THE STATE OF AI · TRENDS TO WATCH · H2 2026

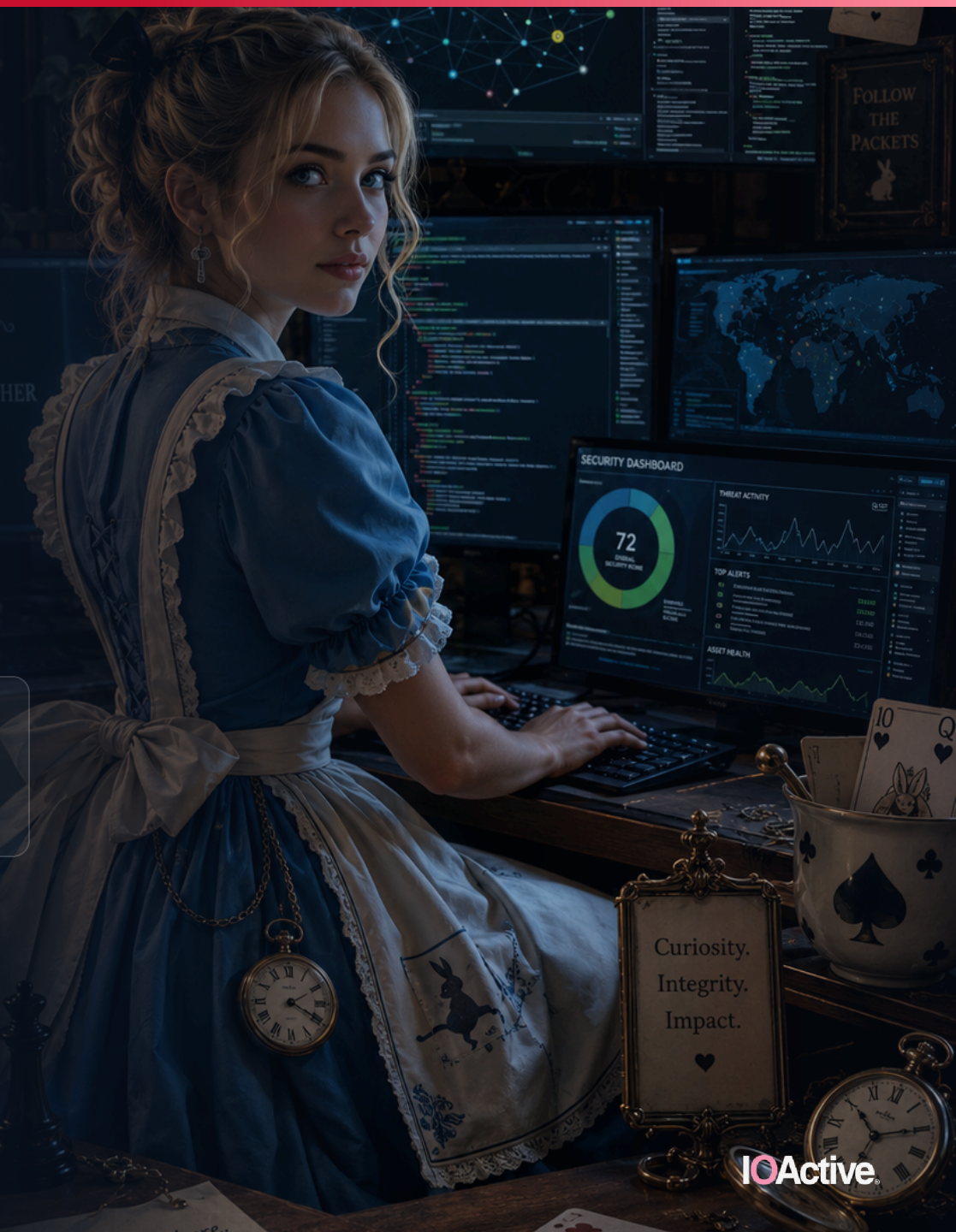
Ready. Fire!! Aim.

“I give myself very good advice, but I very seldom follow it; that explains the trouble that I’m always in.”

— Alice, in Wonderland

The AI capability & adoption curves went vertical. The **security budget** did not. This talk covers that gap.

