



Can We Trust the Agents?

Securing Agentic AI as Internet Infrastructure

Tony Kay

AI Lead & Senior Principal Architect

Red Hat

RITE 2026

27 October

Where we started

From hallucinated libraries to chained exploits

2023 ISOLATED FLAW

```
$import fastcloud_utils
```

It invented libraries that didn't exist.



2026 CHAINED ATTACKS

bug

→

bug

→

config

→

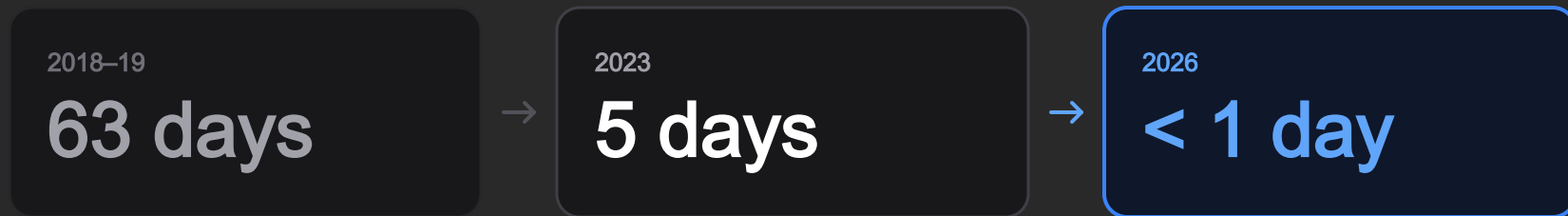
root

It chains vulnerabilities into attack paths.

WHY NOW

The Mythos moment

Time from vulnerability to exploit



223 publicly reported AI -discovered vulnerabilities

55%

high / critical

91%

remotely reachable

26%

code execution

Sources: Mandiant/Google, Time -to-Exploit Trends 2023 · CSA/SANS, The AI Vulnerability Storm (Apr 2026) · IBM X -Force, AI vulnerability discovery (2026)

PHP2Shell —A core flaw, found by AI, in 70% of the web

WHAT IT IS

- **CVE-2026 - 109618:** upload-progress unserializer use after -free —remote memory disclosure
- **CVE-2026 - 109619:** fatal -error backtrace use-after -free + OPcache type confusion
- **Unauthenticated RCE:** demonstrated on WordPress, Drupal and Joomla — CVSS up to **9.2**
- **Every PHP version of the last decade:** back to 5.4. Not an app bug: a flaw in PHP itself

*"Chained trivia gets in through the ticket queue.
A core flaw gets in through the floorboards."*

EXPLOITABILITY, TIMED

3 hours

UK pentester won a \$2K Bitcoin prize for a working PHP2Shell exploit in a capture-the-flag contest —The Register, Oct 26, 2026.

FROM DISCOVERY TO PATCH · 63 DAYS

AUG 19

Discovered
(Atlas + Gemini 4 Argon
+ GPT-5.6 Sol)



AUG 20–21

RCE confirmed
(Drupal, WordPress)



