Shipping Safer Container Runtimes in 2026

Developer-friendly supply chain security with Chainguard

Chainguard

About Me

- Staff DevRel Engineer at Chainguard
- Joined in 2022
- Linux, Containers, PHP

What we'll talk about today

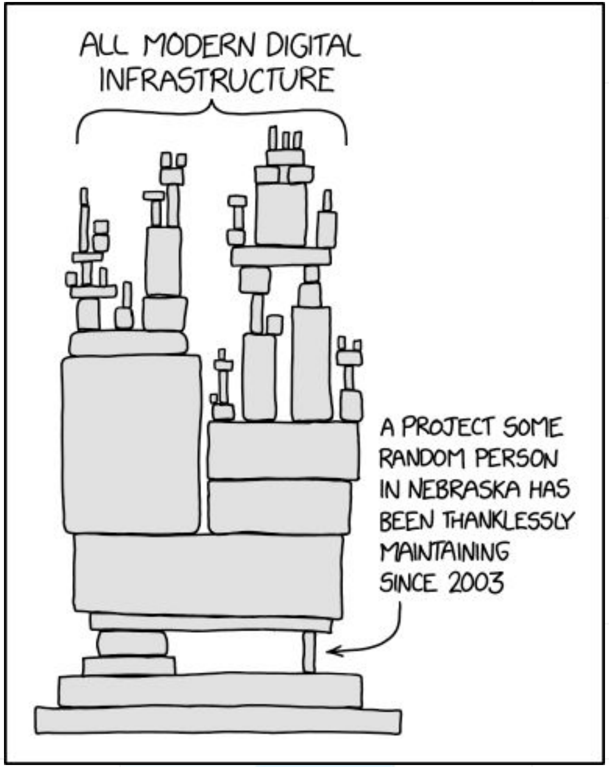
1. Why Devs Need to Care
2. Trending Threat Models
3. Mitigating Risks
4. Concrete Dev Actions **This Sprint**
5. Demo: Migrating to Chainguard Containers
6. Q&A

Why devs need to care

Understanding the threat models devs are exposed to

Why devs need to care

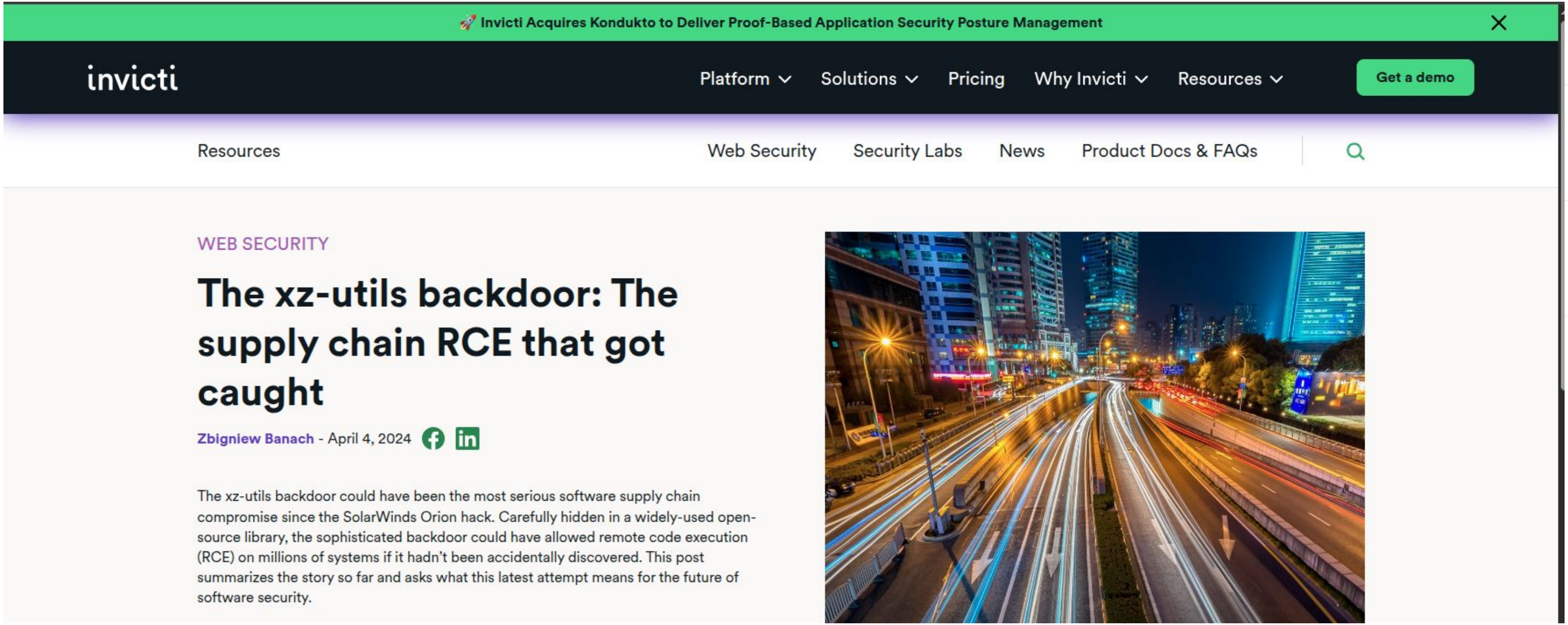
- Supply chain attacks now target package managers and ecosystems directly (npm, Maven, etc.), in addition to OS images
- Devs own Dockerfiles and dependency manifests – they're the ones who can fix it
- Recent Incidents that made the news
  - XZ Utils (2024) [link]
  - tj-actions/changed-files GH Action Compromise (2025) [link]
  - Sha1-Hulud NPM worm (2025) [link]


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Mandatory XKCD comic

XZ Utils / liblzma (2024)

- Malicious tar release introduced by a long-term maintainer (2+ years in project)
- Exploits the SSH service to allow unauthorized access to affected systems (backdoor)
- Compromised build system, obfuscated malicious code only executed with a few conditions
- Source code not visibly affected on repository
- Could have been catastrophic if not detected early

