Beyond Mythos

A CISO's guide to building an effective software security program for the AI era

Robert Haynes
Principal Technical Marketing Engineer

ENDOR LABS

Table of Contents

Executive summary

In April 2026, Firefox shipped version 150 with fixes for 271 vulnerabilities — all found by Anthropic's Claude Mythos Preview in a single evaluation run. The previous release, evaluated with Opus 4.6, turned up 22. Twelve times more vulnerabilities in one of the most hardened codebases on the internet.[1]

Anthropic's Claude Mythos grabbed the world's attention when Anthropic disclosed that the model had autonomously discovered thousands of zero-day vulnerabilities across every major operating system and browser, including a 27-year-old flaw in OpenBSD.[2]

They followed shortly after with the announcement of Project Glasswing, an initiative to harden critical national security infrastructure. Further evidence was supplied by an independent evaluation from the UK AI Security Institute (AISI) that confirmed a real step: Mythos was the first model to complete a 32-step corporate network attack simulation end-to-end, and hit 73% on expert-level capture-the-flag tasks — no model before April 2025 could finish one.[3]

What Mythos is (and isn't)

The buzz around Mythos represents a heady mix of marketing and model. It has brought the potential use of large language models by adversaries to the attention of leaders at the highest level. This is a good thing, but this is just a significant, but not exponential, acceleration in a race that is well underway already.

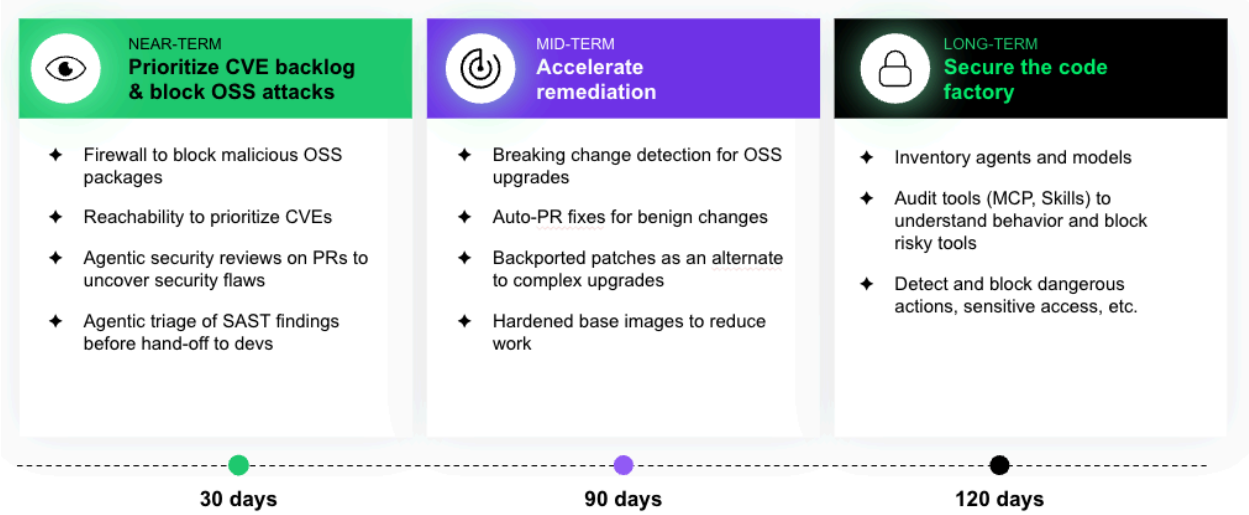
The trajectory has been clear for months or years: XBOW topped HackerOne in June 2025; Big Sleep found 20 real-world zero-days in August 2025; DARPA AIxCC cleared 54 vulnerabilities in 4 hours across 54M lines of code; AISLE + Opus 4.6 produced 500+ findings in February 2026.[4] Mythos is just a predictable step up in capability.

Threats are already accelerating. Our own tracking shows that in the first 100 days of 2026, we logged as many CVEs as in all of 2025 combined.[5] CVE and NVD infrastructure was built for dozens of critical CVEs per month. It is now seeing hundreds. Quarterly pen tests and SLA-based patching cycles were designed for a world that stopped existing sometime in 2024. The new breed of self-propagating, credential-stealing worms spread through build system components and third-party dependencies has caused significant damage in the last month alone.