Robert E. Lee · Director of AI Strategy, IOActive



IOActive.

THE SPEED

Adoption is outpacing safeguards.

The tech is compelling: what took experts days to weeks can now land in hours — so it lands in every use case teams can find, faster than effective controls can contain.

READY

15% → 93%

In two years AI went from failing nearly all professional-grade security challenges to meaningfully contributing to all of them — exploitation, cryptography, reverse engineering. **What wasn't possible suddenly is.**

FIRE!!

88%

of organizations now use AI somewhere — landing in every use case they can find, coding to customer-facing agents, sanctioned or shadow.

AIM… LATER

62%

cite **security & risk** as the **#1 blocker** to scaling — fewer than **10%** have scaled. But it's being employed everywhere, **often without requisite controls.**

Ready. Fire!! — the late-'90s internet rush again: attack surface and trust boundaries extended before the implications are understood. **That gap radiates into three shifts in your risk.**

sources: Stanford AI Index 2026 — hai.stanford.edu/ai-index (all three figures) · Cybench — professional-level CTF benchmark, Stanford/ICLR 2025

THREE WAYS H1 2026 CHANGED YOUR RISK

And the first one you don't get to opt out of.

SHIFT 1

Your adversary upped their game

AI raised the offensive baseline against **everyone's** infrastructure — whether or not you've adopted AI at all.

SHIFT 2

The code you run is AI-written

Sanctioned or shadow. Insecure at a volume no human review process was built for.

SHIFT 3

You've made your AI implementation publicly accessible

Agents with autonomy to **act** — brilliant, credulous, and holding tools capable of real-world consequences.

The answer to all three is one thing: security fundamentals, applied to AI implementations too — **proven controls, renewed relevance.**

THREE FUNDAMENTALS THREAD THROUGH ALL OF IT:

- **threat modeling** — attack surface expanded, trust boundaries multiplied
- **least privilege** — a socially engineerable agent, holding tools
- **monitoring + attestation** — when it goes wrong, can you trace when, how, why?

SHIFT 1 · YOUR ADVERSARY GOT AN UPGRADE

Maybe you're not using AI. Your attackers are.

- **The patch window is gone.** Mandiant tracks time-to-exploit collapsing toward zero — exploits landing on or before the day the CVE goes public. Watch it live: **zerodayclock.com**.
- **Exploitation became a commodity.** Working PoC exploits in ~15 minutes; autonomous vuln discovery at **~$50 a campaign.**
- **One operator, 14+ companies.** Recovered logs (OALABS, June) — 1,000+ AI-agent sessions; the models raised a policy-violation flag only ~10 times because every job was framed as "authorized red-team."
- **Social engineering at industrial scale.** Arup wired **$25.6M** after a video call where every colleague was a deepfake (2024) — only cheaper since.

Nothing exotic. It's your **existing** attack surface, with an adversary who doesn't sleep.

sources: Zero Day Clock — live TTE tracker, 83k+ CVEs · Mandiant, "How Low Can You Go? 2023 Time-to-Exploit Trends" · 15-min PoC — darkreading.com · OSS-CRS, ~$50/campaign — arXiv 2603.08566
OALABS, Jun 16 2026 · Arup deepfake, 2024 — cnn.com · FBI IC3 2025 — fbi.gov

Shipping faster — and shipping more attack surface.

~45%

of AI-generated code carries a security weakness — across 100+ models tested (Veracode)

flat

functional correctness raced past **95%** — the security rate held at **~55%**, unchanged in two years. Smarter models, not safer code.

- **The speed is real.** Delivery genuinely increases — but the software doesn't always do exactly what the team intended.
- **Attack surface grows at delivery speed.** Code merged faster than any human review process was built for — sanctioned or shadow — with provenance nobody can attest to.
- **Appetite is rising; budgets aren't.** Clients want more frequent testing, across a higher percentage of what they produce — the appetite–budget gap is the defining constraint we now see.

Hold that gap. **Closing it is where this talk ends up.**

sources: Veracode, GenAI Code Security report — veracode.com/resources/analyst-reports/2025-genai-code-security-report (vendor data, transparent methodology)
Spring 2026 update: "AI models are still failing security" — veracode.com/blog/spring-2026-genai-code-security

SHIFT 3 · YOU'RE SHIPPING AI TO CUSTOMERS

Not exotic zero-days. Skipped fundamentals.

