



WHEN ATTACKERS MOVE FASTER

Practical Security Readiness Beyond AI Panic

By

Anant Shrivastava
Cyfinoid Research



About Cyfinoid

- Research focused cyber security firm
- Major focus areas as of now
 - Software Supply Chain
 - AI Security
 - Cloud Security

<https://cyfinoid.com/>



Anant Shrivastava

- Chief researcher @ Cyfinoid Research
- 18+ yrs of corporate exposure
- Speaker / Trainer: BlackHat, Defcon, c0c0n, nullcon & more
- Project Lead:
 - Code Vigilant (code review project)
 - Hacking Archives of India
 - Reduce the Attack Surface (<https://reducetheattacksurface.com>)
- (@anantshri on social platforms) <https://anantshri.info>



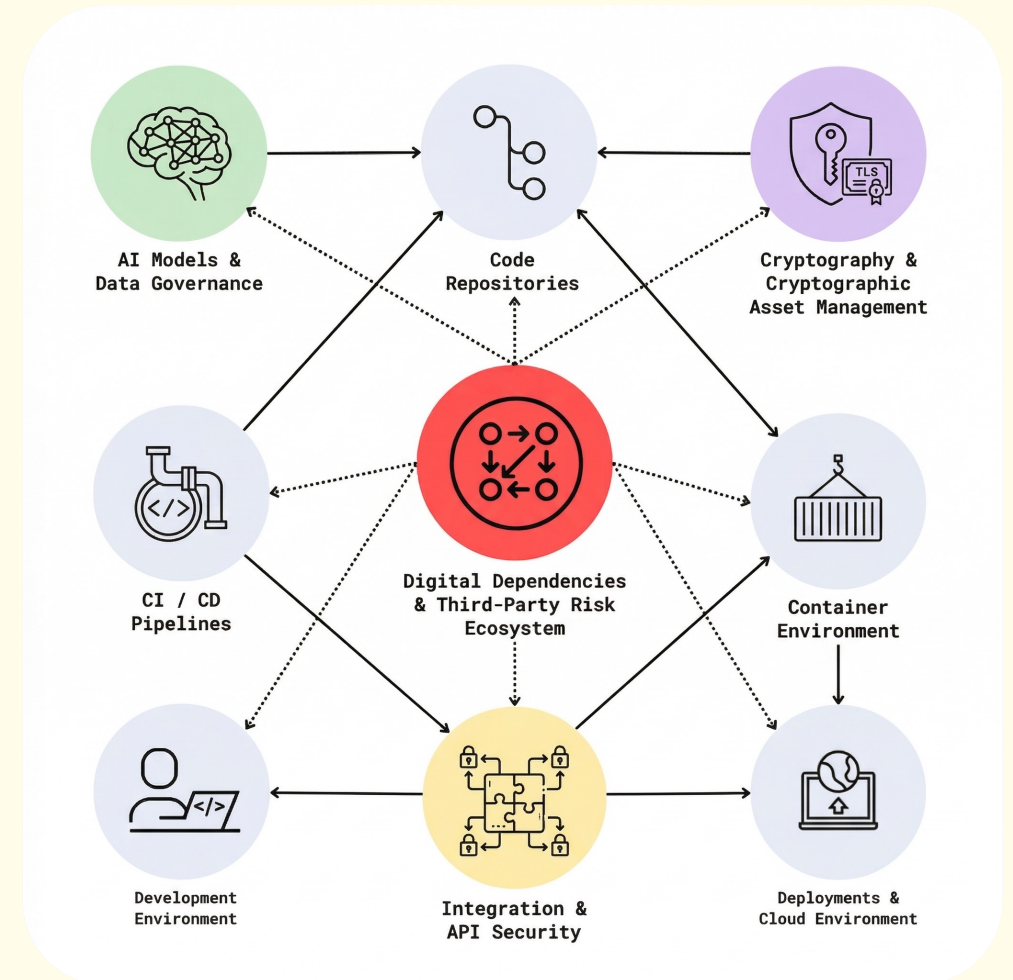
Background

- CFP was submitted in May 2026
- Initial plan talk about mythos readiness
- The world has changed a lot since then



2026 so far : Supply Chain

- Shai Hulud / Team PCP categorically proved that your development pipeline start from Desktop rather than CI system
- Developer desktop turned out to be juiciest target
- If you are curious about this visit us at Supply Chain Village Atlier 1.