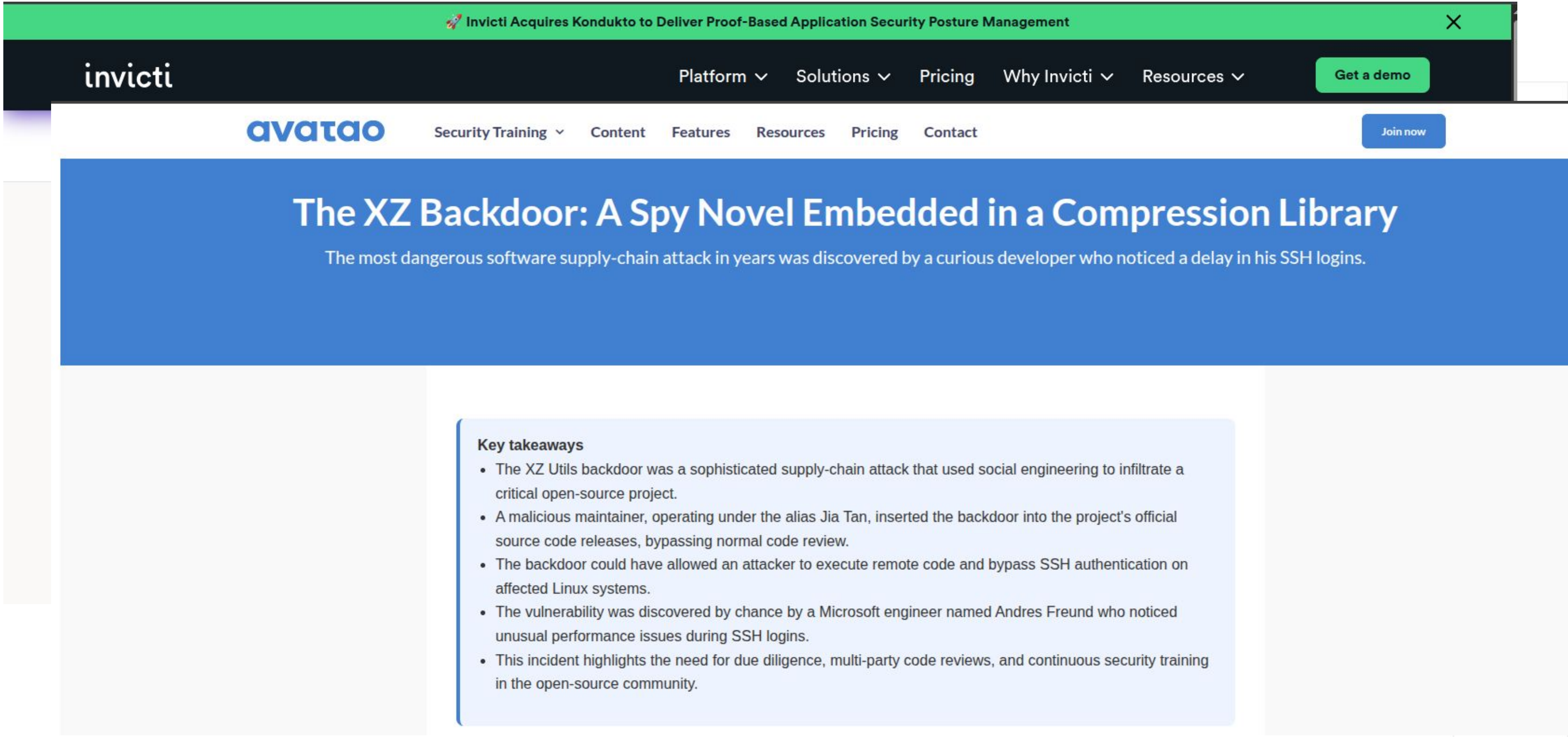


WEB SECURITY

The xz-utils backdoor: The supply chain RCE that got caught

Zbigniew Banach - April 4, 2024

The xz-utils backdoor could have been the most serious software supply chain compromise since the SolarWinds Orion hack. Carefully hidden in a widely-used open-source library, the sophisticated backdoor could have allowed remote code execution (RCE) on millions of systems if it hadn't been accidentally discovered. This post summarizes the story so far and asks what this latest attempt means for the future of software security.