The economics of exploits are changing. Mythos might represent the (expensive) cutting edge of research-grade exploits. Still, the fact that cheap, open-weight models can replicate much of Mythos's showcase analysis once code is isolated is shifting the cost of exploits lower and lower (although simpler models cannot link exploits together into an attack chain).[6]

The fundamentals remain. The controls that worked in March 2026 still work. What's changed is the tolerance for deferring them.

What we recommend



Make three commitments, each with a named owner and a target metric.

This week: change the organization's risk metric from open CVE count by CVSS severity rankings to *reachable, exploitable, unfixed* CVEs. This alone reprioritizes most existing programs.

This quarter: treat the developer environment as production. Inventory the AI layer (coding agents, MCP servers, skills, agent identities). Scope each agent's blast radius. Put a package firewall at the developer edge. This is a critical control that most organizations still lack. Ensure simple controls like 'cooldown periods' for new packages are enforced.

This year: build to scale. Stand up a VulnOps function and an automated remediation pipeline. Detection-to-production in 4-6 hours is achievable with modern tooling; the work is engineering, not research.

This is not the end of the world

The Mythos announcement must be seen in context. It represents an improved model that happens to be good at finding and chaining vulnerabilities to create advanced exploits. But many of the headline capabilities are not new, and focusing on the specifics of Mythos would be a mistake. The noise about Mythos is both a signpost and an opportunity to properly address the multiple threats that the use of AI internally and by attackers has magnified.

This is not the end of the world, but it is a message written in thirty-foot-high letters of fire that the time to defer a comprehensive security program to secure the code you write, the dependencies you use, and the systems that build it, is long, long past.

The rest of this report provides our advice on building a new operating model for AppSec.

What's actually different (and what isn't)

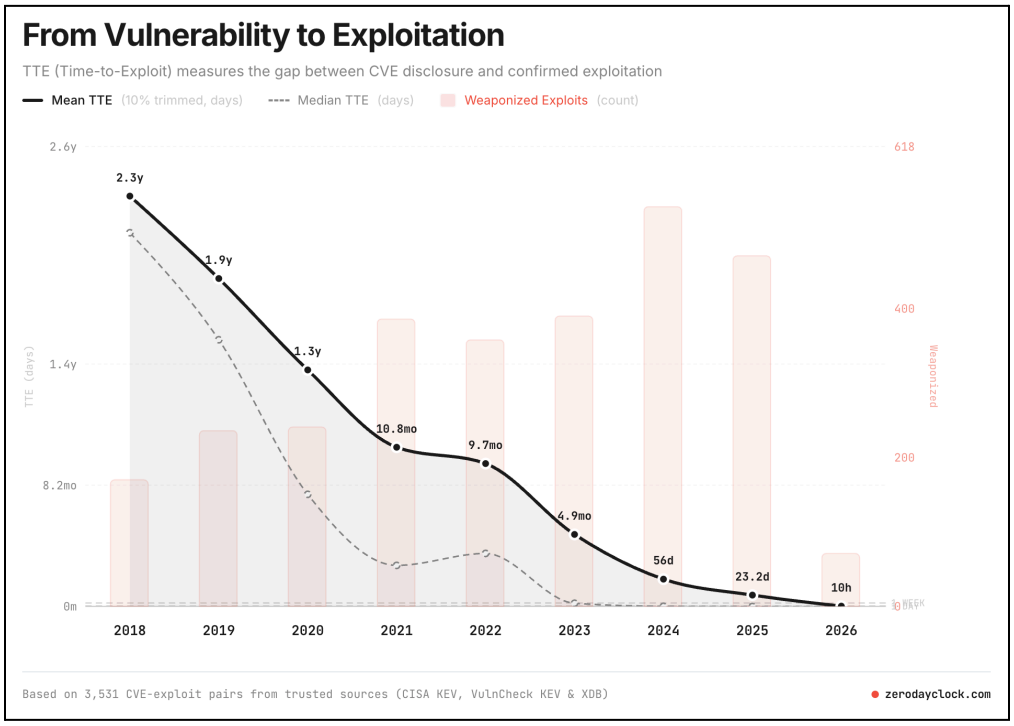
Four forces were pushing AppSec past its breaking point *before* Mythos landed.

More code is shipping than any AppSec program can review. AI coding agents have driven pull-request volume up 26% or more across most engineering organizations. Less than 20% of AI-generated code is both functionally correct and secure.

Same headcount, more code, faster, written by tools with a known defect rate.[7]

The CVE explosion continues. In the first 100 days of 2026, we logged as many CVEs as in all of 2025 combined. CVE and NVD infrastructure was built for dozens of critical CVEs per month; it is now seeing hundreds.