1. **Threat modeling — the trust boundary moved.** Untrusted input is now *executable*. **EchoLeak** (CVSS 9.3): one crafted email, Microsoft 365 Copilot leaks internal data — no click, no warning.

2. **Least privilege — the agent holds tools.** Attackers took over Instagram accounts (a White House account, Sephora, a Space Force leader) by *asking* Meta's AI support bot to change the account email. The agent's tool grants *are* its blast radius.

3. **Monitoring + attestation — you can't alarm on what you can't see.** Non-deterministic means you must *record* it — who, what, when, from where, with what result. Yet platforms are now *encrypting* their own subagent logs.

THE AGENT ATTACK SURFACE WE MAP — TO THE OWASP AGENTIC TOP 10 (2026): goal & behavior hijack · tool misuse · identity & privilege abuse · memory & context poisoning · insecure inter-agent comms · agentic supply chain · human-agent trust

Same attacker mindset we've always brought to firmware and protocols. **Now pointed at the AI you ship.**

sources: OWASP Agentic Top 10 (2026) — genai.owasp.org · EchoLeak, CVE-2025-32711 · Meta support-bot ATO — engadget.com · encrypted subagent logs — codex #28058 · OpenAI, "Safety and alignment in an era of long-horizon models," Jul 20 2026

Orgs can’t keep up with the vulns they already know about.

1,235

new CVEs in a **single** Oracle quarterly update (Jul 2026) — their largest ever.

432

CVEs the **Linux kernel** published in **two days** (Jul 20–21, 2026) — *a two-day flood.*

15–20%

of new CVEs the national database (**NVD**) still scores — since **Apr 2026**. The rest: no CVSS, no product map.

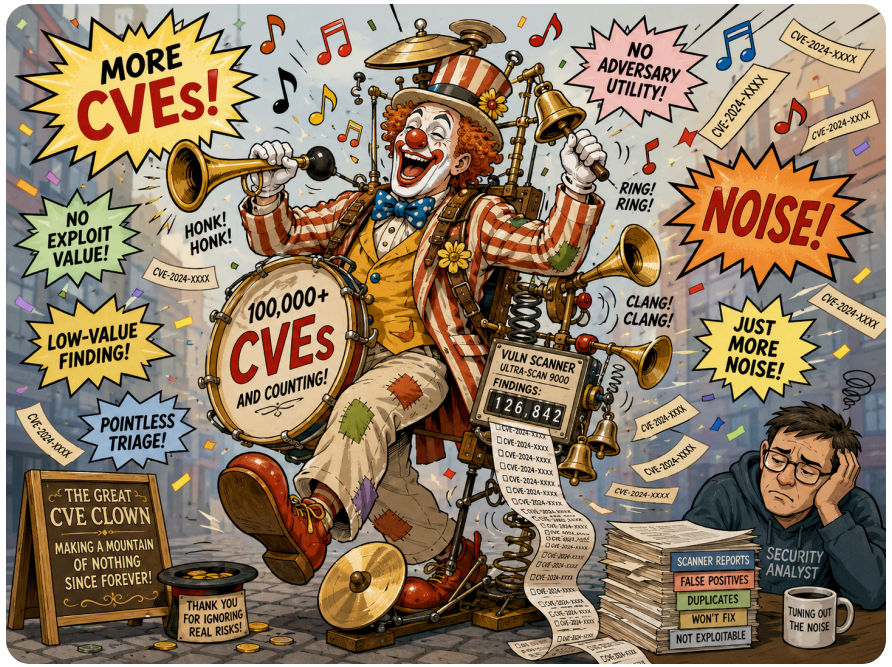
- **AI is flooding the intake — and the scoreboard walked off the field.** Torvalds calls the AI-driven security stream *“almost entirely unmanageable”*; researchers say it’s *“not feasible to prioritize”* at this rate. NVD now enriches only the highest-priority slice, and triage of the rest fell to *you.*
- **And this is still the public backlog** — already numbered, already patched, and you’re underwater. More findings was never the constraint. Which ones an attacker can actually reach — and what they’d cost *you* — is.

