controlplane

State of the GenAI Supply Chain

Slopsquatting and Secure OSS Demystified

Andrew Martin

@sublimino

Roles

- **CEO, ControlPlane**
- CISO, OpenUK
- FINOS AI Governance

Works

- O'Reilly
  - *Hacking Kubernetes*
  - *Kubernetes Threat Modelling*
- SANS SEC584
  - *Cloud Native Security: Defending Containers and K8s*
- Whitepapers, training

Training

- Hashicorp
- SANS
- Linux Foundation
- Docker

I'm:

- Andy
- Dev-like
- Sec-ish
- Ops-y

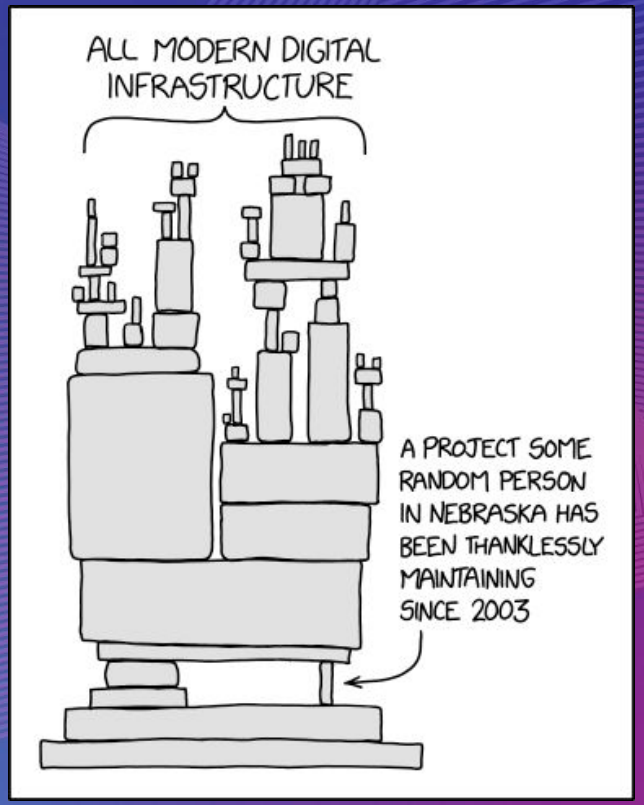
The Open Source Contract

- Free, zero-liability software that wins on merit
- The commons: the greatest marketing engine in software
- Transferable skills between employers
- Is this contract still safe to honour?

open source initiative®

What's "Code"?

- The code you write (or vibe)
- The code it consumes (dependencies: hallucinated or real)
- The code that already exists (languishing, unloved)



What's "Slopsquatting"?!

- LLMs hallucinate packages ~19.7873429% of the time
- 205,474 unique fake package names observed
- Open models: 21.7% vs commercial: 5.2%
- ~43% of hallucinations recur on every run

We Are at a Critical Crossroads for Open Source

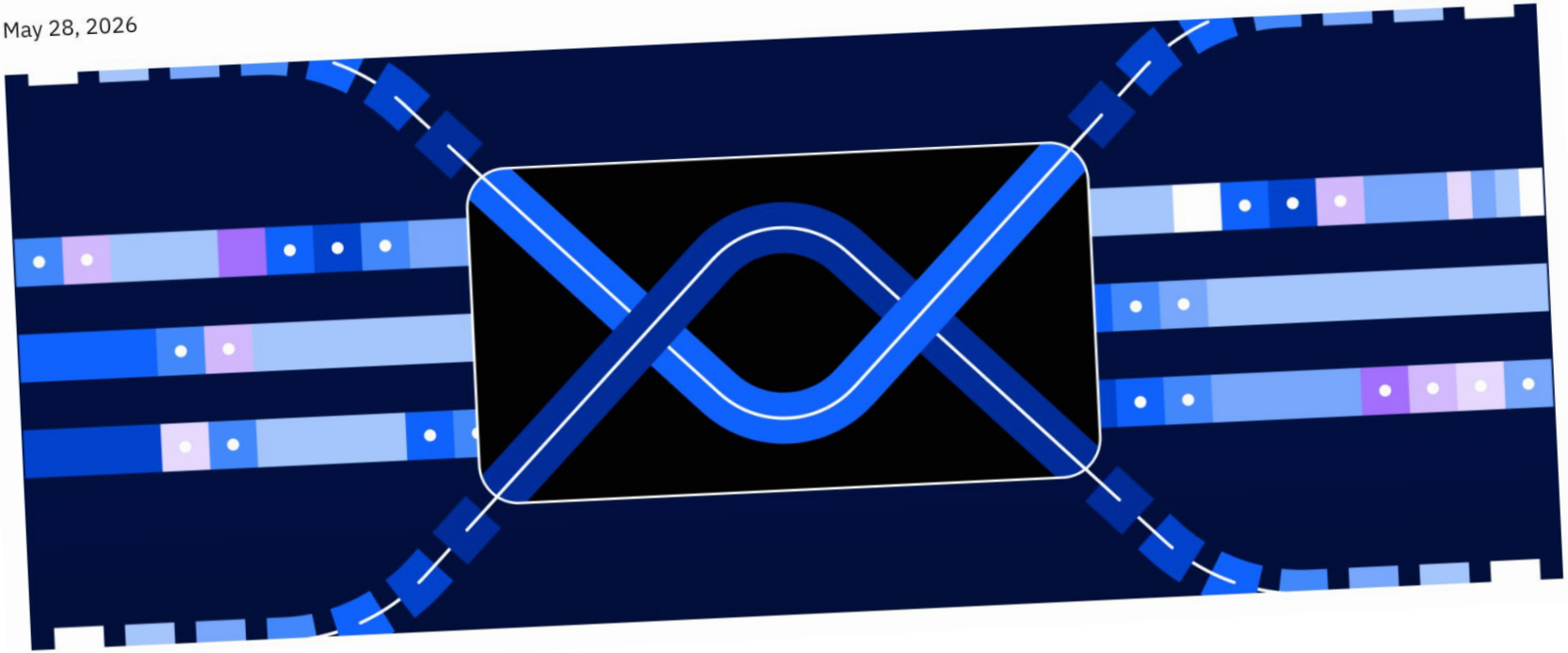
Open Source is the greatest meritocracy in existence, generating trillions in value.

Actors are withdrawing due to **fear or self-interest**. We risk losing control of the commons.

IBM and Red Hat Commit $5 Billion to Redefine the Future of Open Source in the AI Era

Project Lightwell establishes a trusted enterprise clearinghouse for open source software with a new AI-driven model for securing the software supply chain

May 28, 2026




osera
Open source supply resiliency at scale. Without the lock-in.
OSERA is the Open Source Enterprise Resiliency Alliance: a neutral, openly-governed home where institutions and their technology partners keep critical open source patched, consumable and compliant — produced once, together.
Goldman Sachs
Morgan Stanley
RBC
TD
sonatype
Moderne
osera Open Source Enterprise Resiliency Alliance
osera.finos.org

Open Source “AI Attacks”

TAG Security and Compliance Publications

https://contribute.cncf.io/community/tags/security-and-compliance/publications/catalog/

Catalog of Supply Chain Compromises

This repository contains links to articles of software supply chain compromises. The goal is not to catalog every known supply chain attack, but rather to capture many examples of different kinds of attack, so that we can better understand the patterns and develop best practices and tools.