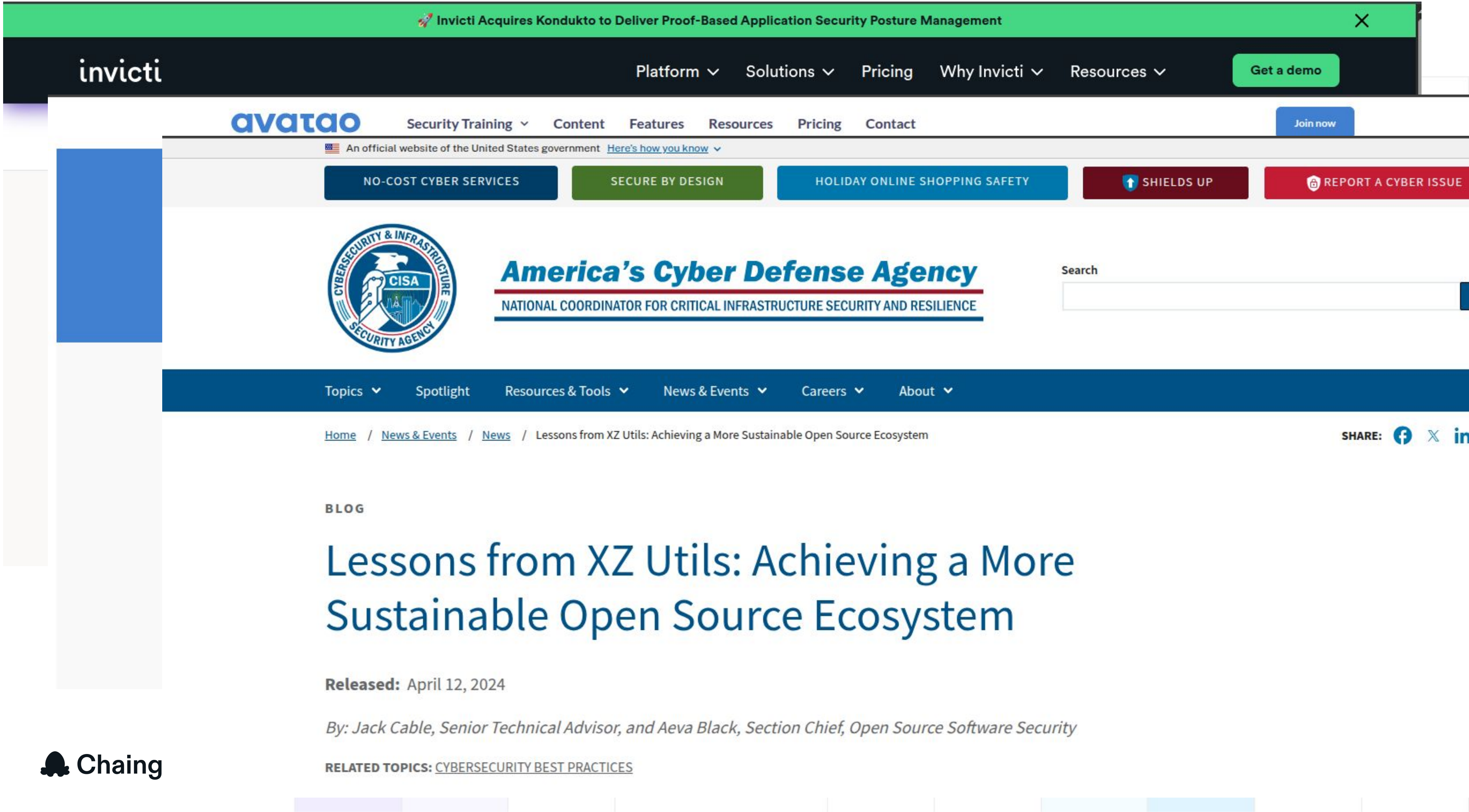


The XZ Backdoor: A Spy Novel Embedded in a Compression Library

The most dangerous software supply-chain attack in years was discovered by a curious developer who noticed a delay in his SSH logins.

Key takeaways

- The XZ Utils backdoor was a sophisticated supply-chain attack that used social engineering to infiltrate a critical open-source project.
- A malicious maintainer, operating under the alias Jia Tan, inserted the backdoor into the project's official source code releases, bypassing normal code review.
- The backdoor could have allowed an attacker to execute remote code and bypass SSH authentication on affected Linux systems.
- The vulnerability was discovered by chance by a Microsoft engineer named Andres Freund who noticed unusual performance issues during SSH logins.
- This incident highlights the need for due diligence, multi-party code reviews, and continuous security training in the open-source community.


🚀 Invicti Acquires Kondukto to Deliver Proof-Based Application Security Posture Management
invicti
Platform
Solutions
Pricing
Why Invicti
Resources
Get a demo
avatao
Security Training
Content
Features
Resources
Pricing
Contact
Join now
An official website of the United States government
Here's how you know
NO-COST CYBER SERVICES
SECURE BY DESIGN
HOLIDAY ONLINE SHOPPING SAFETY
SHIELDS UP
REPORT A CYBER ISSUE
CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY
CISA
America's Cyber Defense Agency
NATIONAL COORDINATOR FOR CRITICAL INFRASTRUCTURE SECURITY AND RESILIENCE
Search
Topics
Spotlight
Resources & Tools
News & Events
Careers
About
SHARE:

Home / News & Events / News / Lessons from XZ Utils: Achieving a More Sustainable Open Source Ecosystem

BLOG

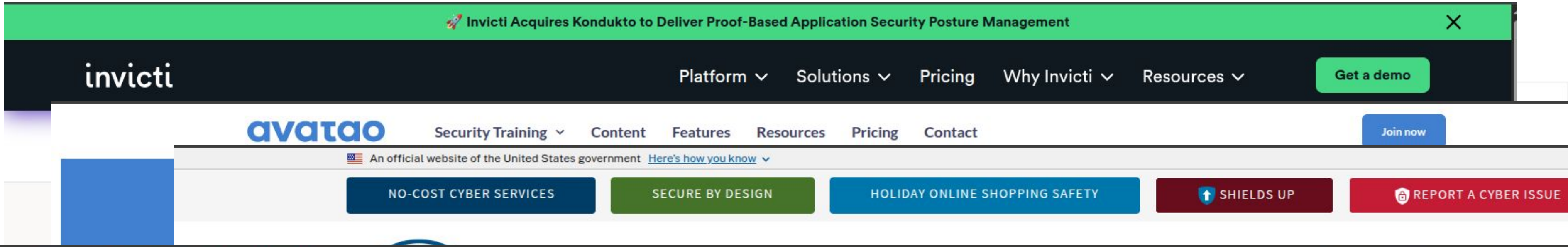
Lessons from XZ Utils: Achieving a More Sustainable Open Source Ecosystem

Released: April 12, 2024

By: Jack Cable, Senior Technical Advisor, and Aeva Black, Section Chief, Open Source Software Security

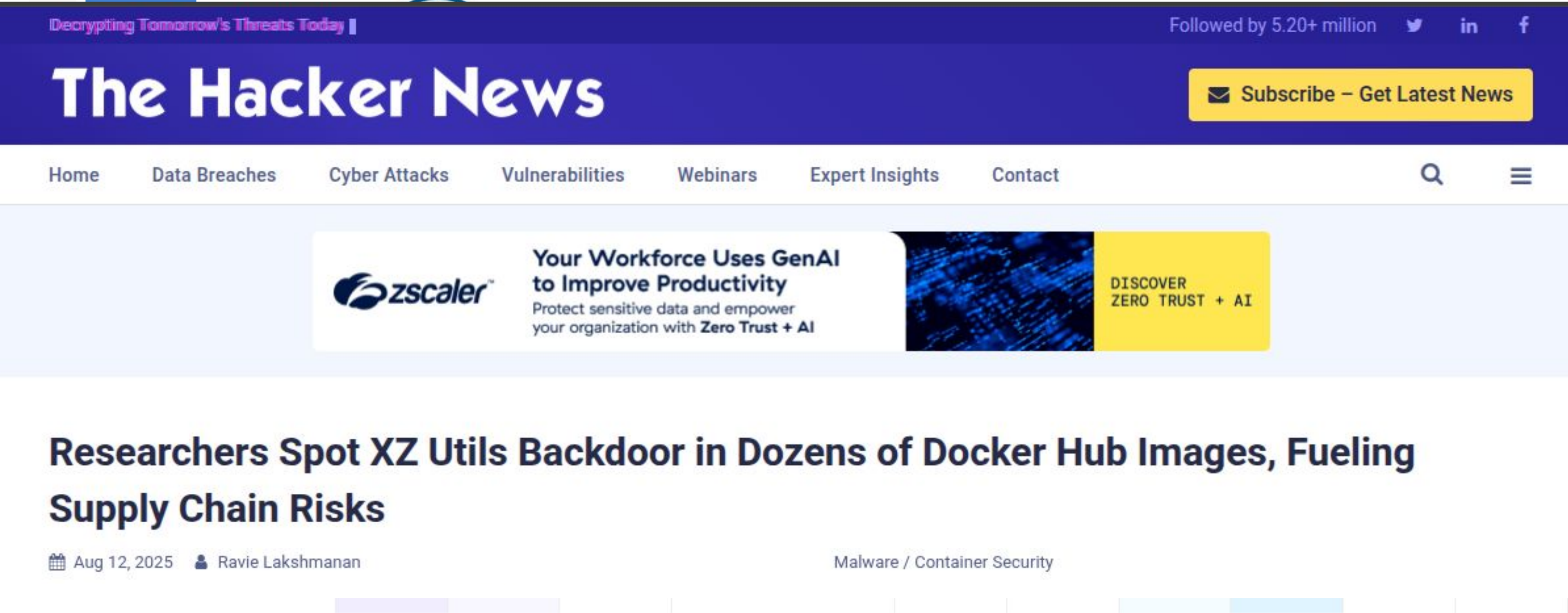
RELATED TOPICS: CYBERSECURITY BEST PRACTICES

Chaing



🚀 Invicti Acquires Kondukto to Deliver Proof-Based Application Security Posture Management