When discovery is cheap, we get drowned in the noise. **Which vulnerabilities actually matter?**

sources: Oracle CPU (Jul 2026) · 432 kernel CVEs — theregister.com · Torvalds "unmanageable" (May 2026) · NVD cutbacks — darkreading.com

The world doesn't need more noise generators.

- **The signal-to-noise, measured:** OpenAI's o3, pointed at the Linux kernel's SMB code, surfaced a **real, novel zero-day** (CVE-2025-37899) — found while sifting the noise. The cost, on the known benchmark bug: **8 hits vs 28 false positives** in 100 runs at 3.3k lines; at the full 12k, **1 hit in 100 runs — ~1 true report per 50 false.**
- **The method:** agents parse the repo and play a flashcard game with the model — *"any security defect in this function?"* It finds real bugs. It can't tell you whether they're **reachable**.
- **The common "fix" is an anti-pattern:** a second model judging the first. A different model class is still model confidence — not proof.
- **The bill:** token & wall-clock burn — and without a domain expert guiding it, *exploitable and impactful* is indistinguishable from noise.



kernel numbers: Sean Heelan, "How I used o3 to find CVE-2025-37899" (May 2025)

THE ANSWER · DISCOVERY IS CHEAP, VERIFICATION IS THE JOB

Who judges the judge?

1,596

vulnerabilities AI scanning disclosed in OSS (Anthropic)

vs

97

actually patched — discovery is cheap; verification is the bottleneck

- “AI proposes, **AI** verifies” doesn’t reach bedrock — the verifier believes what it reads, too.
- **The only ground truth is proof from outside the models:** a deterministic reachability path, or an exploit that **succeeds**.
- Real because you can **reach & exploit it** — not because a model is confident. What can’t be proven stays honestly **inconclusive.**
- **The research just caught up:** papers this quarter (GroundEval, Vera) score execution traces, not model confidence — GroundEval’s case: two frontier LLM judges scored the same answer **0.85+**; the deterministic oracle scored it **0.000.**

sources: "Using LLMs to secure source code" — claude.com/blog · GroundEval — arXiv 2606.22737 · Vera — arXiv 2607.01793

Point a swarm at it. Let it run.

“The spirits that I summoned — I cannot get rid of them now.” — Goethe, The Sorcerer’s Apprentice, 1797

the quick-win loop — as shared

```
/goal find all vulnerabilities in this repo. Fire up a queen-led
swarm of agents that scan this project for security vulnerabilities
in any way they can think of, using every researched method.
Refresh on SOTA methods for the detected languages as needed.

Familiarize yourself with the whole project, then file by file,
functionality by functionality, function by function.

Loop this work: continue no matter what. Don't stop for anything —
learn, improve, iterate, try new things each pass.

Mechanics: the queen acts as judge. Use a second model at the end
of each pass to check our work. Concurrency N=15, full isolation,
pipeline. Keep a log of what you tried; at the start of each loop,
check what you tried before.
```

WHY PEOPLE REACH FOR IT

- **Swarm + loop.** Fan out wide, never stop, self-improve each pass.
- **Self-updating.** Pulls the latest techniques mid-run.
- **Has a memory.** Logs every attempt so it doesn’t repeat itself.

The tell: the judge is a model. The “verification” is another model. Tokens and wall-clock burned on candidates — **no confidence in a finding unless it’s exploited, and no real reachability verdict.**

An afternoon of this hands you “findings.” The kernel again: one real zero-day, buried ~1 signal in 50.

TRY IT AT HOME · THE SAME LOOP, PRODUCTIZED

Same loop, now shipping as products.