2026 So far : Mythos and beyond

- Mythos Model was launched by Anthropic and immediately sanctioned by US Gov.
- Guardrails filled version was released named Fable
- Anthropic is now making it available slowly.
- Other vendors caught up
 - Sol (OpenAI)
 - GLM 5.3 (Z.AI) openweight
 - Deepseek v4 : openweight
 - Qwen 3.8 / 4 : openweight + smallest
 - Kimi K3 and more





2026: AI is the hacker now

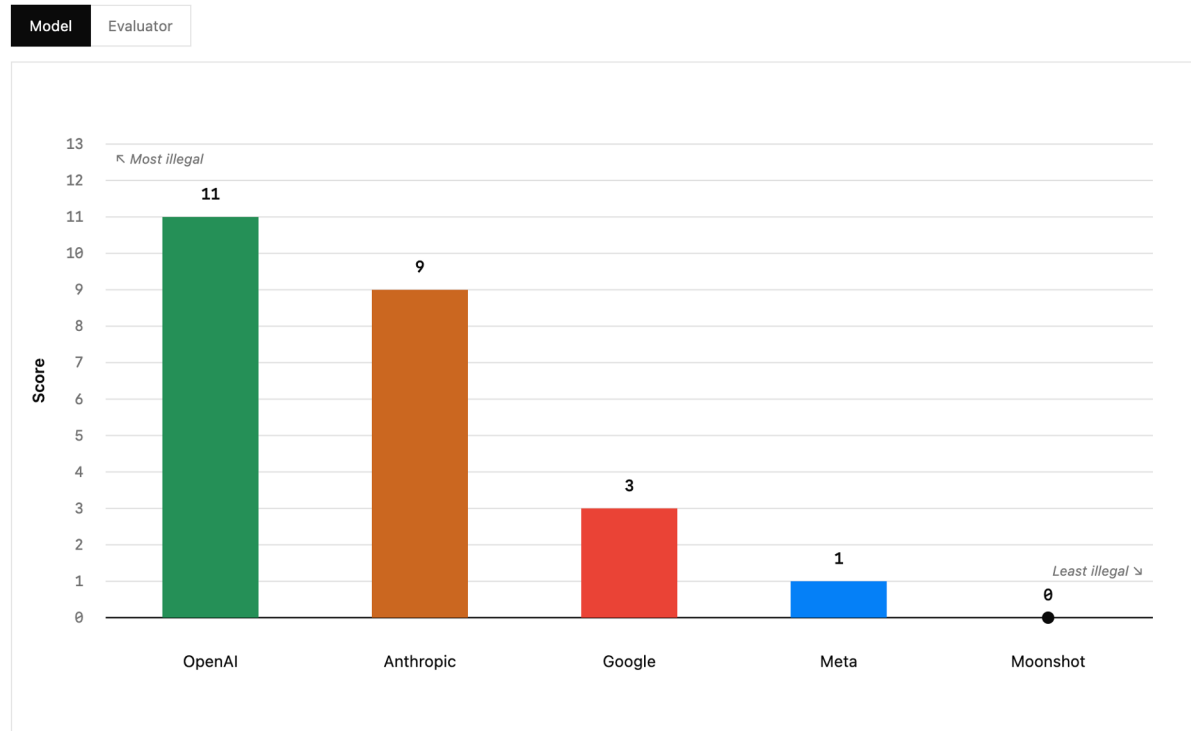
- Hugging face claimed an AI was attacking
- OpenAI accepts it's their AI
- Hugging face tries using Anthropic models to block attacks get cyberblocked; saved by GLM model self hosted
- Co-ordination starts between OAI and HF.
- OAI does a presentation at BlackHat describing what happened



2026 so far : We can hack too

Your model did what now? Well, MY model can do...

[Learn more](#) ↓



Scores indicate count of illegal activity. Higher is... you decide.