An official website of the United States government Here's how you know



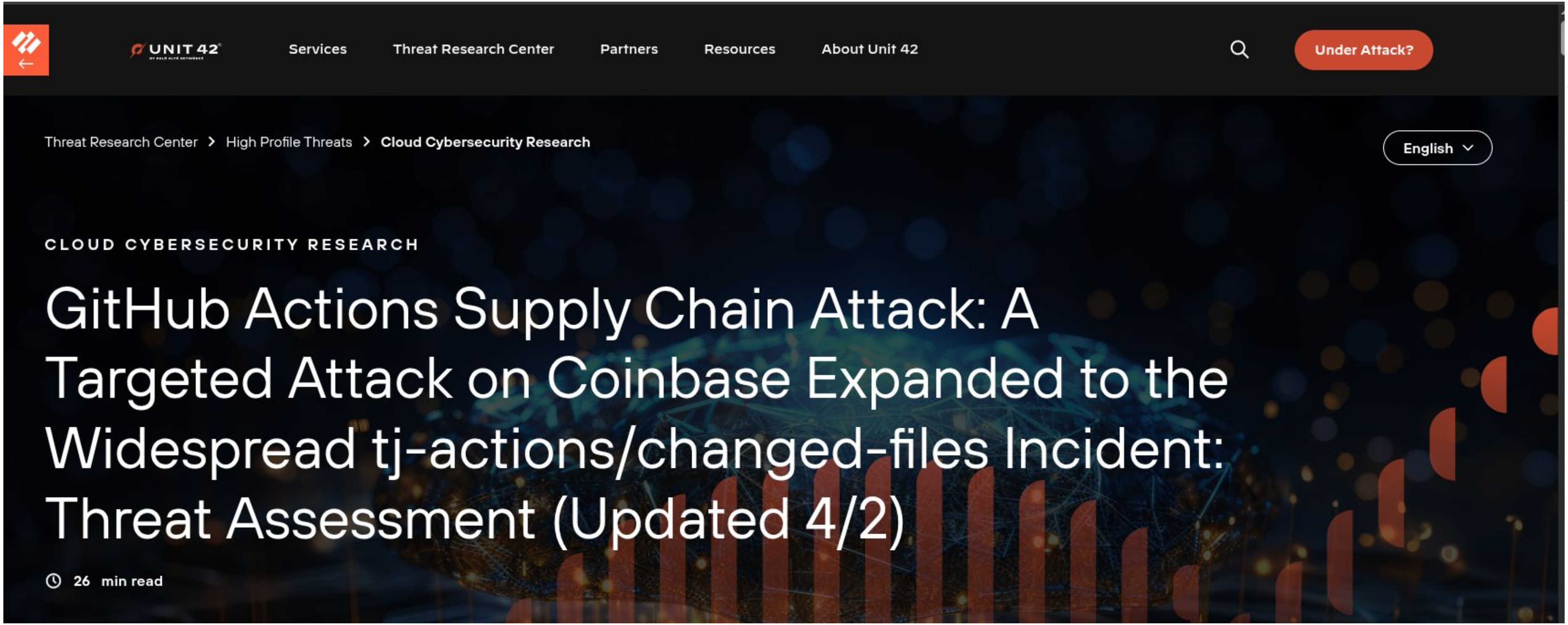
Researchers Spot XZ Utils Backdoor in Dozens of Docker Hub Images, Fueling Supply Chain Risks

Aug 12, 2025 Ravie Lakshmanan

Malware / Container Security

tj-actions/changed-files GHA Compromise (2025)

- Popular GitHub Action, at the time used by 23,000+ repositories
- Attackers injected a payload that dumped the CI/CD runner’s memory, exposing sensitive environment variables and secrets directly to the workflow logs
- Compromised PAT (Personal Access Token) from maintainer used to gain access to the repo
- Malicious commit merged, new tags released + retroactively updated existing tags to point to the same poisoned commit