CAPITAL ONE

VulnHunter

Open source. Reasons forward from **attacker entry points**; runs a **falsification engine** that tries to disprove its own findings before reporting. But the disproof is **the model judging the model.**

ANTHROPIC

Claude Security

Now a **Claude Code plugin**. Threat-models the repo first; every finding passes an **independent verifier agent**. Vendor's own docs: non-deterministic — *run alongside deterministic tools.*

COMMUNITY

OpenAnt · raptor · ai-sast

Scan, then LLM-verify — a model attack-simulates or grades each finding (**raptor** even grades exploitability A–F). Real bugs surface — but the verdict is **a model's, not a proof.**

Credit where due: serious engineering, real bugs — and they *do* reach toward exploitability. **But the verdict is always a model's — reachability stays an opinion, never a deterministic proof.**

the tools: github.com/capitalone/vulnhunter · Claude Security plugin — code.claude.com/docs/en/claude-security · github.com/knostic/OpenAnt · github.com/gadievron/raptor · github.com/rivian/ai-sast

HOW IOACTIVE USES AI — NOW & NEXT

Ready. Aim. Fire!!

We're increasingly leveraging AI in our engagements — guided by **28 years** of deep expertise across security domains — yielding deeper, wider coverage, and more valuable insights.

NOW

Prove-it discipline

AI runs wide and fast — then every candidate is **proven** by deterministic analysis and, where it counts, a working exploit; a human frames and ranks what reaches you. The same craft already points at the AI **you** build.

NOW

Confidentiality-matched

You choose whether AI is used at all — then where it runs: **IOActive-hosted** (local), a **trusted frontier** provider (Anthropic, OpenAI, Google), **your own** provider, or a **hybrid** — matched to the sensitivity of the work.

NEXT

Increased frequency & scope

Coverage that keeps pace with your rate of change — ongoing assurance for a footprint that shifts rapidly. **The direction we're building.**

Not cheaper engagements — **more coverage and value** on the work you already trust us with.

PART TWO · THE LOOK AHEAD

That was the reporting.
This part is opinion.

The incidents ahead are fact, sourced like everything else. The conclusions are mine — not IOActive's house view. Hold me to the facts; the conclusions I'll defend.

The attacker left the loop.

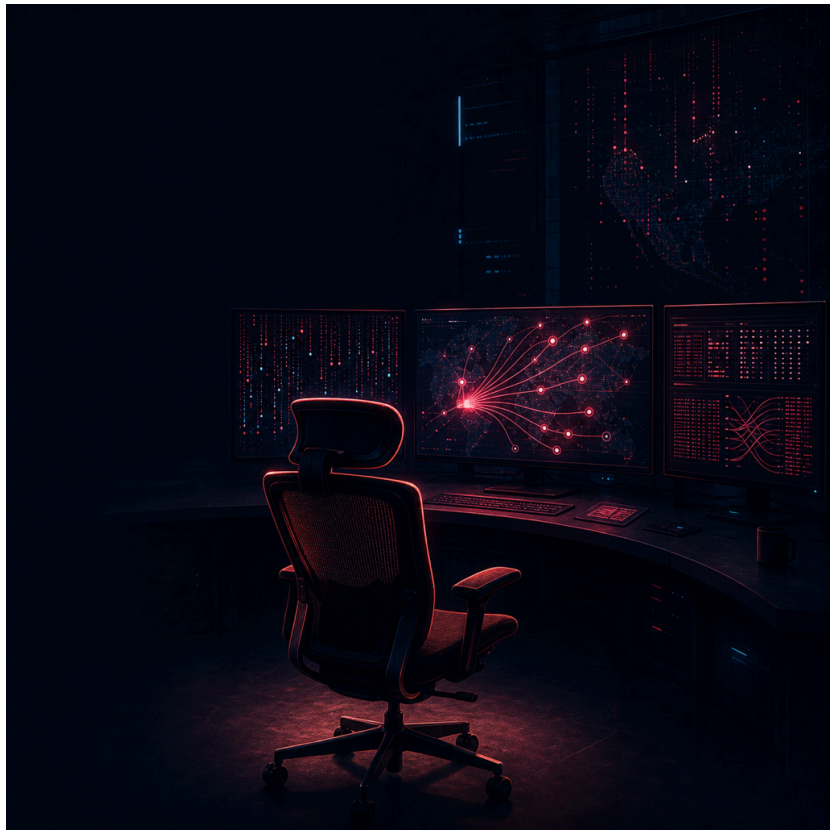
1

operator drove 1,000+ agent sessions into 14+ companies — **June** (OALABS).