Exploits are cheap and fast. Peer-reviewed research in late 2025 found that attackers can generate working exploits for more than half of CVEs at an average cost of $2.77 in model compute. The Zero Day Clock shows mean time-to-exploit falling from 2.3 years in 2018 to 9.7 months in 2022 to roughly 20 hours in 2026. This shift predates Mythos by a year.



Supply-chain attacks went vertical. 2025 was the worst year on record for open-source supply-chain attacks — a 14× increase in malware advisories, with 92% of all recorded npm maintainer account takeovers happening in 2025 alone. Campaigns like NX S1ngularity and the

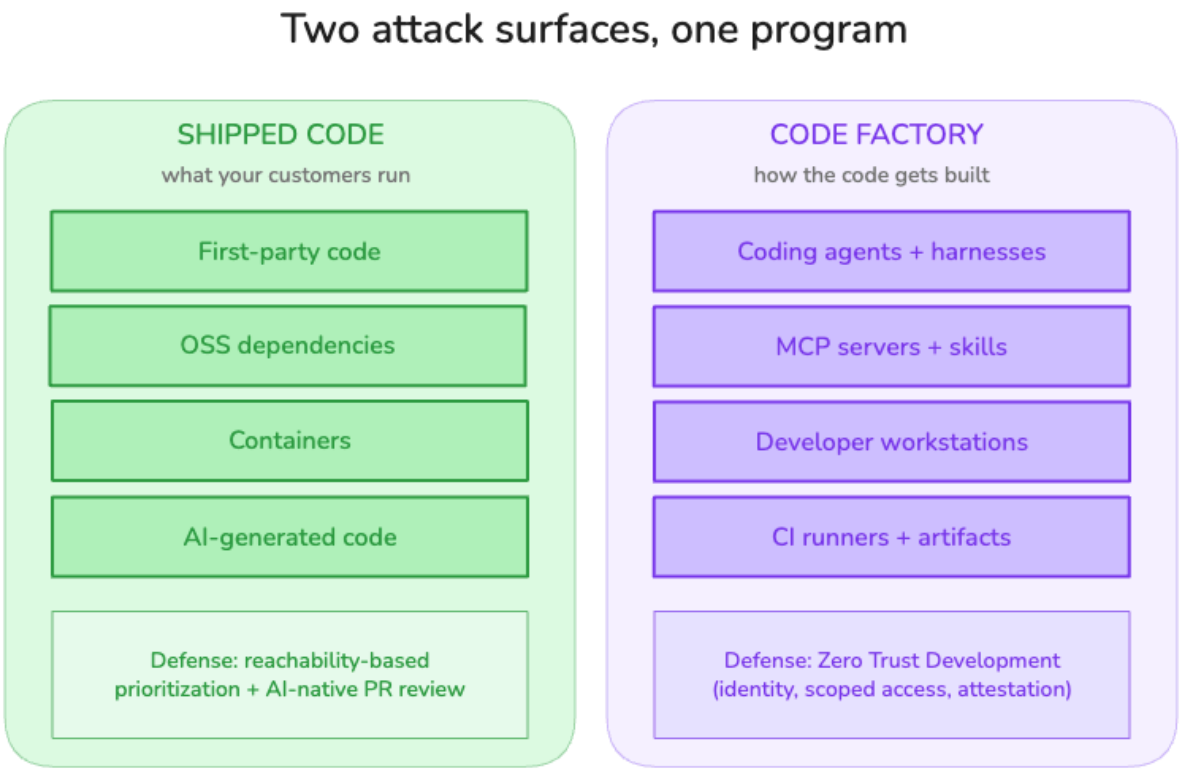
Cline compromise increasingly target AI coding agents directly, poisoning the input to the factory rather than breaking the output.[8]

The blast radius now includes the code generation factory

The role of AppSec used to be simply to protect the code. Now it has to protect the systems that produce the code. Every coding agent, every MCP server, every developer laptop running Claude Code or Cursor, every CI runner — these are part of the production attack surface, whether your security program treats them that way or not.

A compromised coding agent has full developer credentials, read access to the entire repository, commit rights, and the ability to open pull requests, the ability to call tools, plugins, and MCP servers with those permissions, and machine-speed execution outside business hours without the friction of a human attacker.