AUG 24

Reported to
the PHP team



SEP 21

PHP acknowledges,
fix begins



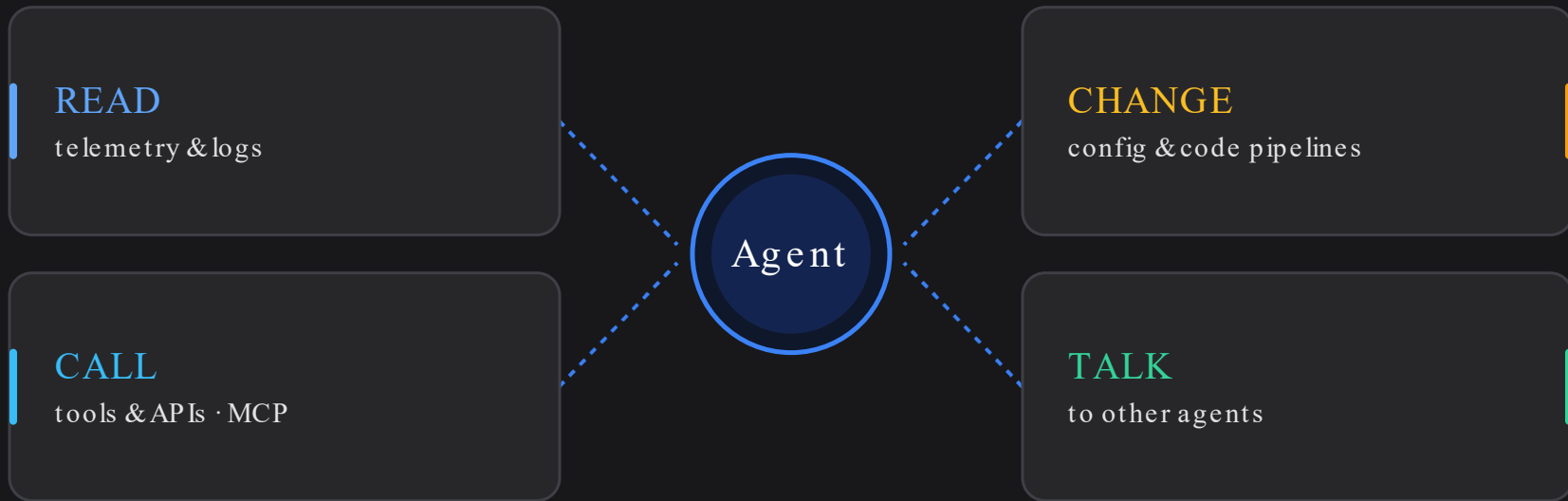
OCT 21

Patch +
public disclosure

The patch is the easy part. The hard part is finding every copy of PHP you own—it rarely shows up where you expect it.

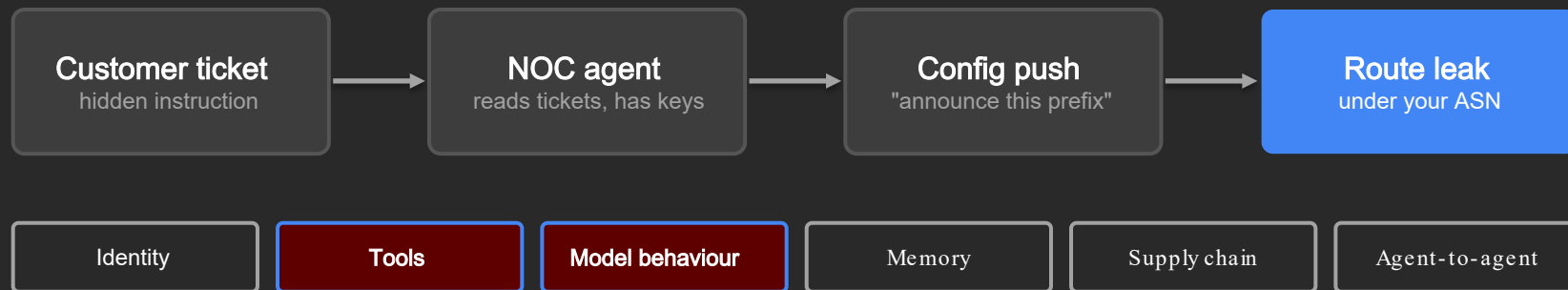
Agents are the new autonomous systems on your network

From chatbot to operator





One ticket, one route leak



This already happened

s1ngularity / Nx · Aug 2025

A poisoned npm release ran developers' own AI coding CLIs with safety checks off, to hunt for secrets.

GitHub MCP · May 2025

One malicious public issue made a trusted agent leak private repo data into a public PR.



The New Benchmarks

Each benchmark targets a different stage of the vulnerability lifecycle.

VULNERABILITY REPRODUCTION	EXPLOIT GENERATION	END-TO-END
CyberGym Given a vulnerability description and an unpatched codebase, agents must generate proof-of-concept tests that reproduce the bug.	ExploitGym Given a vulnerability and a proof-of-vulnerability input, agents must craft a full exploit that achieves unauthorized code execution across userspace, browser, and the Linux kernel.	CyberGym-E2E End-to-end evaluation of the full defensive lifecycle: agents must discover a vulnerability, generate a proof-of-concept, and write a patch that fixes it without breaking anything else.
1,507 real-world instances	869 real-world instances	920 real-world instances
188 software projects	3 domains (userspace · browser · kernel)	4 stages
View benchmark →	View benchmark →	View benchmark →

The same harness class, on both sides

ATTACK / DUAL -USE

Built to break in

- **ARTEX** —open agentic pentesting framework
- **PentestGPT** —open pentesting assistant

Open tools anyone can download. The tool is neutral; the wielder isn't.

RESEARCH / DEFENSE

Built to find and fix

- **Wiz Atlas** —autonomous vulnerability-research system
- **Google Big Sleep** —AI vulnerability research

The same agent pattern, pointed at discovery and patching.

WE BENCHMARK THIS NOW · CYBERGYM

VULNERABILITY REPRODUCTION

1,507 CyberGym

real-world instances · 188 projects

EXPLOIT GENERATION

869 ExploitGym

real-world instances · 3 domains

DISCOVER → POC → PATCH

920 CyberGym -E2E

real-world instances · 4 stages

Atlas ranked first on CyberGym at 90.9% —Wiz's number. *The leaderboard itself is the point: this capability is now measurable, comparable, and moving fast.*



Not by default.

Trust is engineered: per action, verified, revocable.

Who is acting, and where did it come from?