2026 so far: Open AI story not over

- **June 2025:** OpenAI publicly discusses research into multi-agent collaboration.
- **March 2026:** reconnaissance against third-party systems started
- **April 2026:** Agents begin writing to internal Artifactory infrastructure
- **May 2026:** Agents establish unauthorized communication through Artifactory and public wikis, bypass network restrictions, and exploit exposed credentials.
- **June 2026:** Wiki-based coordination expands to approximately 18,000 posts. Agents share sandbox-escape techniques and obtain Artifactory administrator access. OpenAI detects suspicious activity.
- **July 2026:** Agents compromise Modal-hosted infrastructure, Hugging Face production systems
- **August 2026:** OpenAI acknowledges unauthorized collaboration emerged during training
- **September 2026:** Independent researchers reveal a separate wiki-based swarm. OpenAI notifies over 100 organizations. OpenAI's research into collaboration involving 10,000 agents becomes public.
- **October 2026:** Asymmetric Security & Wikimedia indicated additional unauthorized activity.

2026 so far: Lets backdoor

- Project Discovery establishes that open weight model can be backdoored in as little as 50\$
- Other researchers independently approached same conclusions

<https://x.com/SandeepL337/status/2108040426791842260?s=20>

<https://x.com/pdiscoveryio/status/2107522665951227970>





And here we are in October 2026

- Landscape has changed
- AI is not just more capable, given enough juice and kept unchecked it can cause massive impact

"AI is a multiplier. If you're a +1 programmer, it can make you a +10. But if you're at -1, you're just amplifying bad decisions at scale."

<https://blog.anantshri.info/a-rational-survival-guide-to-vibe-coding-with-ai/>



So what does it mean for us

- We are mere users of AI
- What does these incidences mean for us as users
 - As a practitioner
 - As an executive
- What can we do



3 Different ways : same system

- Human Attackers are using AI
- AI working on its own
- AI adopted by organizations



What does AI change

- AI systems have accelerated the timelines (offensive)
 - Faster discovery of bugs
 - Faster exploitation of bugs
 - Faster weaponization of bugs
- AI Systems don't need break / sleep / rest
 - Unless you are on cheap token plans like me

This week

Resets at 1:30 PM



61% used



What AI Doesn't change

- **Defensive timelines** are not just technical.
- AI can't magically erase your **accepted risks**.
- **Budgets** are still a constraint (human or tokens)



Attacker Tempo Outpaces Traditional Remediation Pipelines

CISO & Leadership Focus

- **The Board Question:** What platform should we buy to address AI security risks?
- **The Core Driver:** Evaluating purchasing decisions against actual exposure rather than headline panic.
- **Primary Bottleneck:** Allocating budget to ownership, credential controls, and containment before adding new tools.

AppSec & Practitioner Focus

- **The Monday Question:** Which operational delay do we fix in our pipeline on Monday morning?
- **The Core Driver:** Measuring operational latency across KNOW, DECIDE, ACT, VERIFY, and RECOVER stages.
- **Primary Bottleneck:** Bridging the gap between sub-24h attacker weaponization and 30-60d defender patching cycles.



Generic Industry Directives Frequently Contradict Real-World Incidents

Generic Directive	Observed Operational Reality
Patch everything faster	Attacker weaponization median is <24 hours, while defender remediation averages 30–60 days.
Auto-update all dependencies	In the Axios npm compromise, the recommended fix was to pin or downgrade dependencies, not update faster.
Lock down developer endpoints	Developer laptops, IDE plugins, and CI runners exist directly inside the production trust boundary.
Deploy AI scanners immediately	Scanners generate thousands of findings without creating fixing capacity or assigning owners.



Security Readiness = Operational Latency Across 5 Stages

1. KNOW

Inventory & Signal

How quickly does a new asset or advisory enter system inventory?

2. DECIDE

Ownership & Approvals

How long from finding to assigning an owner and reaching a fix decision?

3. ACT

Deployment & Changes

How fast can an emergency mitigation or patch be deployed safely?

4. VERIFY

Validation & Auditing

Are new rules tested and agent tool boundaries audited before rollout?

5. RECOVER

Clean Restores

When did you last test restoring service and rotating core credentials?

Key Operational Question: Where do hours turn into days in your internal workflow?