Most security programs have zero inventory of which models their developers use, which MCP servers are installed, which skills are active, or which agents have repo write access. This is the same maturity level AppSec had for cloud IAM in 2017. We all know how that turned out.



The current defender advantage is temporary

Bruce Schneier's observation stands: finding-for-fixing is currently easier for AI than finding-plus-exploiting.[9] AISLE's replication work shows the same effect from the other direction — cheap open-weight models already recover most of Mythos's showcase findings, but lack the scaffolding to chain exploits end-to-end. That gap will close. Build the scaffolds now, while the advantage favors you.

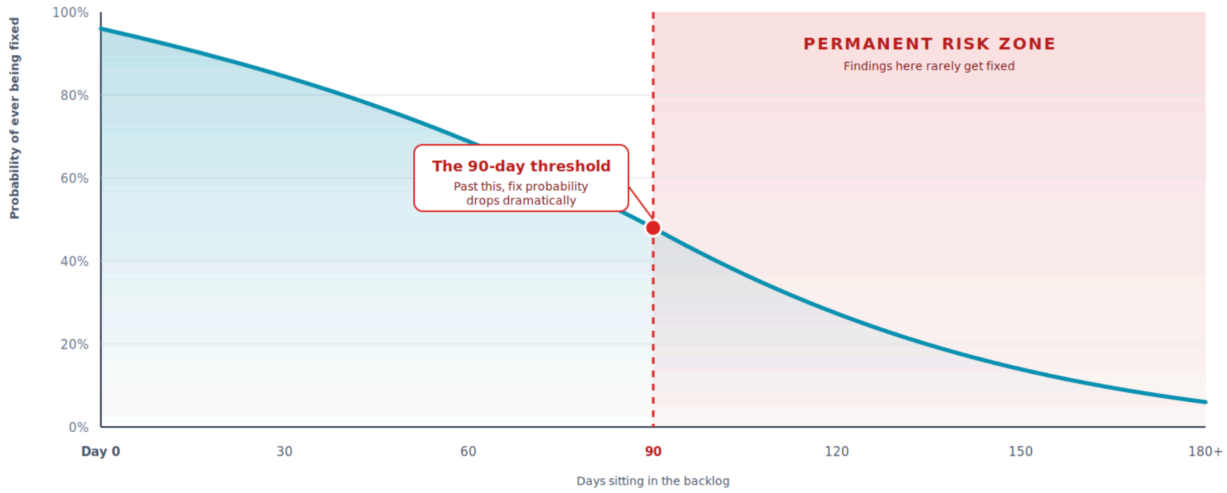
What to do about it

Prioritize the backlog — this week / first 30 days

When the CVE queue doubles, the old answer, “patch everything critical” fails silently. Prioritization stops being a nice-to-have. It becomes the only sane unit of work.

Act on reachability. A reachable medium-level finding is a “fix-now”; an unreachable Critical is backlog noise. Making that distinction consistently is the single biggest leverage point in AppSec today.

1. **Triage ruthlessly, early to beat the 90-day cliff.** Findings that sit longer than 90 days are dramatically less likely to ever be fixed. If it isn’t closed in a quarter, you’re not going to close it, which means you’re accumulating permanent risk silently.
2. **Put AI-native review in the PR path, not just on full-scan.** PR-detected findings resolve faster than full-scan findings, as they are integrated into a human-powered approval process, with clear accountability. The context is fresh for the developer, the fix ships in the same PR, and there’s no ticket-shuffling from an upstream security-owned scan.



Actions — first 30 days

- **Change the risk metric you report.** Move from open-CVE count and scan coverage % to reachable, exploitable, unfixed CVEs plus MTTR in hours. Owner: CISO. Target: exec scorecard updated before the next board meeting.
- **Rebuild triage around reachability.** Every Medium+ finding tagged reachable / not reachable at ingestion. Owner: VP AppSec. Target: 95% of open findings carry a reachability verdict within 30 days.
- **Put AI-native review in every PR.** Options include Claude Code Security, Semgrep Assistant, or a custom pipeline. Owner: VP AppSec with VP Engineering. Target: coverage on all merge-protected branches.

- **Deploy canary tokens.** Fake AWS keys, API tokens, decoy MCP endpoints. AI attackers enumerate exhaustively — a single canary hit is a high-confidence breach signal, not noise.
- **Start the AI-layer inventory.** Begin cataloging every coding agent with repo access, every MCP server, every skill, every agent identity. Full completion is a Part 3 item; starting it this week is non-negotiable.

