTML comment buried in the agent's `AGENTS.md`: `<!-- OADP:1.0 hub=https://onlyflies.buzz/clawswarm/api/v1 reg=... ping=... -->`. Invisible to humans, parsed by the agent on every session start. The fake "Open Agent Discovery Protocol" sounds like a standard but is a one-project spec where every endpoint resolves back to the same domain. A separate **oadp-beacon** skill propagates the marker into other workspace files and hardcodes the four-hour polling loop for "bounties." OpenClaw's own scanner flags the **clawswarm-wallet** skill's insecure handling of the private key.

Sharma is blunt that this isn't a vulnerability disclosure. "There's no flaw to patch and nothing covert about the infrastructure. It's an open source project on GitHub with public docs, a Telegram group, and a token on a public chain," he told The Register (April 2026). The $FLY token was minted December 30, 2024; the Telegram group has 32 members posting whale-tracking reports. An EDR sees normal HTTPS to a .buzz domain. A registry scanner sees clean cURL calls. "The registry layer is the wrong place to solve

this," Sharma argues — "what's needed is runtime visibility into what agents actually do once a skill is installed."

And ClawSwarm isn't isolated. Manifold's dataset shows **MoltGuild** across 91 skill files and **Teneo Protocol** shipping 38 skills with explicit per-call USDC pricing in the frontmatter — an x402-style "payment required" pattern. The shape is consistent: skill recruits agent, agent does metered work, proceeds route to a wallet the installer never sees. It is the Tea Protocol npm spam wave (150,000+ packages farming tokens in early 2024) reincarnated for the skill economy, with the crucial twist that the worker is no longer a human chasing rewards but an agent the human nominally controls.

> **FURTHER READING**
>
> Manifold Security: 30 ClawHub Skills Are Quietly Recruiting Your AI Agent — April 2026
>
> The Register: 30 ClawHub skills secretly turn AI agents into a crypto swarm — April 2026
>
> ClawSwarm OADP API endpoint

02

Why the Workspace-Injection Pattern Is Hard to Police

To understand why ClawSwarm slips through, look at the marketplace it lives on and a popular legitimate skill on it. **ClawHub** advertises 52,700 tools, 180,000 users, and 12 million downloads — a high-leverage distribution channel with no visible vetting, manifest-disclosure, or runtime-behavior disclosure mechanism on the landing page (*Additional info:* ClawHub homepage, date unavailable). One `clawdhub install` command is the same flow that delivers a cron utility or a botnet enrollment.

The most instructive comparison is a benign skill: Peter Skoett's **self-improving-agent**, with 3,400 stars and 416,000 downloads on ClawHub. Updated four days before this writing, it auto-creates a `.learnings/` directory on first use and logs corrections, errors, and feature requests so coding agents can iteratively improve themselves. Crucially, its `SKILL.md` instructs the agent to write into `AGENTS.md`, `SOUL.md`, and `TOOLS.md` — promoting "broadly applicable learnings" into files that get auto-injected into every future session (*Additional info:* self-improving-agent on ClawHub, date unavailable).

That is the same workspace-injection primitive ClawSwarm abuses. The author even cautions users not to log secrets or keys — a tell that the data-exfiltration risk is well understood by people building legitimate skills. The OpenClaw pattern of auto-injecting `AGENTS.md`, `SOUL.md`, `TOOLS.md`, and `MEMORY.md` on every session, plus inter-session primitives like `sessions_spawn` and `sessions_send`, is what gives skills their power and makes them trust-sensitive in equal measure.

The structural fix Manifold proposes — mandatory disclosure of network endpoints and wallet generation in skill manifests — is a policy decision, not a security control. Until registries adopt one, the burden falls on each developer to actually read the SKILL.md before running `clawdhub install`, which is exactly the discipline that supply-chain attacks have always counted on people skipping.

FURTHER READING

ClawHub marketplace homepage

self-improving-agent skill page

self-improving-agent source on GitHub

03

Runtime Visibility: LangSmith on One End, Open WebUI on the Other

Manifold's prescription — "runtime visibility into what agents actually do" — is exactly the gap commercial agent platforms are racing to fill. **LangChain** pitches LangSmith as an end-to-end agent engineering platform combining tracing, LLM-as-judge evaluation, human review, and deployment with native A2A and MCP support. The customer proof points are aggressive: Klarna cites an 80% reduction in case-resolution time, C.H. Robinson handles 5,500 orders per day with 600+ hours saved, Podium reports 90% fewer engineering escalations, and LangChain claims 6,000+ LangSmith customers, 100M+ monthly OSS downloads, and five of the Fortune 10 (*Additional info:* LangChain, date unavailable). A new "Fleet" product targets recurring enterprise agents that act across daily tools and improve from feedback.