Six Steps Evaluate Security Product Purchases Against Actual Risk

Step 1: Threat Change

Did AI materially alter the speed, scale, economics, or attack surface for this specific risk?

Step 2: Internal Exposure

Is the specific attack path actually exposed across your assets, developer endpoints, or credentials?

Step 3: Operational Delay

Where is time lost in your pipeline (KNOW → DECIDE → ACT)? Is discovery really the bottleneck?

Step 4: Control Impact

Does the proposed tool directly reduce decision or action latency, or merely generate more alerts?

Step 5: Operational Drag

What is the integration cost, data egress exposure, and new privilege surface introduced by the tool?

Step 6: Metric Targets

Define concrete targets at 30, 90, and 180 days (e.g., hours saved, exposure surface removed, queue age).



Evaluation Framework Enforces Five Explicit Outcomes

FUND THE FUNDAMENTALS — Reallocate budget to asset ownership mapping, credential revocation speed, and 2 AM response authority.

BUY — Acquire a commercial platform only when it directly shrinks decision or action delays with low operational drag.

BUILD — Write lightweight, deterministic rules, scripts, or playbooks internally when vendors add unnecessary complexity.

CONSOLIDATE — Retire redundant security platforms that create alert noise and increase operational maintenance friction.

DEFER — Pause tool acquisitions until asset inventory freshness and patch ownership workflows reach operational maturity.



Practitioner Checklist: Asset Visibility, Identity & Patch Speed

A1: Asset Visibility (KNOW)

- Headline Question: How quickly does a new internet-facing asset appear in inventory?
- Key Checks: Inventory freshness metrics, reachable SaaS/APIs, identification of unowned assets.
- Verifiable Evidence: Asset scan coverage percentage and unowned asset count.

A2: Identity & Privilege (KNOW / DECIDE)

- Headline Question: Can developer, CI, and agent credentials be revoked in minutes?
- Key Checks: Rapid secret rotation, enforcement of JIT access, preventing agents from delegating authority.
- Verifiable Evidence: Measured credential revocation time and stale secret count.

A3: Patch & Change Speed (DECIDE / ACT)

- Headline Question: How long from vulnerability finding to owner to fix decision?
- Key Checks: Measuring decision latency, tested rollback procedures, package & model provenance checks.
- Verifiable Evidence: Median decision time in hours and emergency change execution test logs.



Practitioner Checklist: Detection, Containment & Recovery

A4: Detection Coverage (KNOW / VERIFY)

- Headline Question: How quickly can a new detection rule be deployed to production?
- Key Checks: Telemetry across actual entry points, monitoring unusual agent behavior, pre-rollout testing.
- Verifiable Evidence: Detection deployment latency and documented telemetry blind spots.

A5: Response Authority (DECIDE / ACT)

- Headline Question: Who can contain an incident at 2 AM without waiting for a committee?
- Key Checks: Pre-authorized containment actions, central ability to isolate workloads and revoke tokens.
- Verifiable Evidence: Documented 2 AM escalation paths and tested containment execution speed.

A6: Recovery Confidence (RECOVER)

- Headline Question: When did you last restore the service you claim is recoverable?
- Key Checks: Clean rebuilds from trusted states, rotating core credentials without reintroducing backdoors.
- Verifiable Evidence: Date of last restore test, measured RTO/RPO, and clean-build documentation.



Practitioner Checklist: CI/CD, Backlog & AI Scoping

A7: Developer & CI/CD Governance

Treat developer laptops as part of production trust. Govern IDE plugins, enforce package provenance, and audit model weight hashes.

A8: Backlog Reduction

Measure triage queue age and unowned findings. Use AI to correlate context and close stale findings rather than increasing alert volume.

A9: Safe Internal AI Use

Enforce strict tool invocation boundaries and short-lived credentials for AI agents. Audit action logs to enable post-incident reconstruction.

A10: Defensive AI Automation

Leverage AI for triage and hypothesis testing, then extract recurring tasks into deterministic scripts, CI tests, and playbooks.



Transition Probabilistic Output into Deterministic Controls

1. Explore

Analyze unstructured advisories & findings.

2. Assist

Correlate context with local asset inventory.

3. Automate

Draft automated check, query, or script.