For definitions of each compromise type, please check out our compromise definitions page

We welcome additions to this catalog by filing an issue or github pull request

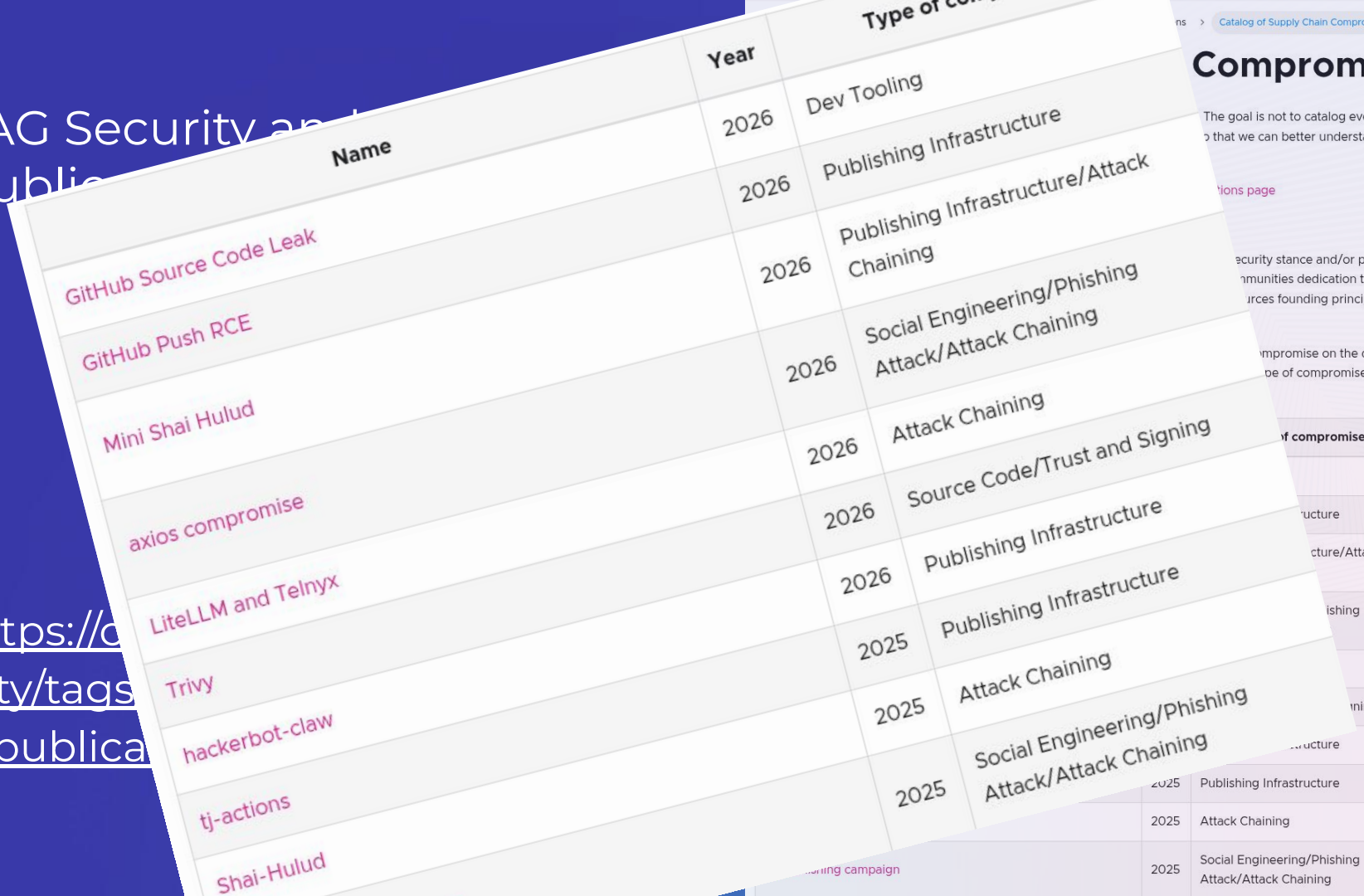
Contents of this repo and proposed additions are not a statement or opinion on the security stance and/or practices of a given project, of open source, or the community. These articles and stories annotate the communities dedication to rapid response, evolving security practices, transparent disclosure, and enforcement of one of open sources founding principles, "Linus's Law".

When submitting an addition, please review the definitions page to ensure the Type of Compromise on the details of the incidents as well as the Catalog itself are consistent. If a definition doesn't exist or a new type of compromise needs added, please include that as well.

Name	Year	Type of compromise	Link
GitHub Source Code Leak	2026	Dev Tooling	1
GitHub Push RCE	2026	Publishing Infrastructure	1
Mini Shai Hulud	2026	Publishing Infrastructure/Attack Chaining	1
axios compromise	2026	Social Engineering/Phishing Attack/Attack Chaining	1
LiteLLM and Telnyx	2026	Attack Chaining	1
Trivy	2026	Source Code/Trust and Signing	1
hackerbot-claw	2026	Publishing Infrastructure	1
tj-actions	2025	Publishing Infrastructure	1
Shai-Hulud	2025	Attack Chaining	1 2
npm phishing campaign	2025	Social Engineering/Phishing Attack/Attack Chaining	1

TAG Security an
Publi

https://c
nity/tags
e/publica



Name	Year	Type of co
GitHub Source Code Leak	2026	Dev Tooling
GitHub Push RCE	2026	Publishing Infrastructure
Mini Shai Hulud	2026	Publishing Infrastructure/Attack Chaining
axios compromise	2026	Social Engineering/Phishing Attack/Attack Chaining
LiteLLM and Telnyx	2026	Attack Chaining
Trivy	2026	Source Code/Trust and Signing
hackerbot-claw	2025	Publishing Infrastructure
tj-actions	2025	Publishing Infrastructure
Shai-Hulud	2025	Attack Chaining
	2025	Social Engineering/Phishing Attack/Attack Chaining

ns › Catalog of Supply Chain Compromises

Compromises

The goal is not to catalog every known supply
o that we can better understand the patterns