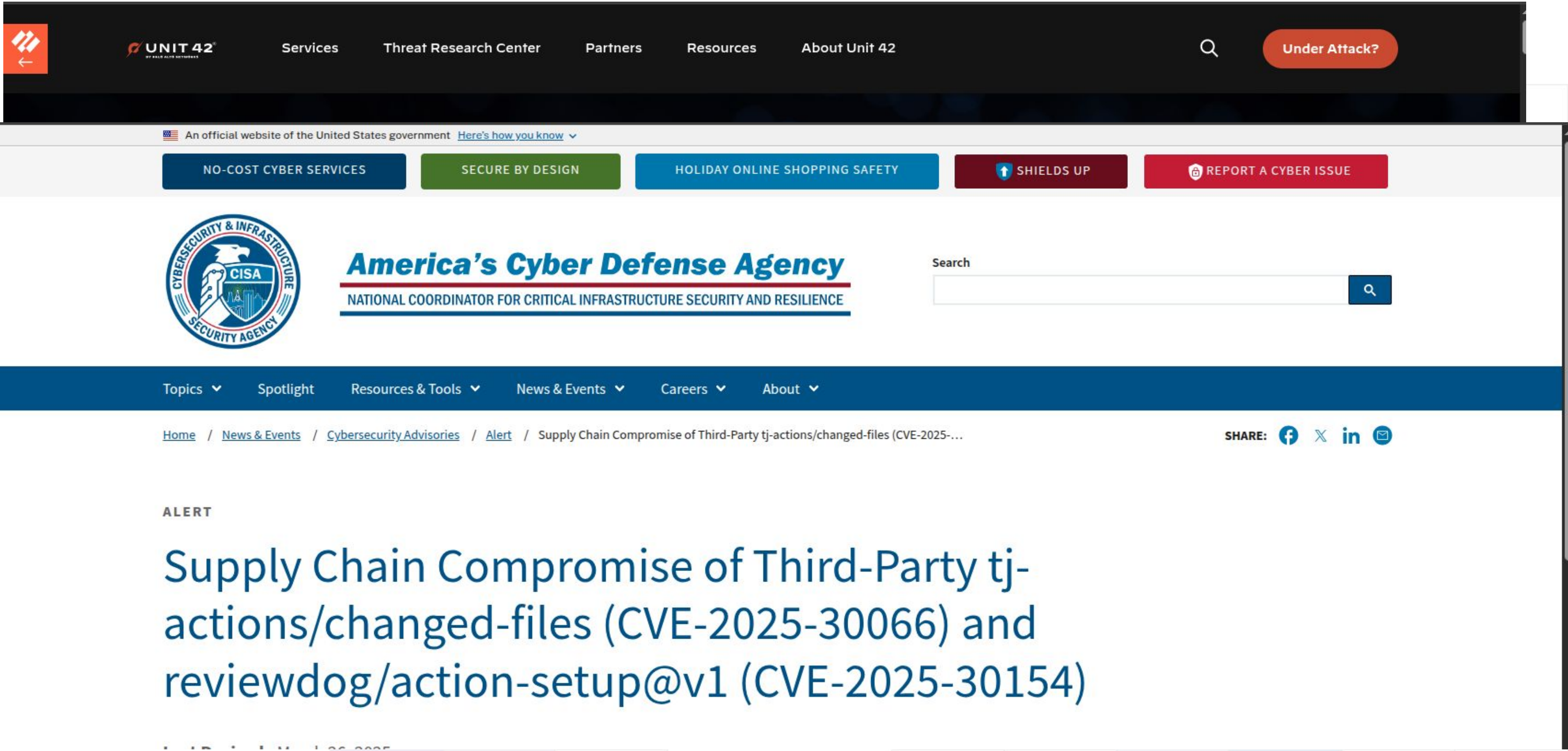


Threat Research Center > High Profile Threats > Cloud Cybersecurity Research

CLOUD CYBERSECURITY RESEARCH

GitHub Actions Supply Chain Attack: A Targeted Attack on Coinbase Expanded to the Widespread tj-actions/changed-files Incident: Threat Assessment (Updated 4/2)

26 min read

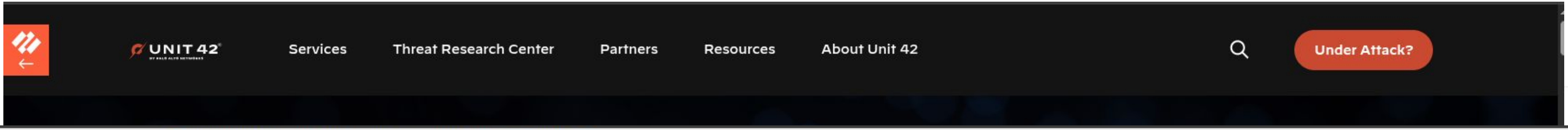


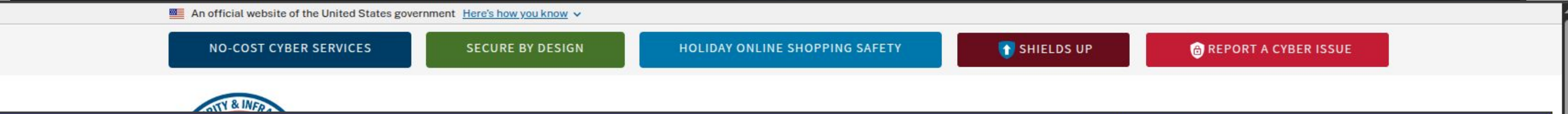
UNIT 42
Services
Threat Research Center
Partners
Resources
About Unit 42
Under Attack?
An official website of the United States government Here's how you know
NO-COST CYBER SERVICES
SECURE BY DESIGN
HOLIDAY ONLINE SHOPPING SAFETY
SHIELDS UP
REPORT A CYBER ISSUE
CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY CISA
America's Cyber Defense Agency
NATIONAL COORDINATOR FOR CRITICAL INFRASTRUCTURE SECURITY AND RESILIENCE
Search
Topics
Spotlight
Resources & Tools
News & Events
Careers
About
Home / News & Events / Cybersecurity Advisories / Alert / Supply Chain Compromise of Third-Party tj-actions/changed-files (CVE-2025-...
SHARE:

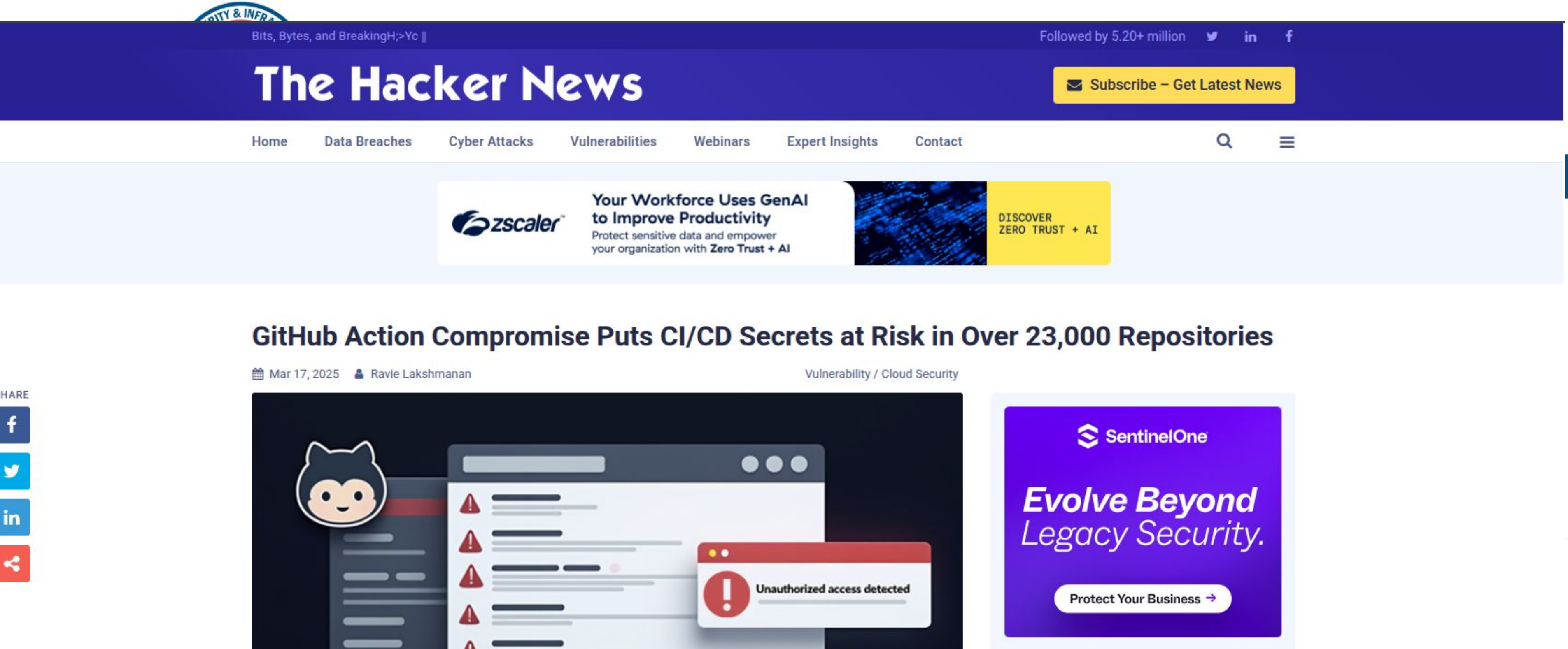
ALERT

Supply Chain Compromise of Third-Party tj-actions/changed-files (CVE-2025-30066) and reviewdog/action-setup@v1 (CVE-2025-30154)

Last Revised: March 26, 2025




An official website of the United States government Here's how you know
NO-COST CYBER SERVICES
SECURE BY DESIGN
HOLIDAY ONLINE SHOPPING SAFETY
SHIELDS UP
REPORT A CYBER ISSUE


Bits, Bytes, and BreakingH;>Yc
Followed by 5.20+ million
The Hacker News
Subscribe – Get Latest News
Home
Data Breaches
Cyber Attacks
Vulnerabilities
Webinars
Expert Insights
Contact
HARE

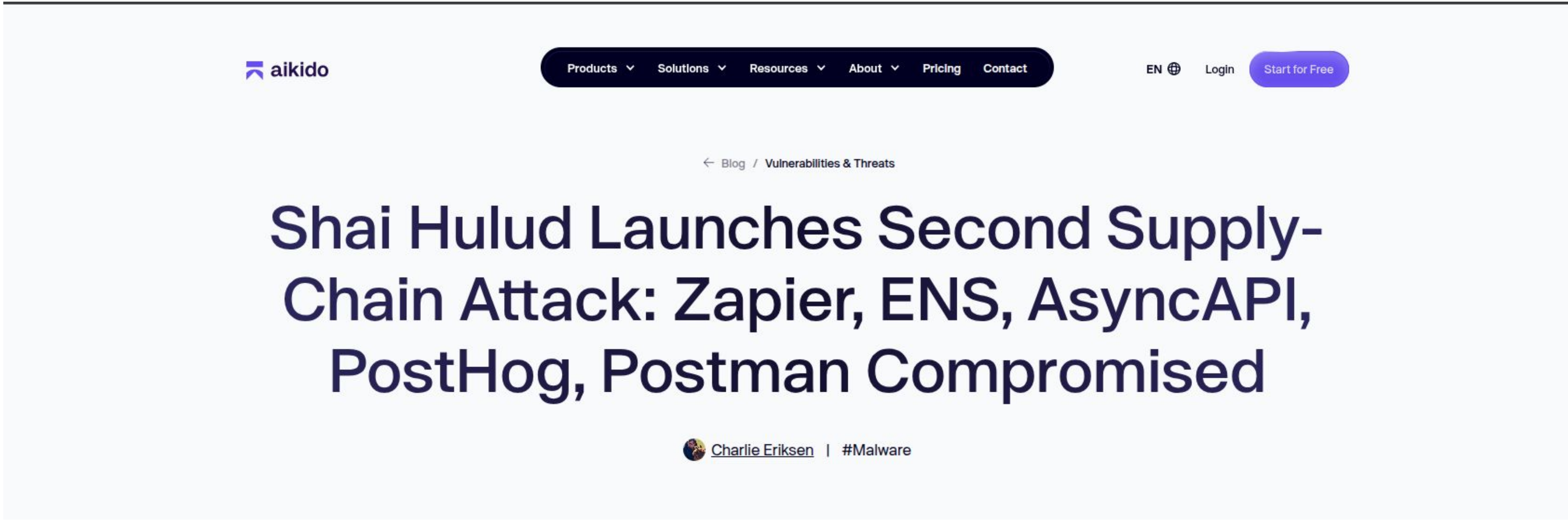
GitHub Action Compromise Puts CI/CD Secrets at Risk in Over 23,000 Repositories

Mar 17, 2025 Ravie Lakshmanan Vulnerability / Cloud Security

Unauthorized access detected

Sha1-Hulud second coming (2025)

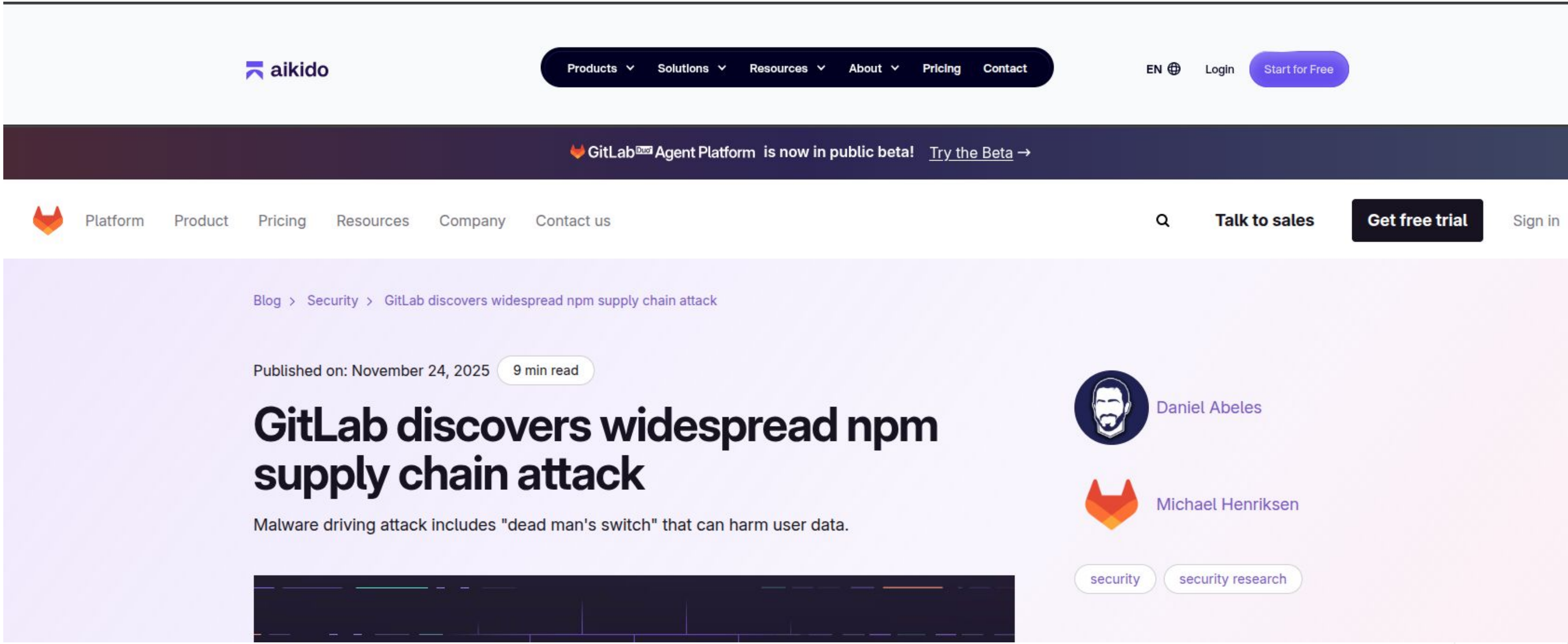
- Sophisticated worm that weaponized the npm **preinstall** hook on infected packages
- Harvests credentials from GitHub, npm, AWS, GCP, and Azure and exfiltrate data to attacker-controlled GitHub repositories
- Auto-infects any other packages maintained by victim
- ~500 packages poisoned (132M+ downloads) and 30,000+ impacted repositories in 72 hours
- Trojanized packages from industry giants like Zapier, Postman, and PostHog spread the worm
- Features a "dead man's switch" capable of destroying user data if its propagation and exfiltration channels are severed