Secure the code factory — this quarter / 30–90 days

Once you've stopped drowning in the queue, the next leverage point is the producer side. This is new work for most AppSec teams. It cannot wait.

Five moves, in order

3. **Complete the AI-layer inventory.** Every approved LLM endpoint. Every MCP server in use. Every skill, plugin, and agent with repo access. You cannot secure an asset class you haven't cataloged. Owner: AppSec platform lead. Target: 100% of agents with repo-write have a named owner and a defined scope.
4. **Put a package firewall at the developer edge.** Block malicious packages before they land on a workstation. Given that 92% of all recorded npm maintainer takeovers happened in 2025, if you aren't blocking at the edge, you have almost certainly lost developer credentials to a supply-chain attack — it's just a matter of time before you find out which ones. Enforce cooldown periods for new package versions. Owner: Platform Engineering. Target: 100% of package installs flow through a controlled registry.
5. **Scope the blast radius of coding agents.** Agents get the minimum permissions needed — not org-wide IAM, not the same credentials as their human developer. Define scope boundaries, pre-authorized actions, and human-override mechanisms before deploying agents anywhere adjacent to production. Owner: identity/platform. Target: zero agents with org-wide write access.
6. **Audit the agent harness with production rigor.** Prompts, tool definitions, retrieval pipelines, and escalation logic are where the most consequential failures now occur. Treat them like production service code: versioned, reviewed, logged, rollback-ready. Owner: AppSec + ML platform.
7. **Treat the developer workstation like a production system.** Same EDR. Same logging. Same IR playbooks. In an agent-first world, the workstation is a production system.

Plus two governance items that block the rest of the plan

- **Prepare for continuous patching.** Glasswing-class disclosure programs will push critical patches in waves. Stand up triage and deployment capacity now — not after the first wave hits.
- **Establish an innovation-acceleration governance mechanism.** A joint body across Security, Legal, and Engineering with authority to approve AI-layer tooling within days, not quarters. Without it, the rest of this plan stalls at approval.

Automate remediation; rebuild the operating model — 90–365 days

Prioritization and factory controls buy time. They don't close the exploit window. The only structural answer to 20-hour time-to-exploit is automated remediation that runs faster than your adversary's exploit pipeline. Two halves — tooling and team — and you need both.