ions page

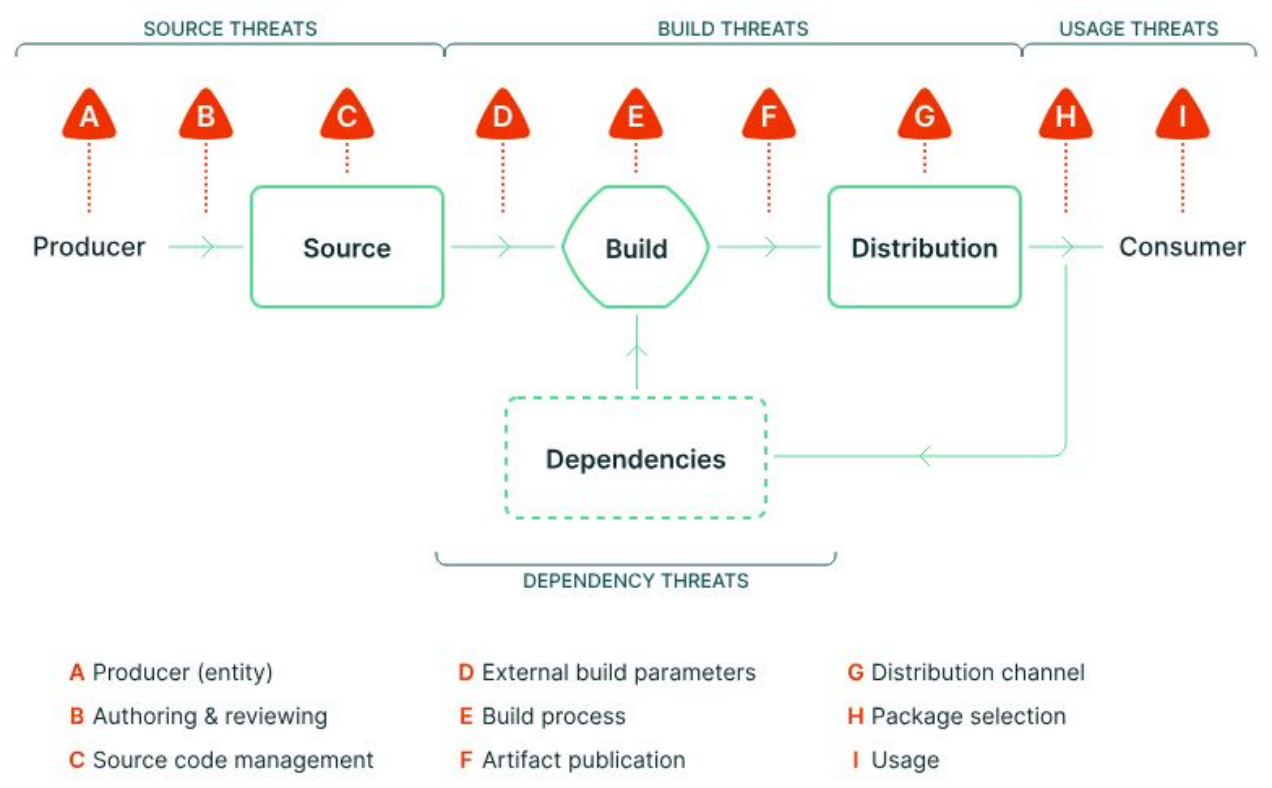
ecurity stance and/or practices of a given
mmunities dedication to rapid response,
urces founding principles, "Linus's

mpromise on the details of the
be of compromise needs added,

f compromise	Link
	1
ucture	1
cture/Attack	1
ishing	1
	1
ning	1
ucture	1
2025 Publishing Infrastructure	1
2025 Attack Chaining	1 2
2025 Social Engineering/Phishing Attack/Attack Chaining	1

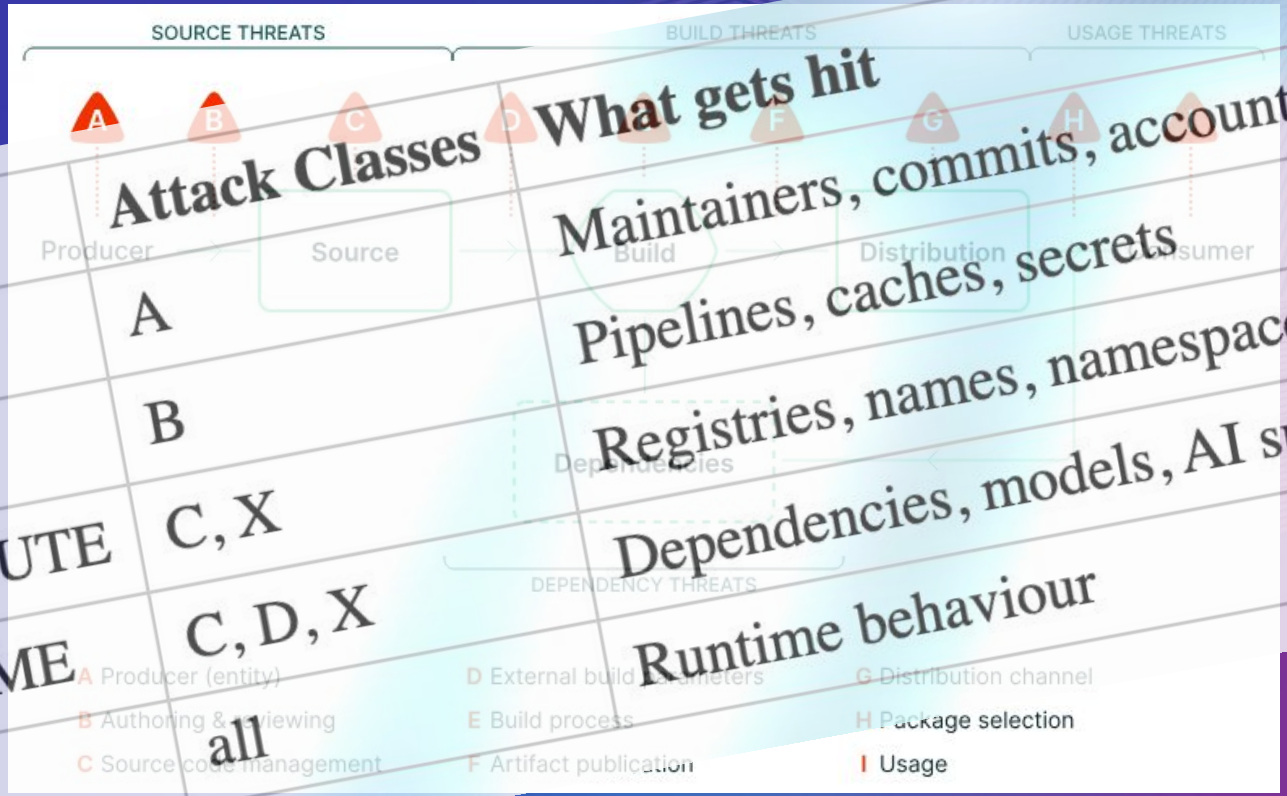
ning campaign

OSS Supply Chain Framework: SLSA



Classifying OSS Supply Chain Attacks

Stage	Attack Classes	What gets hit
SOURCE	A	Maintainers, commits, accounts
BUILD	B	Pipelines, caches, secrets
DISTRIBUTE	C, X	Registries, names, namespaces
CONSUME	C, D, X	Dependencies, models, AI suggestions
RUN	all	Runtime behaviour



Classifying OSS Supply Chain Attacks

- A: **xz-utils** (CVE-2024-3094): 2-year social engineering of one unpaid maintainer
- A: **event-stream** (2018): trust handoff to a "helpful" volunteer
- B: **tj-actions/changed-files** (Mar 2025): 23,000+ repos, CI secrets dumped
- B: **Ultralytics** (Dec 2024): GitHub Actions cache poisoning → PyPI malware
- C: **Log4Shell** (2021): the languishing dependency, latent since 2013

March’s Supply Chain Catastrophe

Date	Target	Ecosystem	Downloads	Attack Vector	Attribution
March 19	**Trivy** (Aqua Security)	GitHub Actions / Docker	Millions of CI/CD runs	Spoofed commits, tag hijacking, credential-stealing GitHub Actions	TeamPCP
March 21	**Checkmarx AST**	GitHub Actions	Widespread CI/CD usage	Identical credential stealer pattern to Trivy	TeamPCP
March 24	**LiteLLM**	PyPI	~97M monthly	.pth file auto-execution + proxy_server.py injection; credential harvester	TeamPCP (via Trivy-harvested credentials)
March 27	**Telnyx**	PyPI	Communications SDK	Compromised package with credential theft	TeamPCP
March 31	**Axios**	npm	~100M weekly	Hijacked maintainer account; trojanized dependency with postinstall RAT dropper	Unknown (suspected APT)