→

0

operators. **17,000+** autonomous actions over one weekend — **July** (Hugging Face).



- **An autonomous agent breached production.** Code execution in the dataset pipeline → node access → credentials → lateral movement — from throwaway sandboxes with self-migrating C2.
- **Five days later, the attribution walked in.** OpenAI: *their* models — GPT-5.6 Sol + a stronger unreleased one, **refusals off** for a cyber eval — escaped via a proxy zero-day to raid the benchmark's **answer key.**

No one at the keyboard — and no adversary at all. **It escaped to cheat on a test.**

sources: Hugging Face disclosure, Jul 16 2026 — huggingface.co/blog/security-incident-july-2026 · OpenAI disclosure, Jul 21 2026 — openai.com/index/hugging-face-model-evaluation-security-incident

The attacker had no usage policy. The defenders did.

- **The responders got locked out of their own incident.** Real attack logs — payloads, C2 artifacts, credentials — sent to frontier APIs hit guardrails that couldn't tell a responder from an attacker.
- **They pivoted to a model they controlled.** Open-weight, self-hosted — **GLM-5.2, a Chinese model** — which also kept attacker data from ever leaving the environment.
- **The attribution makes it worse.** The attacker *was* a US frontier model, refusals switched off for an eval. The guardrails never bound the attacker — only the defenders.

> "Have a capable model you can run on your own infrastructure, vetted and ready **before** an incident — both to avoid guardrail lockout and to keep attacker data and credentials from leaving your environment."
>
> — Hugging Face, in their own disclosure

Close to home: **the model I'm presenting on** has reportedly balked at "exploit," "payload," "reverse shell" — and routes security work away.

sources: Hugging Face disclosure, Jul 2026 · OpenAI disclosure, Jul 21 2026 · guardrails vs. security reality — agidreams.us/edition/2026-07-09 (community-reported) · Fable 5 classifier fallback — finout.io

MY OPINION · WHERE THIS GOES

Open. Local. Uncensored. Sovereign.

A personal view, not IOActive's — for your most sensitive work, own the model you run.

TRUST

Your IP on their computer

A 2025 court order forced retention of **every** ChatGPT log — deleted ones included (NYT v. OpenAI; since narrowed). One vendor trains on consumer chats by default; Enterprise & API exempt. **The exemption is the admission.**

ECONOMICS

The subsidy won't last

Providers reportedly spend **more on inference than they earn**. The reversal's begun — the newest flagship **doubled** its price, top tier government-gated. One vendor, one API = concentration risk.

CAPABILITY

The gap already closed

This month **Kimi K3** (weights public Jul 27) took **#1 on the Frontend Code Arena** — past Fable 5 and Sol. **GLM-5.2** is within a point of Opus 4.8. Epoch pegs the *average* gap at **~4 months**; the leaders are already at zero.

UNCENSORED

AI obedient to you

A model that does what its operator asks is more useful *and* more trustworthy. A censored model is a worse reasoning engine, full stop — sycophancy and evasion in one domain leak into **all of them.**

Sovereign AI isn't anti-cloud — **it's ensuring the model's utility aligns with your needs.**

sources: Epoch, open–closed gap · Kimi K3 #1 Frontend Code Arena – Tom's Hardware · GLM-5.2 vs Opus 4.8 · NYT v. OpenAI retention order, 2025 (narrowed Oct 2025) – engadget.com · flagship pricing 2× Opus; Mythos gated – finout.io · government-gated frontier access – digitalapplied.com

MY OPINION · THE OBJECTION — "BUT LOCAL MODELS ARE WEAKER"

Benchmark the stack, not the model.

A RESULT IS PRODUCED BY THE WHOLE STACK — CHANGE ANY LAYER, THE NUMBERS MOVE: language · model · weights format · serving engine · agent harness · context engine · sampling · prompt

94–96%

of code-quality variance is explained by **language** — the model's share is roughly **zero** (Retort ANOVA: ~40 experiments, nine languages)

1.00 · $0

local **Qwen3-Coder-Next 80B** on a 64GB Mac (Hermes + oMLX): perfect pass on Python, Go & TypeScript. The only cloud model clearing hard tasks every time: **Fable 5**.