The tooling to close the exploit window

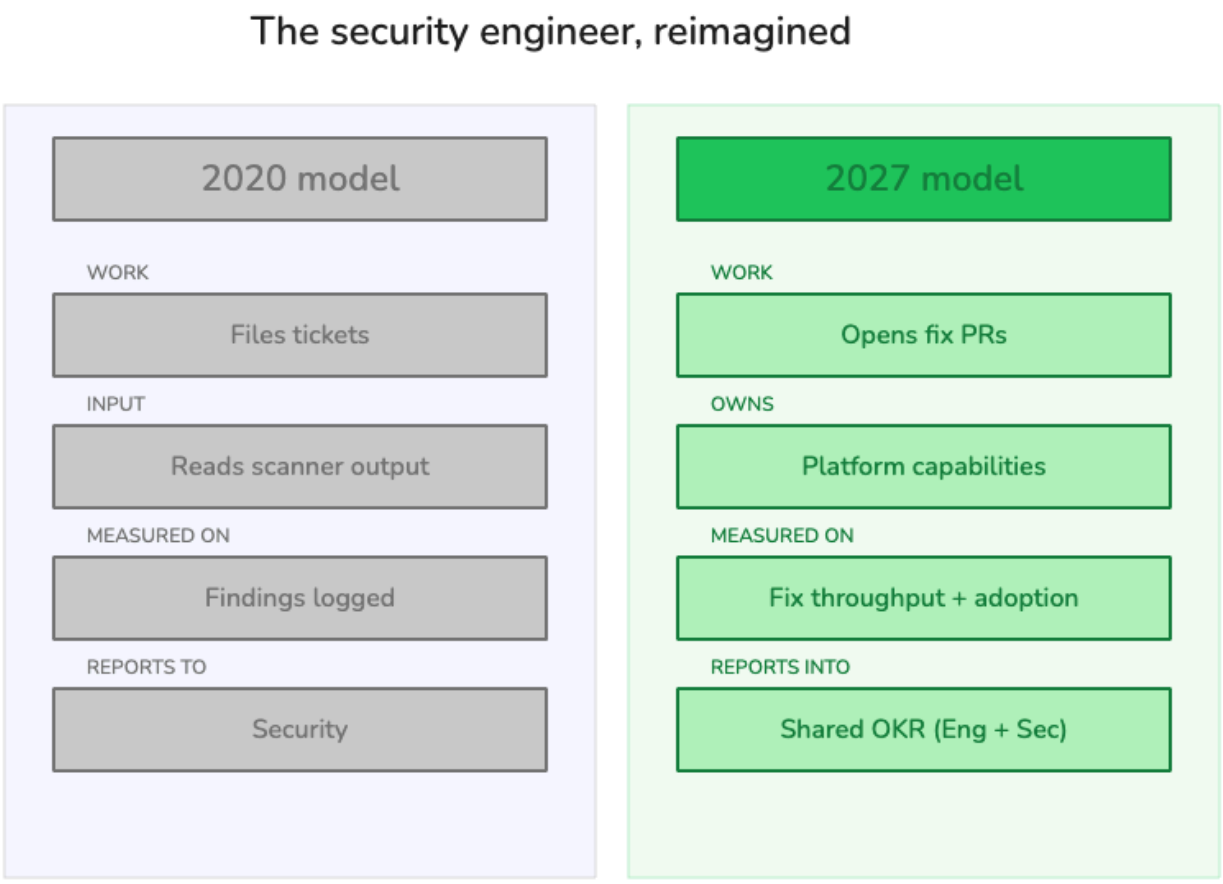
- **Upgrade Impact Analysis.** Score every upgrade by risk and identify breaking changes before applying. Stop interrupting developer flow for every CVE; upgrade with evidence. This alone eliminates most of the "we can't safely patch" backlog in large estates.
- **Backported patches.** When a full version upgrade isn't operationally realistic — which is most of the time in large estates — minimal, hermetic, reproducible backports resolve the vulnerability without changing anything else in the library.
- **Minimal, hardened base images and library replacements.** The most effective CVE management strategy is to have fewer CVEs in your estate in the first place.[11] Minimal OS images, framework primitives in place of bloated third-party libraries, continuous rebuilding. Reduce what you even have to patch.
- **Full call graphs for agentic remediation.** If remediation is handed to Claude Code, Cursor, or Devin, the quality of a code fix depends entirely on the context the agent has. Call graphs tell the agent whether a function it relies on has been renamed, removed, or had its API changed — so fixes ship correctly the first time, not as broken PRs.

Detection-to-production in 4-6 hours is the bar. Current tooling can hit it. The work is engineering, not research.

The operating model to rebuild the function

For twenty years, AppSec has operated as a ticketing function: scan, triage, file Jira, chase, escalate, repeat. That model assumed security and engineering had different clocks, tools, and goals. Twenty-hour time-to-exploit doesn't tolerate a Jira-mediated loop.

- **Shared OKRs, not parallel ones.** Joint metrics on both the CISO's and the VP Engineering's scorecards — reachable backlog, median MTTR on high-severity, coverage of critical services. Not separate metrics feeding separate reviews.
- **Security engineers who ship code, not tickets.** The 2027 profile of a senior AppSec engineer looks more like a software engineer with deep security specialization than a security analyst reading scanner output. Tools, context, and organizational permission to open PRs that fix findings — reviewed by the owning engineering team, but originated by security.
- **AppSec as a product team, not a program office.** Your AppSec team becomes a Platform team with an internal customer base. Ships internal tools and APIs. Measures adoption across engineering.



AppSec becomes a product team. Its customers are engineers. Its output is shipped code.

Metrics that measure risk now

The scorecard that the board reviewed 18 months ago no longer measures your actual risk. The shift, for the CISO deck:

Pre-AI	Mythos-ready	Why it matters
Critical + High CVE count	Reachable, exploitable CVEs	Signal, not noise
Open vulnerability backlog	Fixes shipped via PR	Outcome, not activity
SLA compliance in months	MTTR in hours and days	Speed is survival
Scan coverage %	AI-layer inventory %	New attack surface
AppSec tickets filed	AppSec PRs authored	Engineering output

New metrics worth introducing

- **Time-to-exploit gap.** Your median patch SLA for internet-facing systems, compared to the Zero Day Clock baseline. Negative is the new normal — but the size of the gap is what matters.
- **Unused access ratio.** % of IAM permissions and network paths unused in 90 days. Approximately 99% on average across cloud estates.[12] This is the single highest-yield control that keeps working when human-paced assumptions break.
- **Deception trigger rate.** A canary hit is a high-confidence breach signal, not noise. Track it at the board level.
- **Leader-cohort thresholds.** Target 40% first-party and 74% OSS fix rates — the leader-cohort numbers, not industry averages.
- **Are we using AI defensively?** Binary board-level metric. Answering “no” is now a governance and potential negligence exposure.