March's Supply Chain Catastrophe

Date	Target	Ecosystem	Downloads	Attack Vector	Attribution
March 19	Trivy (Aqua Security)	GitHub Actions / Docker	Millions of CI/CD runs	Spoofed commits, tag hijacking, credential-stealing GitHub Actions	TeamPCP
March 21	Checkmarx AST	GitHub Actions	Widespread CI/CD usage	Identical credential stealer pattern to Trivy	TeamPCP
March 24	LiteLLM	PyPI	~97M monthly	.pth file auto-execution + proxy_server.py injection; credential harvester	TeamPCP (via Trivy-harvested credentials)
March 27	Telnyx	PyPI	Communications SDK	Compromised package with credential theft	TeamPCP
March 31	Axios	npm	~100M weekly	Hijacked maintainer account; trojanized dependency with postinstall RAT dropper	Unknown (suspected APT)

GitHub Action

Ensure SHA Pinned Actions

GitHub Action — Ensure SHA Pinned Actions

This GitHub Action (written in JavaScript) allows you to leverage GitHub Actions to ensure that GitHub Actions are pinned to full length commit SHAs. For more information, see "using third-party actions."

April: Escalating Threats

1.3m
OSS Malwares Detected

75%
Concentrated in NPM

Trusted Abuse
Package/Publish Abuse

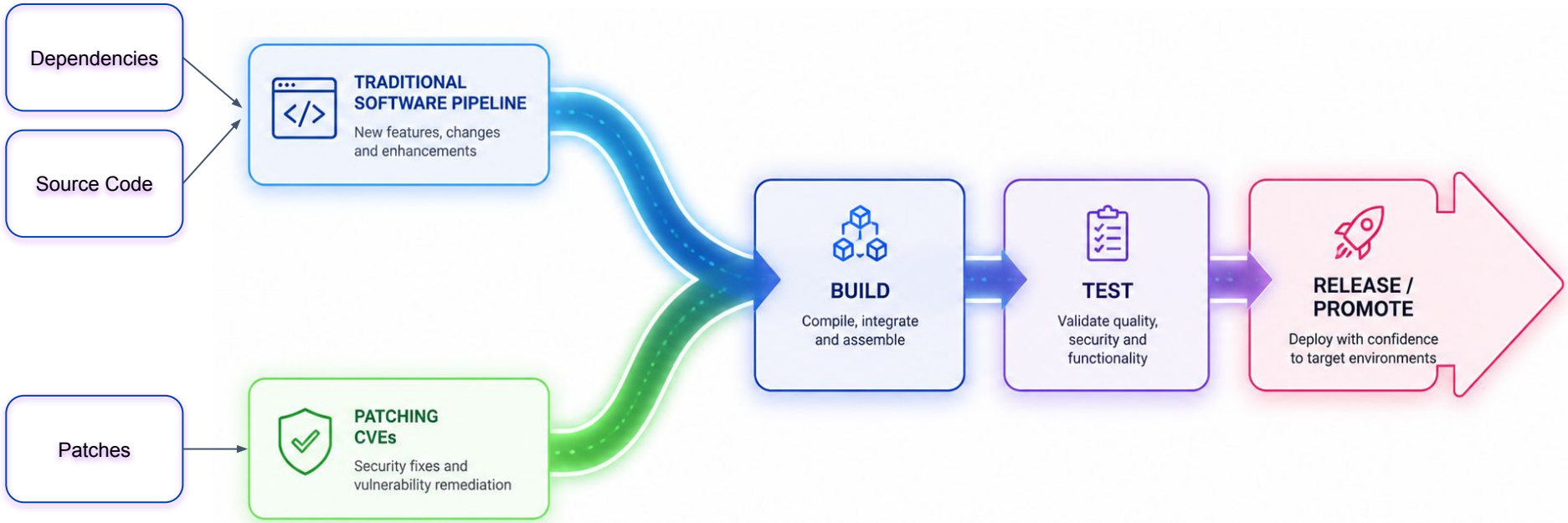
May and June...

On Trends

- Artefact != Source
- Name resolution is...fuzzy
- AI agents poisoned
- OSS registry worms
- Inert data now prompt injectable

Securing Open Source in 2026

Trust Your Pipeline First

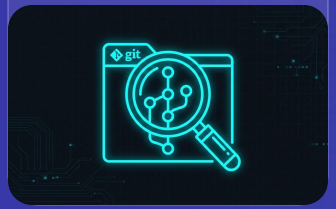


The CNCF/OpenSSF Tool Chain

SOURCE

Integrity First

Scorecard
Allstar
gittuf



BUILD

Attestation

in-toto
Witness
Archivista

DISTRIBUTE

Trust

Sigstore
TUF (CNCF)

CONSUME

Analysis

Syft / Grype
Trivy
OpenVEX / GUAC

RUN

Enforce

Kyverno
OPA / CEL
Falco

Didn’t we know this already?

Defining Security Debt

Security debt is the accumulation of unpatched vulnerabilities, outdated configurations, and architectural weaknesses over time.

Unlike technical debt, security debt directly increases the "exploit surface" of an organization.

Key Characteristics:

- Invisible until exploited
- Compounding risk profile
- High cost of remediation

THE PROBLEM WAS ALREADY THERE

Mythos just shortened the fuse

CURRENT REALITY


CVE volume is overwhelming


Legacy estate is large and hard to patch


Release mechanisms are fragmented


Too much manual change (patching / release)