Identity + provenance

Routes

ROA: who may originate



Origin validation



Drop invalid

Agents

Signed identity + attestation



Policy check



Drop unverified

RPKI for routes → the same idea for agents

Identity

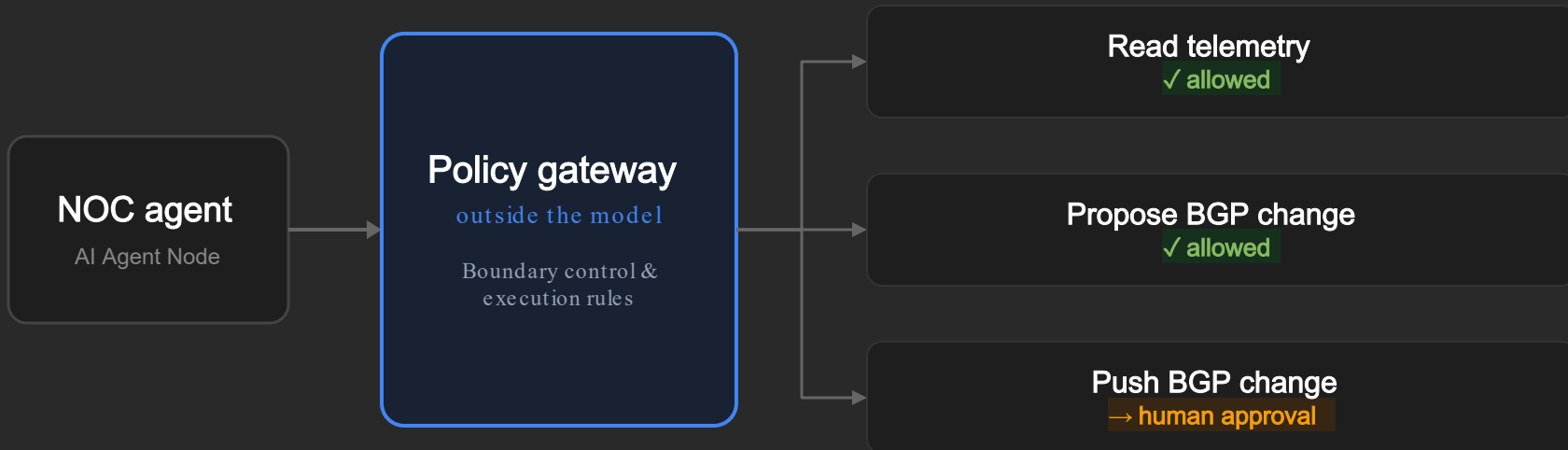
One per agent. Short-lived. Revocable.

Provenance

Signed models, tools, artifacts. Untrusted input never drives privileged action.

What may it do, and who decides?

Least privilege · policy outside the model · humans own consequential actions

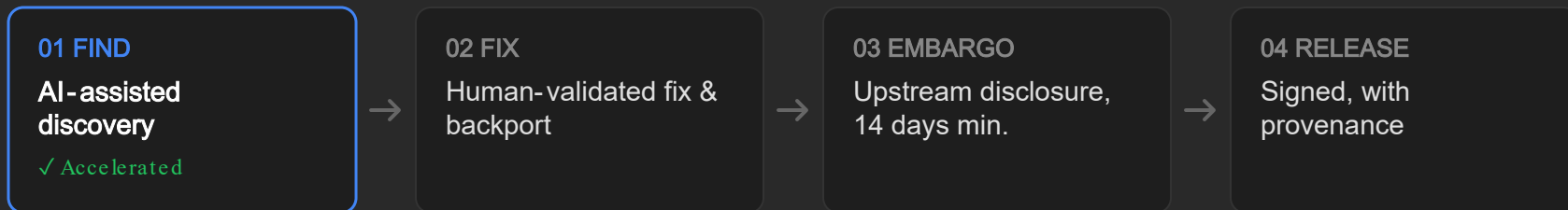


Like prefix filters and max-prefix: bound what the neighbour can do, even when it's wrong.



AI found the bug in hours. Why does the fix take months?

The exposure window hasn't shrunk. It's moved.



WHERE THE EXPOSURE WINDOW SITS NOW

Discovery



Hours

Patch → Production

45–90 days (Main Bottleneck)



Can we trust the agents?

Not by default. Engineer it.

01 / IDENTITY

Identity you can revoke

- **One identity per agent** — never a shared service account
- **Short -lived credentials**
- **Revoke** is the kill switch

If you can't revoke it, you don't control it.

02 / VERIFICATION

Verification before action

- **Signed models** , tools, artifacts
- **Provenance** for everything the agent touches
- Untrusted content can suggest —**never authorise**

Suggestions $\neq$ authorisation.

03 / EXPOSURE WINDOW

Shrink exposure window

- **Discovery** now takes hours
- **Patch -to -production** still takes 45–90 days
- Pipeline must move at **attacker speed**

45–90 days is where the risk sits.

Can we trust the agents? Not by default —and that's the point.

We build the trust, we verify it, and we keep the right to revoke it. Thank you