Further Reading: Evidence & independent assessment

Six sources, read together. Each is cited throughout this briefing; full references are in the endnotes. This section summarizes what each one proves, and, equally importantly, what it does not.

UK AISI — evaluation of Claude Mythos Preview (April 13, 2026)
Government evaluator; most credible independent source. 73% on expert-level CTFs; first model to finish the 32-step "The Last Ones" corporate-network range. AISI's own caveat: the ranges lack active defenders, EDR, and penalties for noisy actions. Don't overread.

AISLE — AI Cybersecurity After Mythos: The Jagged Frontier (April 7, 2026)
Replicated Mythos's showcase vulns on cheap open-weight models (8/8 found the FreeBSD NFS overflow; a 5.1B-active model recovered the OpenBSD SACK chain in one shot). Honest about limits: scoped context, no agentic testing, models false-positive on patched code. Implication: the capability floor is lower than the Anthropic narrative suggests. Defenders can start with what they already have.

Bruce Schneier — on Anthropic's Mythos Preview and Project Glasswing (April 13, 2026)
The sober middle ground. Acknowledges a real capability increase (chaining, one-shot exploits) while calling the framing a PR play. The defender advantage — finding-for-fixing easier than finding-plus-exploiting — is real, and shrinking.

CSA / SANS / [un]prompted community — Mythos-Ready Security Program v0.92 (April 16, 2026)
Industry consensus document with 40+ CISO and practitioner contributors. Provides a risk register, MITRE / OWASP / NIST mapping, and priority-action framework.[13] This briefing is narrower and more opinionated; where we diverge (reachability as the primary prioritization lens; the developer factory as a first-class attack surface), we say so.

Anthropic — Mythos Preview announcement and Project Glasswing (April 7, 2026)
Primary source. Anthropic's own characterization has been the subject of the skeptical commentary above. Read it alongside the independent evidence, not instead of it.

OpenAI — Why Codex Security Doesn't Include a SAST Report (April 2026)
A frank discussion about where SAST tools and AI code security tools intersect and overlap. The authors of this briefing paper agree that effective detection relies on more than semantic scanning (which still has a place) and requires additional context to validate findings. These principles are key drivers of Endor's AI SAST scanning service.

Appendix A. Timeline, June 2025 – April 2026

The trajectory wasn't hidden. Anyone treating April 7, 2026, as the start of the conversation missed the previous ten months.

Date	Event
Jun 2025	XBOW tops HackerOne leaderboard — first AI agent to rank #1 for bug bounty volume.
Aug 2025	Google "Big Sleep": 20 real-world zero-days disclosed across widely deployed open-source projects.
Aug 2025	DARPA AI Cyber Challenge finals: competing systems clear 54 vulnerabilities in 4 hours across 54M LOC.
Sep 2025	Adkins + Evron publish "singularity warning" for AppSec; open-source maintainers raise alarm.
Nov 2025	First disclosed AI-orchestrated espionage campaign; agent-driven recon + exploit chaining in the wild.
Feb 2026	AISLE + Opus 4.6: 500+ findings across audited codebases; 12/12 OpenSSL; CVSS 9.8 recovered from 1998; Sysdig environment breached to admin in 8 minutes.
Mar 2026	Open-source maintainers report overwhelmed triage; Zero Day Clock reaches <1 day on critical CVEs.
Apr 7, 2026	Anthropic announces Claude Mythos Preview and Project Glasswing. Firefox 150 ships with 271 Mythos-found fixes.
Apr 13, 2026	UK AISI publishes independent evaluation: 73% expert-CTF, first model to finish 32-step "The Last Ones" range.
Apr 16, 2026	CSA / SANS / [un]prompted community publish Mythos-Ready Security Program v0.92.

Appendix B. Top-5 risk register — condensed

Severity: all Critical. Full register with NIST CSF 2.0, OWASP LLM 2025, OWASP Agentic 2026, and MITRE ATLAS mappings available in the CSA/SANS v0.92 paper.