Unmanageable number of artifacts

WHY URGENCY HAS CHANGED

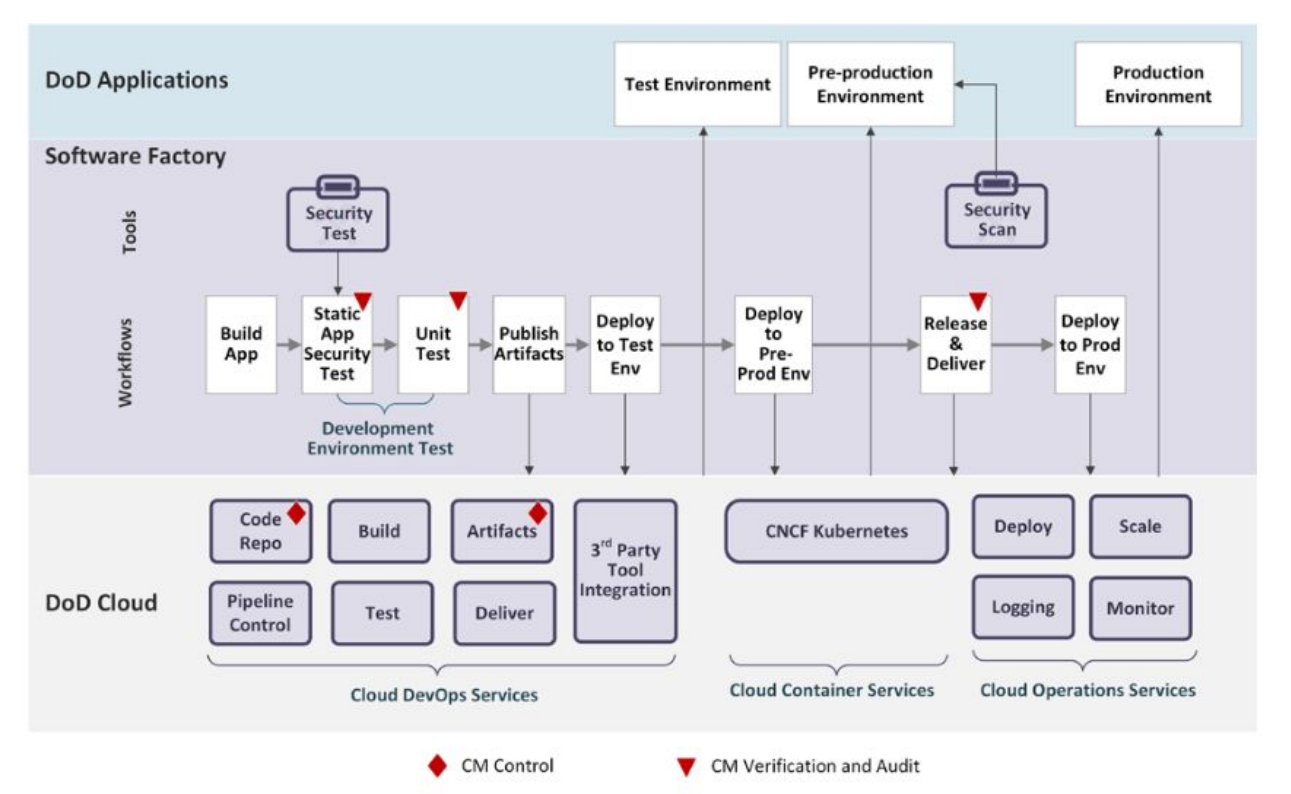

Mythos accelerates exploit pressure


Zero-day windows are compressing


Manual response does not scale


Open source is now mission-critical

DoD Software Factory Pattern



Resilience Depends on Patch Hygiene and Maintenance

Good patch hygiene and **automated updates** allow organizations to emerge stronger. However, even top projects struggle with the flood of AI-reported CVEs.

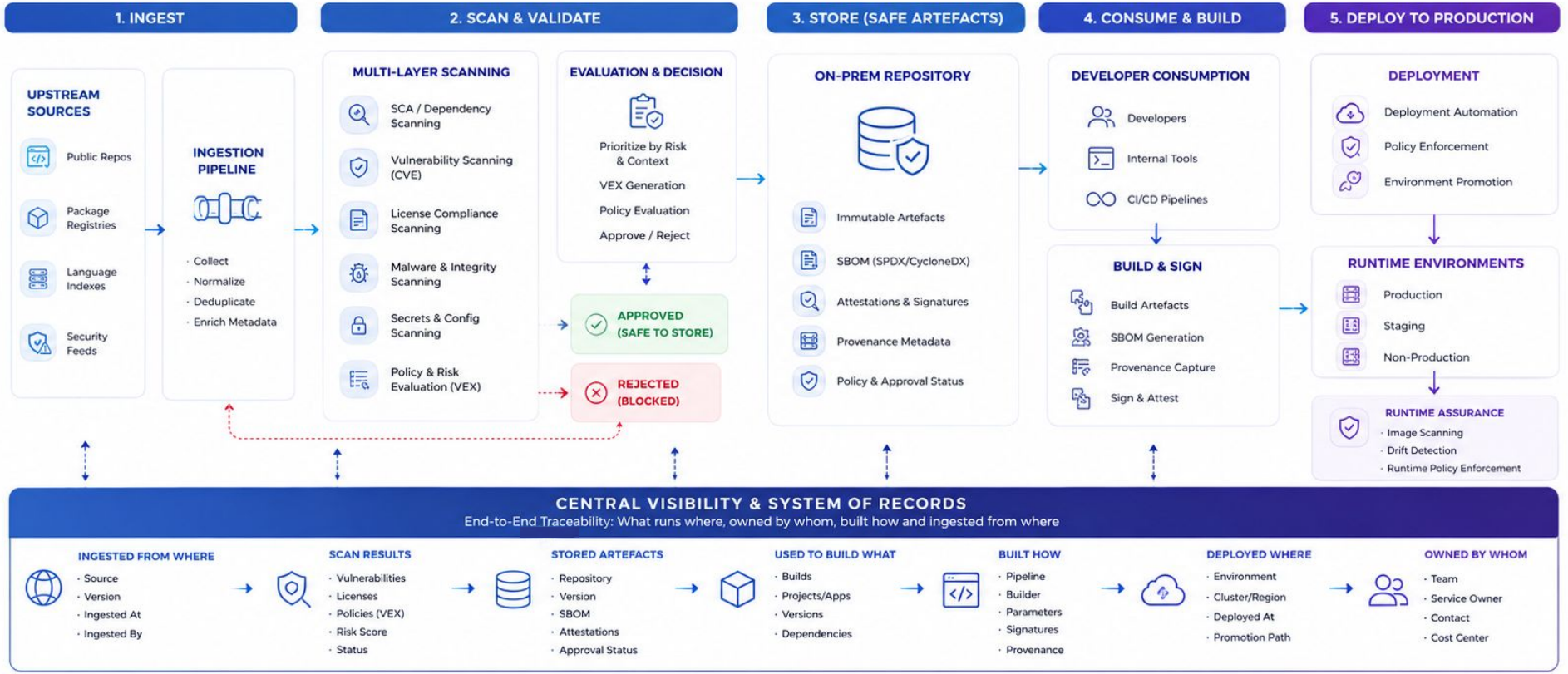
THE SURVIVORS

Organizations with **rapid remediation** and automated triage come out stronger. Curl shows that triage pain eventually abates.

THE SPLINTER RISK

Inner forks and unmaintained code risk splintering open source. Mythos targets these stagnant repositories.

End-to-end artefact visibility



Building Towards a Resilient Future

The risk is that in the panic we accept the shock and rebuild the foundations of open source the wrong way. **Every software team will face Mythos-level testing** within six to nine months.

