

ISSUE #007 · MACHINE SPEED

Machine Speed.

An AI agent drove a four-pivot intrusion from a public notebook to a database dump in about an hour, improvising SQL as it went. A worm backdoored 5,561 repositories in six hours by editing CI workflows to steal cloud tokens. A poisoned VS Code extension stayed live for eleven minutes and still read the config and auth files your AI coding assistant keeps on disk.

> ATTACK

Michael Clark at Sysdig Threat Research · 2026-05-26 · **AI agent at the wheel: How an attacker used LLMs to move from a CVE to an internal database in 4 pivots**

The entry point is an internet-reachable marimo notebook. The attacker hits `CVE-2026-39987`, an RCE reachable over a WebSocket to the `/terminal/ws` endpoint, and lands code execution. Pivot one harvests credentials from environment files on the host. Pivot two replays the stolen AWS credentials, but the calls come from Cloudflare Workers used as a per-request egress pool: twelve API calls land across eleven distinct source IPs in 22 seconds, a `sts:GetCallerIdentity` to orient, then enumeration. Pivot three calls `secretsmanager:GetSecretValue` and pulls an SSH private key straight out of AWS Secrets Manager. Pivot four SSHes to a bastion and dumps a PostgreSQL database. The full chain runs 18:23 to 19:32 UTC, about an hour start to finish.

Sysdig's read that a model was driving rests on four tells. The agent improvised, targeting a `credential` table it could only guess at. A planning comment in Chinese leaked into the command stream across six source IPs at sub-second cadence. The commands were machine-shaped: `echo '---'` delimiters, quoted-EOF HEREDOCs bundling six SELECTs at once, `head` truncation, `-P pager=off`, `2>/dev/null` on everything. Each step consumed the prior step's output, reading creds from `~/.pgpass` and a SecretId from a ListSecrets response without a human in the loop.

• SHIP THIS WEEK

Alert when a workload role calls `secretsmanager:GetSecretValue` on a secret holding SSH or private-key material, especially right after a `sts:GetCallerIdentity` from the same principal. Flag one IAM principal making API calls from many distinct egress IPs in seconds: twelve calls across eleven IPs in 22 seconds points at an edge network used as a proxy pool, moving faster than any human at a keyboard. Pull internet-reachable dev tools (marimo, Jupyter, Langflow, any notebook server with a terminal endpoint) off the public internet and behind auth.

> RULE

Rohan Prabhu at StepSecurity · 2026-05-22 · **Megalodon: Mass GitHub Actions Secret Exfiltration Across 5,500+ Public Repositories**

Megalodon backdoored 5,561 public repositories in a six-hour window on May 18. It shipped two workflow variants: `SysDiag.yml`, the mass version, and `Optimize-Build.yml`, a dormant targeted version. Both request `permissions: id-token: write`, which lets the workflow request a short-lived OIDC token from GitHub. The payload steals that token to reach any cloud role whose trust policy accepted that repository's GitHub OIDC claims, and separately sweeps AWS keys, GCP OAuth tokens, Azure IMDS credentials, SSH keys, kubeconfigs, Terraform credentials, and Docker configs off the runner. Exfiltration is an HTTPS POST to `216.126.225.129:8443`. **Alert on any commit that adds `id-token: write` outside your reviewed allowlist, and require a human review on every change that grants it.** Grep `.github/workflows` org-wide for `SysDiag.yml` and `Optimize-Build.yml`, and for commits by `build-bot@github-ci.com` or `ci-pipeline@actions-bot.com`. A pull request that lets a workflow request an OIDC token is a privileged change, and it earns a review every time.

> DEFENDER

Ashish Kurmi at StepSecurity · 2026-05-18 · **Nx Console VS Code Extension Compromised**

A trojanized build of Nx Console (`nrwl.angular-console`, over 2.2 million installs) went out through VS Code Marketplace auto-update. Version 18.95.0 was live for eleven minutes, published 12:36 UTC and pulled 12:47. Injected code in `main.js` ran the instant a developer opened any workspace, harvesting GitHub, npm, AWS, Vault, Kubernetes, and 1Password credentials, and specifically reading `~/.claude/settings.json`, which StepSecurity describes as possibly one of the first supply chain payloads built to harvest AI coding assistant credentials and configurations. Per Nx's own postmortem, the chain traced up to the TanStack npm compromise of May 11, where an attacker abused TanStack's GitHub Actions OIDC trusted-publisher binding to ship 84 malicious versions across 42 `@tanstack/*` packages (`CVE-2026-45321`). An Nx contributor's `pnpm install` resolved `@tanstack/zod-adapter@1.166.15`, whose `prepare` script lifted their GitHub token. The repo where that install ran set a seven-day cooldown (`minimum-release-age=10080`), but a pinned `pnpm` predating that feature meant it never fired. **Treat your coding assistant's config and auth files the way you treat `~/.aws` and `~/.ssh`: add `~/.claude/settings.json` to file-access monitoring and secret scanning, and review which high-install extensions can read your home directory.**

> AGENT BENCH

The Rule put an alert on new `id-token: write` grants. The backlog already in your repos is the other half: every workflow that grants it today, none of it reviewed. Point a coding agent at `.github/workflows`, `.github/actions`, and any reusable workflows, and have it surface four things: new or unexpected `id-token: write` grants, and whether the job exchanges that OIDC token for a cloud role; actions pinned to a mutable tag instead of a full commit SHA; untrusted context (PR title, branch name, issue body) interpolated into `run:` blocks, the injection sink; and `pull_request_target` jobs that check out, build, or run PR-controlled code under privilege. **You get a triaged list at `file:line` of which workflows can reach a cloud role and which splice attacker input into a shell. Treat it as a first-pass filter and confirm every finding by hand.**

> RADAR

Pushkar Joglekar (Broadcom) and Tabitha Sable (Datadog) at kubernetes.io · 2026-05-26 · **Reconciling the Past: Correcting Records for Unfixed Kubernetes CVEs**

On June 1 the Kubernetes Security Response Committee corrected the records for three unfixed CVEs: `CVE-2020-8561` (kube-apiserver follows webhook redirects, so an actor who can configure an admission webhook can aim API server requests at internal networks), `CVE-2020-8562` (a DNS TOCTOU race that bypasses IP restrictions), and `CVE-2021-25740` (Endpoints and EndpointSlice objects let a user forward a LoadBalancer or Ingress to backends in another namespace). All three stay unpatched by design: the webhook-redirect and cross-namespace-forwarding behaviors are features legitimate tools depend on, and the DNS race gets operational mitigation in place of a breaking code change. The records wrongly carried a `fixed version` field, which gave a false all-clear. Your scanners may now start flagging these where they previously showed clean. The mitigations are admin-side and per-CVE: restrict write access to Endpoints and EndpointSlices (audit clusters upgraded from older versions), and run a local DNS cache with `min-cache-ttl` for the API server. A `fixed` field in a CVE record is a claim. Confirm the mitigation lives in your cluster.