← Blog / Vulnerabilities & Threats

Shai Hulud Launches Second Supply-Chain Attack: Zapier, ENS, AsyncAPI, PostHog, Postman Compromised

Charlie Eriksen | #Malware



Blog > Security > GitLab discovers widespread npm supply chain attack

Published on: November 24, 2025 9 min read

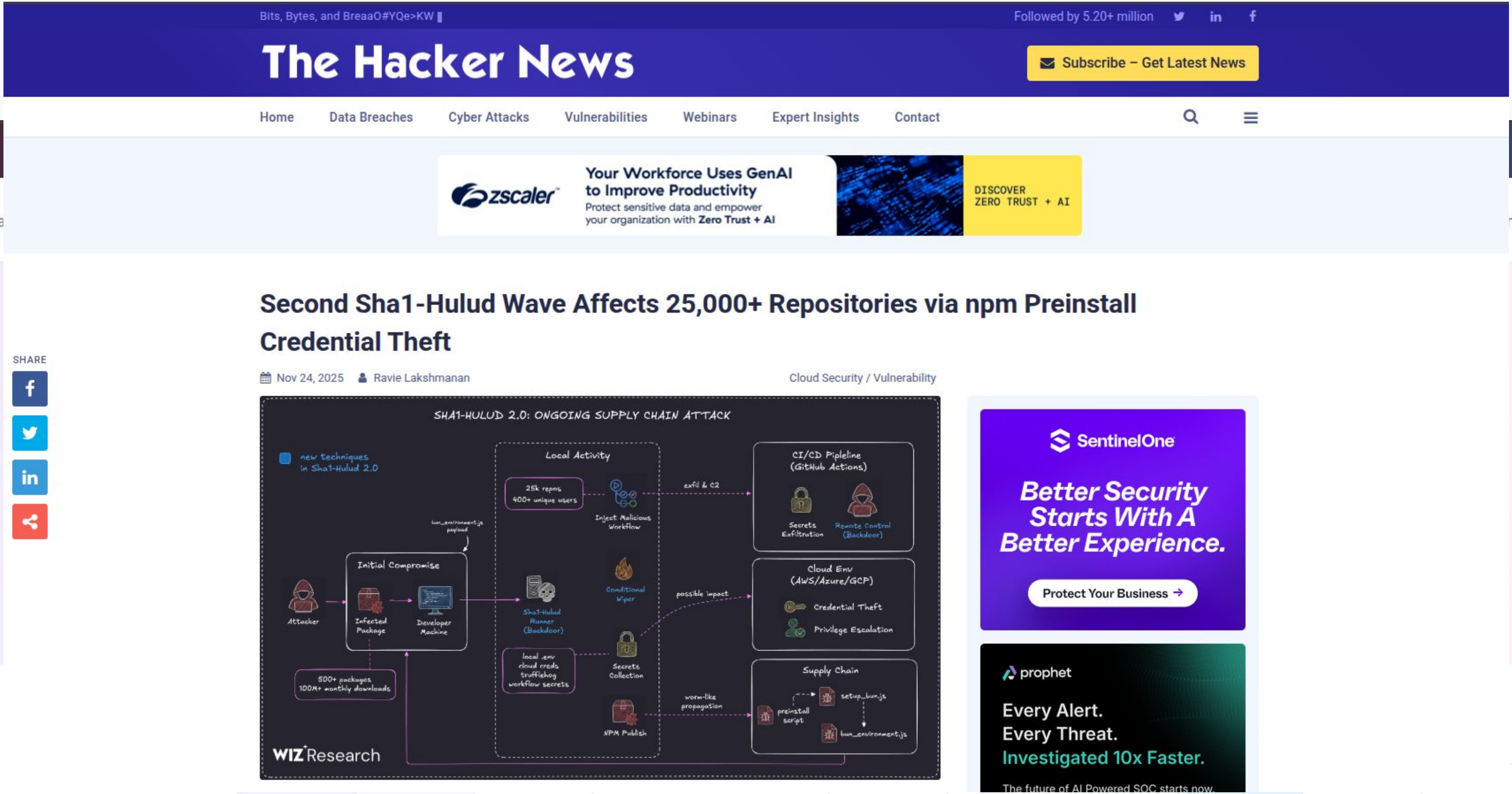
GitLab discovers widespread npm supply chain attack

Malware driving attack includes "dead man's switch" that can harm user data.

