The AI Agent That Just Became Shadow IT

The 'We Will Just Use OpenClaw' Problem No One Is Talking About

---

Last week I had a conversation I keep thinking about.

A sharp and commercially astute general manager was walking me through his automation ambitions. Midway through, he dropped a phrase I've been hearing a lot:

> *"We were thinking of just using OpenClaw for that."*

He said it the way people used to say, "we'll just put it on the cloud." Casually. As if the hard part was already solved.

I didn't dismiss it. OpenClaw is genuinely impressive. But I did slow the conversation down, because that phrase, "just use OpenClaw", contains an assumption that concerns me far more than any specific tool.

First, Let Me Be Fair About What OpenClaw Is

OpenClaw is an open-source autonomous AI agent framework. You define an objective and it plans the steps, executes actions across APIs and tools, maintains memory across sessions, and keeps working toward the goal without waiting for step-by-step instructions. It sends emails, controls browsers, runs shell commands, manages Slack and Telegram integrations, and automates workflows end-to-end.

That is not a chatbot. That is closer to an autonomous operator.

For comparison, Claude Code, which I use heavily in my own development work, operates on a fundamentally different philosophy. It works inside your coding environment, reads your codebase, and helps you write, refactor, and maintain code faster. You remain in control throughout. It accelerates execution but does not independently manage tasks. The design principle is augmentation, not autonomy.

The distinction matters enormously in enterprise contexts: one explores how independently AI can function; the other focuses on making humans more effective. Neither is superior in the abstract. They answer different requirements.

What I Told Him, and Why

When I gently raised the security question, he looked at me the way clients sometimes do when they think the consultant is padding the engagement.

So, I was direct.

OpenClaw went from a weekend project to 179,000 GitHub stars in under a month. In that same period, it triggered security advisories from Cisco, CrowdStrike, Kaspersky, Microsoft, and Palo Alto Networks. Independent scans found over 40,000 exposed instances across 52 countries, with 93% exhibiting authentication bypass conditions. A Kaspersky audit identified 512 vulnerabilities, eight of them critical.

The most severe, CVE-2026-25253, allowed an attacker to create a malicious webpage, have the agent visit it, and steal the gateway token. Full administrative control, in milliseconds, without a login barrier.

But the vulnerability I spent more time explaining was subtler. Security researchers at Cisco and Sophos call it the Lethal Trifecta: an agent that simultaneously holds access to sensitive data, has external communication capability, and reads untrusted input. When all three conditions exist together, and they do by default in OpenClaw, a single malicious email, document, or webpage can instruct the agent to silently forward your client data, extract credentials, or create backdoor integrations. Without you ever typing a prompt.

The agent does not know it has been weaponised. It is simply following instructions, the way it was designed to.

That got his attention.

The Real Issue Is Not the Tool

Here is where I want to be honest about something, because it matters beyond this conversation.

OpenClaw's security problems are real and documented. But the pattern they reveal is not unique to OpenClaw. It is a pattern that emerges whenever powerful capability is adopted faster than governance structures can accommodate it.

The skills ecosystem, the plugins and extensions that give AI agents their "tentacles", is the new attack surface. Supply-chain poisoning is already happening at scale: over 1,100 malicious packages appeared in OpenClaw's ClawHub marketplace in a single campaign, disguised as productivity tools and trading bots. Users installed them without hesitation, because they looked useful and the documentation was convincing.

Old tricks. New environment. Faster consequences.

What changes with autonomous agents is not the class of vulnerability. It is the speed and persistence of the impact. Traditional software runs attacker input once. An agent may reason over it indefinitely. Context becomes memory, memory becomes authority, and authority compounds over time. The attack surface becomes temporal, not just technical.

Why "Just Deploying AI Agents" Is a Programme, not a Task

This is the point I made most firmly, and it is the point I make to every organisation considering agentic AI.

Deploying autonomous AI agents into a business is a transformation initiative.

It requires:

- **Governance architecture before deployment:** Who authorises which agents to do what? What data can they access? What actions require human approval before execution? These are not IT questions. They are business policy questions that need documented answers before a single agent touches a live system.
- **Change management, not just change:** The people whose workflows are being automated need to understand what the agent can and cannot do, what safe usage looks like, and how to recognise when something has gone wrong. An autonomous agent that malfunctions at machine speed causes machine-speed damage before anyone notices.
- **Risk classification that treats agents as privileged infrastructure:** An agent with access to your email, CRM, file systems, and external APIs is not a productivity tool. It is a privileged system. It deserves the same isolation, monitoring, and access governance you would apply to your most sensitive infrastructure.
- **Structured rollout, not viral adoption:** The most dangerous scenario I see in enterprises today is employees self-deploying powerful autonomous agents without security review, connected to enterprise SaaS tools, often without IT approval. OpenClaw's own maintainer stated publicly: "If you can't understand how to run a command line, this is far too dangerous a project for you to use safely." That is not a ringing endorsement for broad organisational rollout.
- **Incident response planning before, not after:** What happens when an agent acts unexpectedly? Who is accountable? What is the rollback procedure? These questions have answers in well-governed programmes. They do not have answers in ad-hoc tool deployments.

What I Recommended

I did not tell my client to avoid agentic AI. The productivity benefits are real, and the market is moving decisively in this direction.

I recommended a different approach to sequencing.

Start with bounded, high visibility use cases where the agent's actions are auditable and the blast radius of an error is limited. Build your governance layer first, then expand agent capability within it. Choose tools that ship with security defaults rather than requiring manual hardening, six to eight hours of security configuration per deployment is not a scalable model for any enterprise team.

Use the first deployment as your change management proof-of-concept. Document what worked, what surprised people, what required escalation. That knowledge base is what allows you to scale with confidence rather than with fingers crossed.

And critically: get your programme management foundations right before you get your agent configuration right. The technology will evolve rapidly. The discipline of structured, governed change is what keeps the evolution under control.

The Pattern I Keep Seeing

Every major technology wave has a moment where the capability outpaces the governance. Cloud computing went through it. Mobile went through it. Open source went through it.

Agentic AI is going through it now, in accelerated form. OpenClaw's rise from obscurity to enterprise security crisis in under a month is not an anomaly. It is a preview.

The organisations that will capture the value from agentic AI are not the ones that move fastest. They are the ones that build governance structures capable of moving fast safely. That requires transformation knowledge, project and programme management experience, change management discipline, and AI implementation experience working together, not in sequence.

When a client says "we can just use OpenClaw for that," what I hear is an organisation ready to invest in automation, but not yet ready to govern it.

That gap is exactly where the real work begins.

If this sounds like an issue you had or even if you are curious about this, I would welcome a conversation. And if you've had your own "we can just use X for that" moment, I am curious to hear how it played out.

Disclaimer and Disclosure

Third-party Content and AI Assistance: This article references tools and software that are publicly available and proprietary to their respective creators. The author does not claim ownership or affiliation with these third-party products. This article was written by the author with assistance from Generative AI Language Models.
Transparency Notice: While every effort has been made to ensure accuracy, readers should verify information independently and consult official sources or documentation for the mentioned tools and software. The use of AI in the writing process is disclosed in the interest of transparency, but all opinions and analyses are the author's own unless otherwise stated.