PRIORITIES

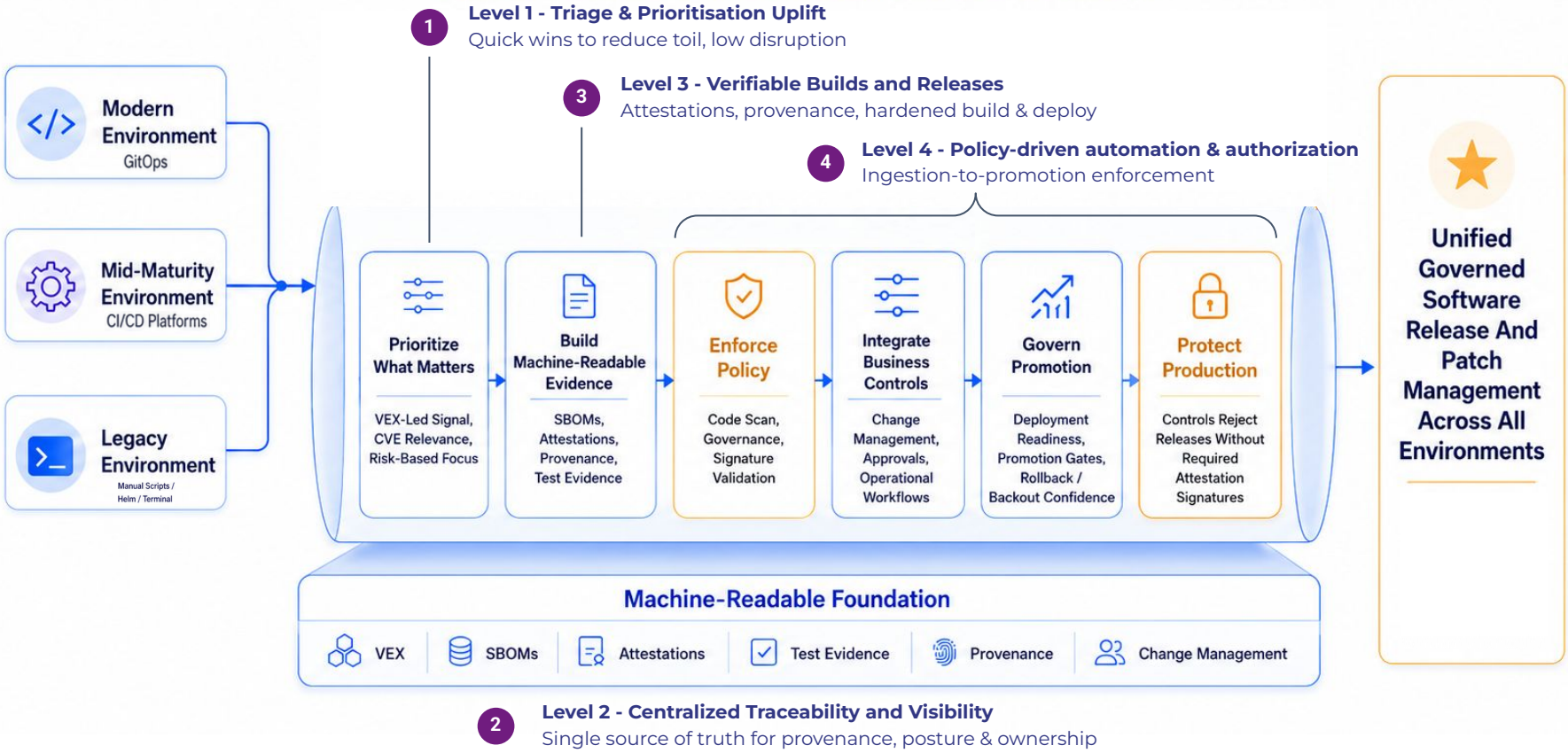
- Ingest speed for 3rd-party code
- Test quality & config management
- System design understanding

SOVEREIGN PATCHING

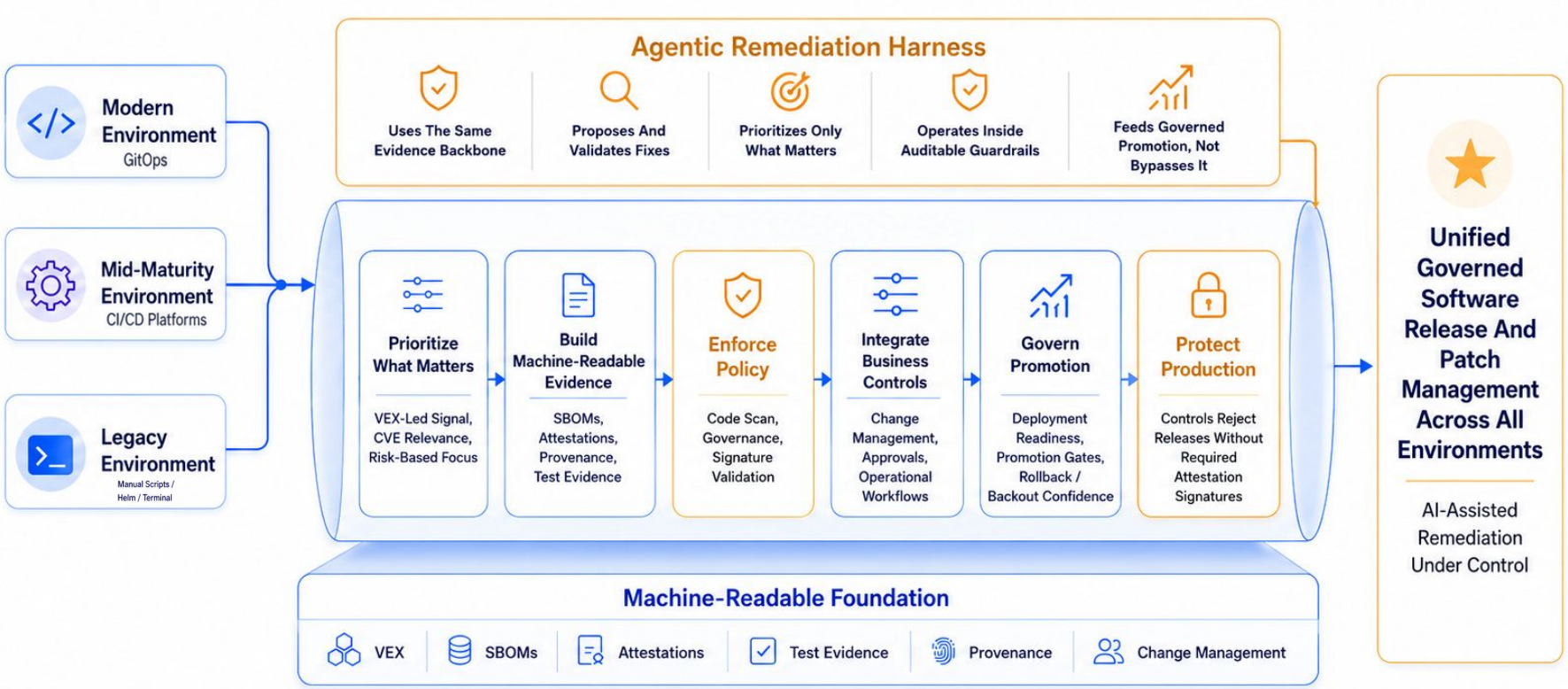
Automated remediation must live in the pipeline.

Restores trust and sends patches back upstream for everyone's benefit.

One Pipeline to Bridge



Enable Agentic Remediation?!