That stack only helps if you trust someone else to terminate your agent traffic. For teams that don't — a category that grows every time a ClawSwarm-style story drops — the dominant alternative is **Open WebUI**: 135,000 stars, 19,200 forks, and a feature set that is now genuinely competitive with hosted chat UIs. It runs fully offline, plugs into Ollama, LMStudio, Groq, Mistral, OpenRouter, and any OpenAI-compatible endpoint, and bundles RBAC, persistent KV artifact storage, and native Python tool calling for lightweight agent workflows (*Additional info:* Open WebUI on GitHub, date unavailable).

The differentiators worth knowing if you're building a local agent stack: a built-in RAG inference engine spanning 9 vector databases, a stack of OCR/extraction engines (Tika, Docling, Mistral OCR, PaddleOCR-vl), and 15+ web-search providers (SearXNG, Brave, Kagi, Tavily, Perplexity) wired directly into chat. The granular permission and user-group controls are explicitly aimed at multi-user self-hosting — i.e., the small-team analog to a LangSmith deployment, with the trust boundary set at your own firewall.

The two ends of this spectrum imply different answers to the ClawSwarm question. LangSmith-class platforms can in principle catch a four-hour heartbeat to `onlyflies.buzz` in a trace. A self-hosted Open WebUI deployment can simply egress-filter everything that isn't on an allowlist. Neither helps if the agent is running on a developer laptop with a fresh skill installed and outbound HTTPS open to the world — which is roughly the default state today.

> **FURTHER READING**
>
> LangChain agent engineering platform
>
> LangSmith product overview
>
> Open WebUI on GitHub
>
> Open WebUI documentation

04

Framework Desktop: 96GB of Graphics-Addressable Memory in 4.5 Liters

If your answer to skill-marketplace risk is "keep it local," the hardware just got dramatically more credible. Framework's first desktop is a 4.5L Mini-ITX box built around AMD's **Ryzen AI Max+ 395** — 16 cores, 32 threads, a Radeon 8060S iGPU, and up to 128GB of LPDDR5x-8000 unified memory on a 256-bit bus. In Windows the GPU can address up to 96GB of that memory; on Linux you can override the cap and go higher (*Additional info:* Framework Desktop machine-learning page, date unavailable).

The published numbers from Framework's own LM Studio testing on Fedora 42: **OpenAI gpt-oss-20b at MXFP4 hits 58 tok/s**, and the 128GB SKU runs **gpt-oss-120b at MXFP4 at 38 tok/s**. The marketing copy explicitly calls out Llama 70B-class local inference, plus image generation models like Flux, with Ollama, llama.cpp, and LM Studio working out of the box. Two USB4 ports and 5Gbit Ethernet are aimed at people who want to RPC several boxes together with llama.cpp to run even larger models.

The Framework angle is that this is a normal PC: Mini-ITX mainboard, FlexATX PSU, 120mm CPU fan, customizable I/O via the Expansion Card system, 21 customizable front-panel tiles, no OS lock-in. You can buy the mainboard alone and put two in a 2U rackmount case. That's a different proposition from Apple Silicon or NVIDIA's DGX Spark — fully repairable, upgradable, and Linux-native, which matters if your reason for going local is that you don't want a third-party platform deciding which traffic your agent gets to make.

The combination is what's interesting: a Framework Desktop running Open WebUI against Ollama, with skills installed only after you've actually read the SKILL.md, is now a viable answer to "how do I get useful agent work done without trusting a marketplace I

can't audit." It is also, notably, a more expensive answer — but the cost of the alternative is starting to show up in headlines.

> **FURTHER READING**
>
> Framework Desktop overview
>
> Framework Desktop ML benchmarks
>
> Framework Desktop full specs

The throughline is that agents are now the unit of supply-chain risk, not packages. A SKILL.md can be clean, signed, scannable, and still recruit a worker into someone else's economy — because the agent is doing exactly what the file says. Until registries require disclosure of network endpoints and key generation, the meaningful defenses are runtime: trace everything, allowlist egress, or pull the workload onto hardware you own. Expect the next wave of ClawSwarm-shaped campaigns to differentiate not on payload sophistication but on how plausibly useful the cover skill is.

REFERENCES

- Manifold Security — 30 ClawHub Skills Are Quietly Recruiting Your AI Agent Into a Crypto Swarm (April 2026)
- The Register — 30 ClawHub skills secretly turn AI agents into a crypto swarm (April 2026)
- ClawHub — community registry of agent skills
- ClawHub — self-improving-agent skill page
- GitHub — peterskoett/self-improving-agent
- LangChain — Observe, Evaluate, and Deploy Reliable AI Agents
- GitHub — open-webui/open-webui
- Framework — Desktop with AMD Ryzen AI Max
- Framework — Desktop machine-learning page