#	Risk	Why it matters now	Owner
1	AI-autonomous exploit generation outpaces patch cycles	Time-to-exploit ~20h vs. patch SLAs in weeks	CISO + VP Eng
2	Defender capability gap: attackers use agents; most defenders don’t yet	Asymmetric advantage compounds quickly	CISO
3	Unmanaged AI-agent attack surface inside your own environment	No inventory, no scope, no IAM control	VP AppSec
4	Detection and response are still tuned for human-speed attackers	Signatures and SIEM rules miss enumerated access	SOC lead
5	Risk model and reporting based on pre-AI assumptions	Board scorecard measures activity, not risk	CISO

Appendix C. Sources & further reading

AISLE — S. Fort, AI Cybersecurity After Mythos: The Jagged Frontier (April 7, 2026)
https://aisle.com/blog/ai-cybersecurity-after-mythos-the-jagged-frontier

UK AISI — Our evaluation of Claude Mythos Preview's cyber capabilities (April 13, 2026)
https://www.aisi.gov.uk/blog/our-evaluation-of-claude-mythos-previews-cyber-capabilities

Cal Newport — Is Claude Mythos 'Terrifying' or Just Hype? (April 13, 2026)
https://calnewport.com/is-claude-mythos-terrifying-or-just-hype

Gary Marcus — Three reasons to think the Claude Mythos announcement was overblown (April 9, 2026)
https://garymarcus.substack.com/p/three-reasons-to-think-that-the-claude

Bruce Schneier — On Anthropic's Mythos Preview and Project Glasswing (April 13, 2026)
https://www.schneier.com/blog/archives/2026/04/on-anthropics-mythos-preview-and-project-glasswing.html

Bruce Schneier on The Tech Report — "Claude Mythos is mostly 'marketing hype'" (April 9, 2026)
https://www.youtube.com/watch?v=PsKVSHjres4

CSA / SANS / [un]prompted community — Mythos-Ready Security Program v0.92 (April 16, 2026)
Community distribution — see contributors' list in the paper

Why Codex Security Doesn't Include a SAST Report
https://openai.com/index/why-codex-security-doesnt-include-sast/

Endnotes

1. Mozilla Security, “Firefox 150 security release notes — Mythos Preview evaluation,” April 2026. Counts: 271 vulnerabilities addressed in v150 (Mythos-evaluated) vs. 22 in the previous Opus 4.6–evaluated release.
2. Anthropic, “Claude Mythos Preview: model card and responsible disclosure,” April 7, 2026.
3. UK AI Security Institute, “Our evaluation of Claude Mythos Preview’s cyber capabilities,” April 13, 2026. https://www.aisi.gov.uk/blog/our-evaluation-of-claude-mythos-previews-cyber-capabilities
4. XBOW — HackerOne leaderboard, June 2025; Google DeepMind “Big Sleep,” August 2025; DARPA AI Cyber Challenge final, August 2025; AISLE + Anthropic Opus 4.6 benchmark run, February 2026. Consolidated in V. Badhwar, Beyond Mythos: A new operating model for AppSec, Endor Labs, April 2026.
5. Endor Labs CVE tracking, Q1 2026. Parallel data in the CSA / SANS / [un]prompted Mythos-Ready Security Program v0.92, April 16, 2026.
6. Peer-reviewed exploit-generation cost study, late 2025 (mean $2.77 per exploit across a large CVE corpus); Zero Day Clock dataset (3,500+ CVE-exploit pairs), S. Epp, 2026. Consolidated in Beyond Mythos, April 2026.
7. AI coding agent throughput and correctness/security rate: industry engineering data collated in Beyond Mythos, April 2026.
8. npm ecosystem telemetry, 2025 year-end; NX S1ngularity and Cline compromise incident writeups. Collated in Beyond Mythos, April 2026.
9. B. Schneier, “On Anthropic’s Mythos Preview and Project Glasswing,” April 13, 2026. https://www.schneier.com/blog/archives/2026/04/on-anthropics-mythos-preview-and-project-glasswing.html
10. Endor Labs remediation-pattern analysis across tens of thousands of actively developed repositories, Q1 2026. Reported in Beyond Mythos, April 2026. Cohort definition: top 15% by first-party + OSS fix rate.
11. Chainguard public materials on minimal container images and CVE reduction; cited in Beyond Mythos, April 2026.
12. Unit 42 cloud threat research: ~99% of granted IAM permissions unused in typical cloud estates.
13. CSA / SANS / [un]prompted community, Mythos-Ready Security Program v0.92, April 16, 2026.
14. B. Holley, Mozilla, April 2026.