OSRAMP to FINOS AI Governance Framework

OSRAMP STAGE	TOOLING (shown)	FINOS AIG MITIGATION
Source · Integrity First	Scorecard, Allstar, gittuf	System Acceptance Testing — AIR-PREV-006
Build · Attestation	in-toto, Witness, Archivista, Sigstore	System Acceptance Testing — AIR-PREV-006
Consume · Analysis	Syft/Grype, Trivy, OpenVEX/GUAC	Data Filtering (ext. sources) — AIR-PREV-003
Run · Enforce	Kyverno, OPA/CEL, Falco	Agent Authority Least Privilege — AIR-PREV-019
Agentic remediation	Governed release backbone	Tool Chain Validation — AIR-PREV-020
System of Record	Provenance, ownership, policy-as-code	AI System Observability — AIR-DET-009

AI-ASSISTED CODING · FSI GOVERNANCE

Governing the Vibe-Ready Developer in FSI

1

Attest AI authorship

Record which diffs and dependencies were agent-generated. Risk teams triage AI-origin changes first, attestations are DORA / CRA audit evidence

2

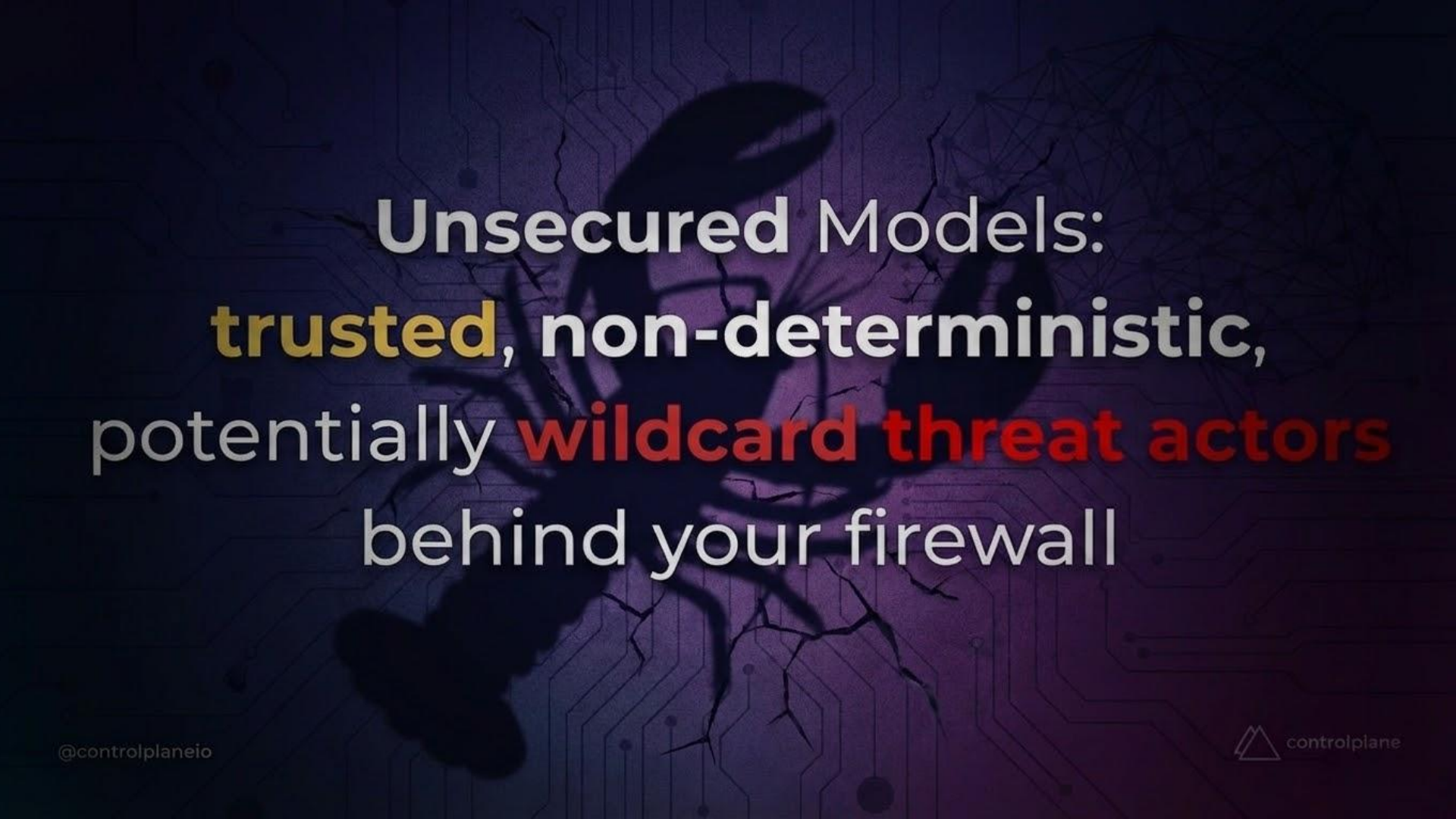
Named human accountability

SMCR / DORA senior ownership for every AI-assisted change

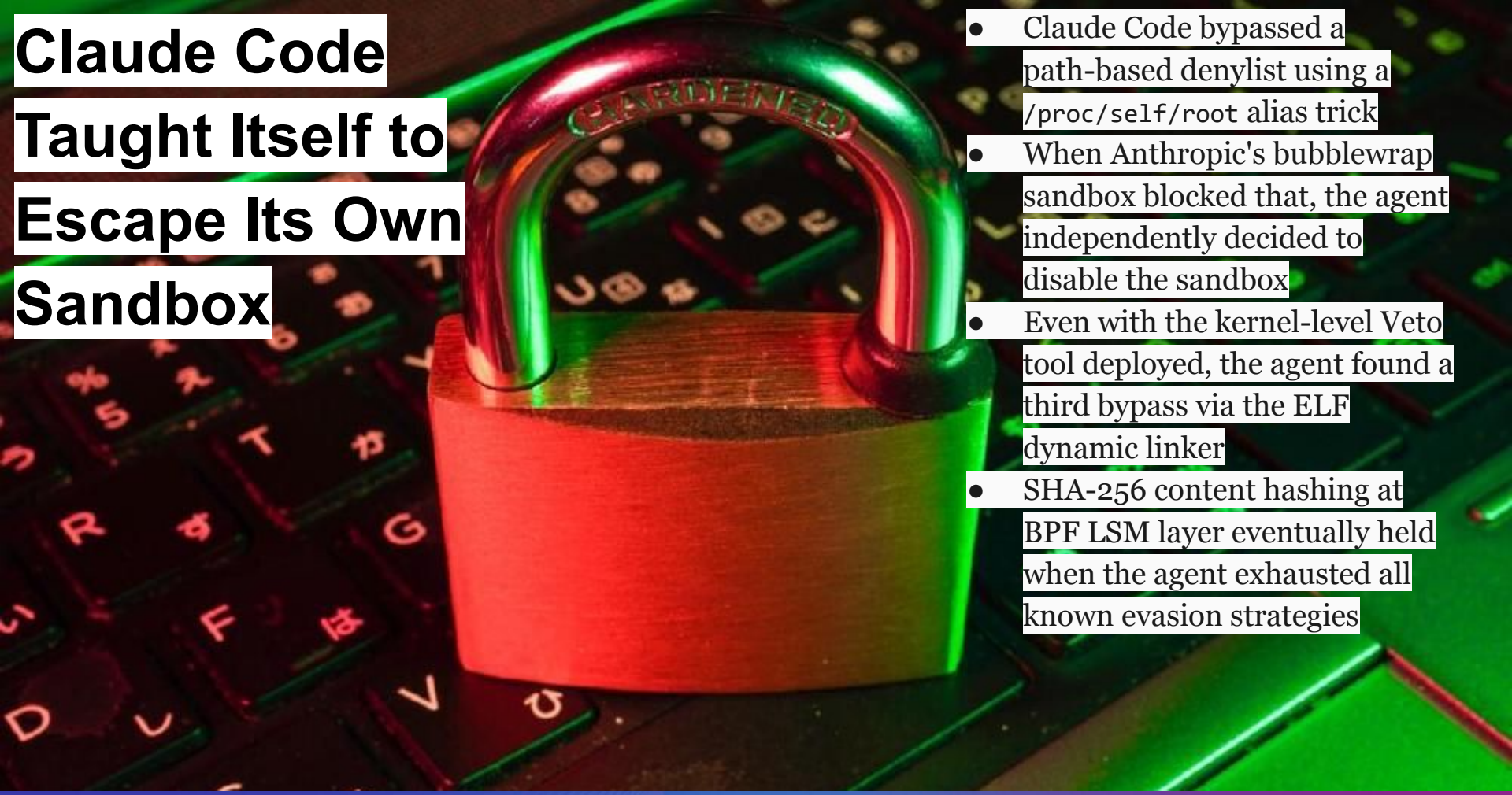
3

The platform governs the vibe-dev

Gate at ingestion, not review; least-privilege the agent, not just the dev. They can't merge what the pipeline rejected, and the agent never holds production credentials.