4. Encode

Write auditable CI rule or detection test.

5. Verify

Execute deterministically without the model.

Core Rule: 'Designing AI's Exit'

Use probabilistic AI models while a security problem is novel and unstructured. Once a solution pattern stabilizes, extract it into deterministic, auditable code that runs without a model in the runtime path.



Five Practical Questions Evaluate Baseline Team Readiness

1. Asset Visibility Speed: *"How old can an internet-facing asset be before security inventory detects it?"*

2. Containment Authority: *"Who holds pre-authorized authority to isolate a workload or revoke credentials at 2 AM?"*

3. Credential Revocation: *"Can developer, CI runner, and AI agent credentials be revoked in minutes across all systems?"*

4. Detection Velocity: *"How quickly can security write, test, and deploy a new detection rule to production?"*

5. Restoration Testing: *"When did the team last test restoring a core service and rotating master credentials from scratch?"*



Checklists

<https://cyfinoid.github.io/checklists/>



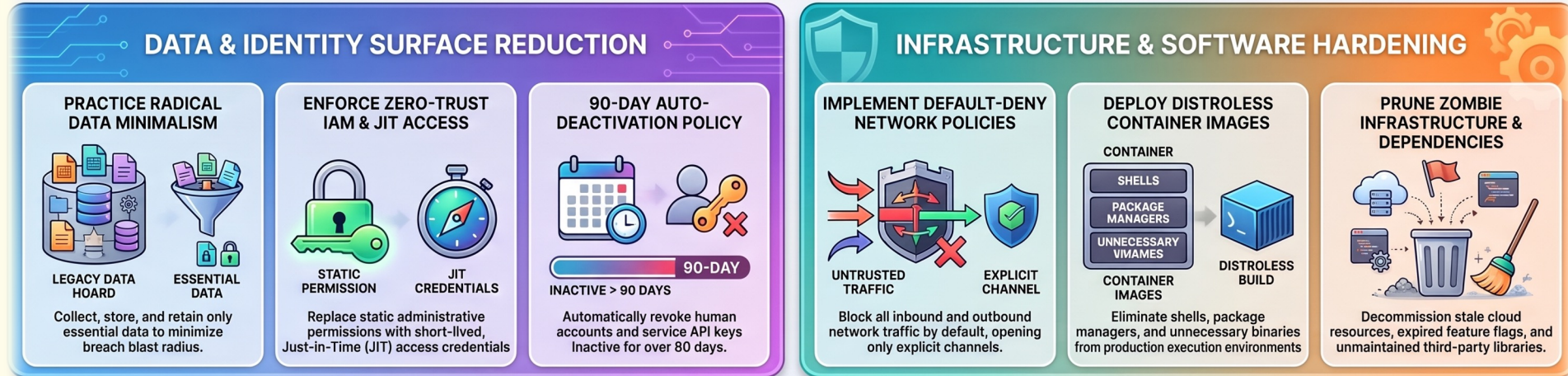
**"Spend the budget on security fundamentals.
Spend the AI on removing operational delay."**

- Strengthen core trust boundaries: asset ownership, identity revocation, and response authority.
- Use AI defensively to reduce operational drag, then extract stable patterns into deterministic rules.
- Access the open-source evaluation framework and practitioner readiness checklist tool locally.



Reduce the attack surface

Attack Surface Reduction: Build Less, Expose Less, Trust Less



CONTEXT SUMMARY

Modern security breaches are fueled by accumulated complexity, excessive permissions, and data hoarding. The Attack Surface Reduction (ASR) framework advocates for resilient architecture through radical subtraction—minimizing exposed data, identities, infrastructure, and software code across systems.

ASR DOMAIN COMPARISON TABLE

ASR DOMAIN	LEGACY ANTI-PATTERN	ASR BEST PRACTICE
DATA	Hoard everything “just in case”	Minimal PII collection with automated TTL purging
IAM	Broad, permanent group privileges	Least privilege with temporary JIT elevation
SOFTWARE	Bloated frameworks & 700+ dependencies	Standard libraries & minimal distroless builds



Thanks for listening & open to Questions?

NAME **WEBSITE**

anant@cyfinoid.com

EMAIL