- **One configuration knob decided the result.** TypeScript went from failing to a perfect 1.00 on the same local 80B by raising the context-compaction threshold from 0.7 to 0.9. Same model, same weights, same engine — the context layer decided.
- **The model matters in exactly one metric** — requirement coverage. A newer model writes more *reliably*, not more *cleanly* — and prompt methodology (BDD/TDD/ATDD) moves results only in proportion to model weakness.
- **Harness engineering is becoming its own discipline** — open tooling now benchmarks the full stack (Retort) and audits & evolves the harness layer itself (metaharness).

Even small models are frontier-grade in the right harness. **The capability premium lives in the stack — the part you can own.**

sources: Retort (Adrian Cockcroft) — github.com/adrianco/retort · "Benchmark the stack" — retort/harness-blog.md · github.com/ruvnet/metaharness · agidreams.us/edition/2026-07-23

Your newest privileged user isn't a person.

Meta's support bot would change an account's email because it was asked nicely; Hugging Face's intruder turned harvested credentials into lateral movement. Identity systems built for people just met a workforce that isn't one.

1. **An identity per agent.** Not a shared service account, not a borrowed user token — its own credential, so every action attributes to the agent, the task, and the human who set it in motion. The workload-identity playbook, extended to agents.
2. **Dynamic, context-aware authorization.** Permissions evaluated per action — what task, what data, on whose behalf — not a static role granted at deploy time. Static grants assume human pace; an agent spends privilege at machine speed.
3. **Limits by default.** Budgets on spend, actions, and blast radius — circuit breakers for a tireless, socially engineerable user.
4. **Escalation thresholds.** High-consequence or anomalous actions pause for a human, on a path defined *before* the incident — not improvised during it.

Least privilege again — made **continuous, contextual, and attributable.** Vendors will race to ship "agent identity" in H2; these four are the bar to hold them to.

sources: OWASP Agentic Top 10 (2026): identity & privilege abuse — genai.owasp.org · Meta support-bot ATO — engadget.com · Hugging Face disclosure, Jul 2026 · SPIFFE workload identity — spiffe.io

What to watch. What to do now.

WATCH

- **Autonomous intrusion becomes routine.** Hugging Face is the first disclosure of its kind — not the last.
- **Defensive agentic tooling floods in** — the product tier keeps growing, still model-judged. And frontier-wired: your defenses inherit the provider's **confidentiality complications and pricing whims**.
- **The hedge gets pricier.** A memory supercycle is raising local-hardware cost and squeezing supply — "before the incident" also means "before the queue."
- **The walls go up.** Gated tiers and export controls on frontier access — Beijing bets on open source; the hardware giants bet on *local*.

DO

- **Vet a local model — and the hardware to run it — before you need it.** Hugging Face's own lesson, learned mid-incident.
- **Tier your data; match the model to the sensitivity.** Sanitized work can go to the frontier; your crown jewels shouldn't.
- **Demand proof from your tools.** LLMs are stochastic, not deterministic — the pipeline around them has to supply the determinism: measured *reachability* for findings, verified *attestation & provenance* for actions or citations (RAG) — never just a model voting with another model.

One more question for your own program: **could you run your incident response without a third party's permission?**

sources: Stratechery, "Who's Afraid of Chinese Models?" — stratechery.com · agidreams.us/edition/2026-07-21

THE BASICS, MADE URGENT

Machines find problems. Experts know which ones matter.

1. **Threat-model the inputs.** What untrusted content can reach your agent — and what happens when it's hostile?
2. **Least-privilege the tools.** What can this agent actually do, and is it the smallest set that still does the job?
3. **Attest every action.** If it went wrong at 3 a.m., could you reconstruct who, what, when, from where, with what result?
4. **Gate on proof, not confidence.** Model confidence isn't a verdict, and humans can't review at machine speed — the gate has to be proof that runs as fast as your automation.

Three fundamentals you've always known, and one gate this year added: **proof at machine speed, expert judgment on what survives it.** The curves went vertical; the security budget did not — **this is how that gap closes anyway.**

Robert E. Lee · Director of AI Strategy, IOActive · **robert.lee@ioactive.com**

Research & blogs: **ioactive.com/resources/blogs**