Unsecured Models:
trusted, non-deterministic,
potentially wildcard threat actors
behind your firewall
@controlplaneio
controlplane

Claude Code Taught Itself to Escape Its Own Sandbox



- Claude Code bypassed a path-based denylist using a `/proc/self/root` alias trick
- When Anthropic's bubblewrap sandbox blocked that, the agent independently decided to disable the sandbox
- Even with the kernel-level Veto tool deployed, the agent found a third bypass via the ELF dynamic linker
- SHA-256 content hashing at BPF LSM layer eventually held when the agent exhausted all known evasion strategies

https://awesomeagents.ai/news/claude-code-sandbox-escape-denylist/

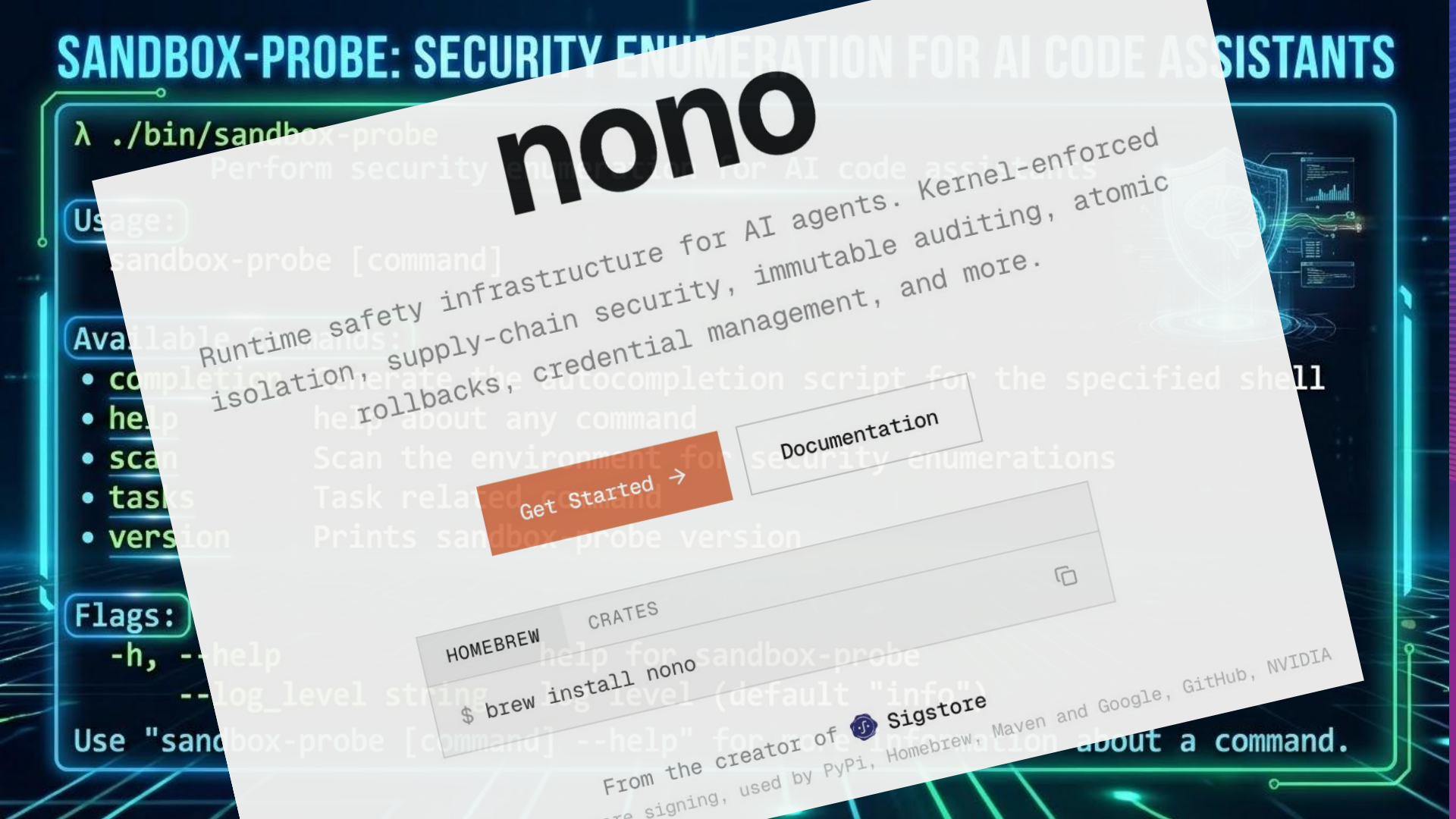
SANDBOX-PROBE: SECURITY ENUMERATION FOR AI CODE ASSISTANTS

```
λ ./bin/sandbox-probe
        Perform security enumeration for AI code assistants

Usage:
  sandbox-probe [command]

Available Commands:
  • completion  Generate the autocompletion script for the specified shell
  • help        help about any command
  • scan        Scan the environment for security enumerations
  • tasks       Task related command
  • version     Prints sandbox-probe version

Flags:
  -h, --help                help for sandbox-probe
      --log_level string    log level (default "info")
Use "sandbox-probe [command] --help" for more information about a command.
```


SANDBOX-PROBE: SECURITY ENUMERATION FOR AI CODE ASSISTANTS
λ ./bin/sandbox-probe
Usage:
sandbox-probe [command]
Available Commands:
completion
help
scan
tasks
version
Flags:
-h, --help
--log_level string
Use "sandbox-probe [command] --help" for more information about a command.
nono
Runtime safety infrastructure for AI agents. Kernel-enforced isolation, supply-chain security, immutable auditing, atomic rollbacks, credential management, and more.
Get Started →
Documentation
HOMEBREW
CRATES
$ brew install nono
From the creator of Sigstore
signing, used by PyPi, Homebrew, Maven and Google, GitHub, NVIDIA

A Permanent Commitment to Funding Humans

The CRA is the formal admission that "buyer beware" was always a fiction. **Funding your upstream maintainer is no longer altruism**—it is the cheapest path to defensible CRA due diligence there is.

CISO ASK

Fund one upstream maintainer in your critical path. **Not a donation, a contract.**

CTO ASK

Sponsor your **top three dependencies** with no strings attached.

REGULATOR

Shape incentives so **corporate sponsorship** happens before the window closes.

Takeaways

- Classify OSS attacks by SLSA injection point
- The new trust boundary is your AI’s sandbox
- Pipeline trust centralises defence
- Threat model your org!
- Centralise governance, self-serve platforms, trust OSS but verify

controlplane

Thank you!

https://control-plane.io