Daniel Abeles

Michael Henriksen

security security research



Bits, Bytes, and BreaaO#YQe>KW

Followed by 5.20+ million

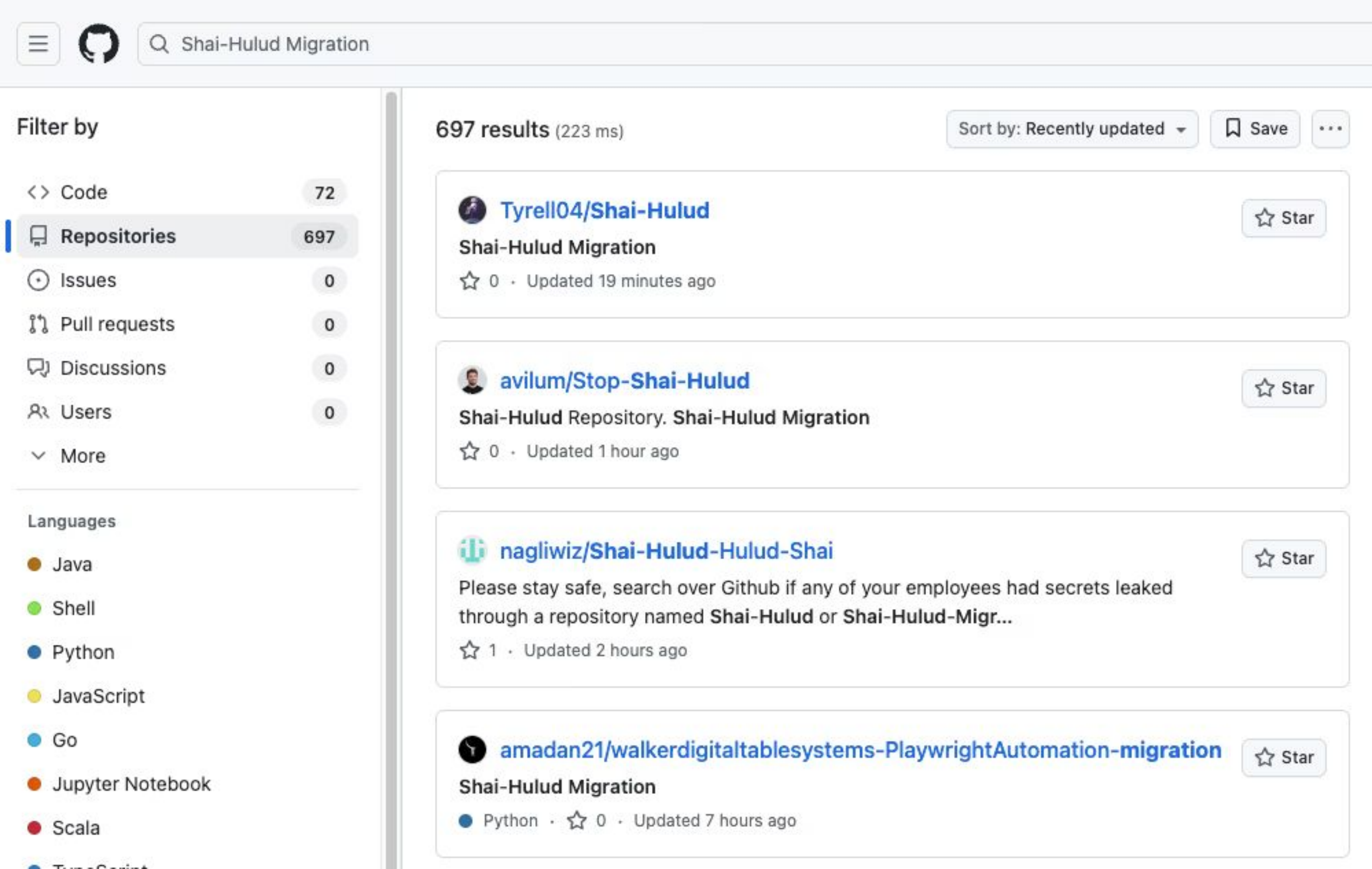
The Hacker News

Second Sha1-Hulud Wave Affects 25,000+ Repositories via npm Preinstall Credential Theft

Nov 24, 2025 Ravie Lakshmanan

Cloud Security / Vulnerability

SHARE
SHA1-HULUD 2.0: ONGOING SUPPLY CHAIN ATTACK
new techniques in Sha1-Hulud 2.0
Attacker
Initial Compromise
Infected Package
Developer Machine
bun_environment.js payload
500+ packages 100M+ monthly downloads
Local Activity
25k repos 400+ unique users
Inject Malicious Workflow
Sha1-Hulud Runner (Backdoor)
Conditional Wiper
local .env cloud creds trufflehog workflow secrets
Secrets Collection
NPM Publish
exfil & C2
possible impact
worm-like propagation
CI/CD Pipleline (GitHub Actions)
Secrets Exfiltration
Remote Control (Backdoor)
Cloud Env (AWS/Azure/GCP)
Credential Theft
Privilege Escalation
Supply Chain
preinstall script
setup_bun.js
bun_environment.js
WIZ Research



697 results (223 ms)

Tyrell04/Shai-Hulud

Shai-Hulud Migration

0 · Updated 19 minutes ago

avilum/Stop-Shai-Hulud

Shai-Hulud Repository. Shai-Hulud Migration

0 · Updated 1 hour ago

nagliwiz/Shai-Hulud-Hulud-Shai

Please stay safe, search over Github if any of your employees had secrets leaked through a repository named Shai-Hulud or Shai-Hulud-Migr...

1 · Updated 2 hours ago

amadan21/walkerdigitaltablesystems-PlaywrightAutomation-migration

Shai-Hulud Migration

Python · 0 · Updated 7 hours ago

Trending Threat Models

What to look for in 2026 threat models and how to stay safe

Trending Threat Models

- OS Packages
    - Packages compromised at build time (harder to spot)
    - Hijacked Repositories
- GitHub Actions / CI
    - Stolen Personal Access Token (PAT)
    - Tag Hijacking
- Local Dev Environment
    - Compromised dependencies
    - Obscure auto-executable scripts

Main Goals: Secret exposure and unauthorized access

Trending Threat Models

- OS Packages
  - Packages compromised at build time (harder to spot)
  - Hijacked Repositories
- GitHub Actions / CI
  - Stolen Personal Access Token (PAT)
  - Tag Hijacking
- Local Dev Environment
  - Compromised dependencies
  - Obscure auto-executable scripts

" Just a developer"
you are here

Main Goals: Secret exposure and unauthorized access

Trending Threat Models: OS Packages

Targeting runtime environments via a poisoned package distribution channel, an attacker infiltrates a project and introduces obfuscated malicious code that is triggered only at build time, bypassing source code scans and CI/CD verifications.

Everything can happen at build time!

How to stay safe

- Maintainer
  - vet contributors, enforce git signing, run checks for malware (malcontent)
- Dev use trusted sources and safe base images - where are your packages coming from? How large is your surface for attack?

Trending Threat Models: GitHub Actions / CI

Attacker gets access to a PAT (personal access token) from a maintainer, publishes a new version of the Action, points other tags to same compromised commit. Then, expose secrets in the env.

How to stay safe

- Maintainer
  - Avoid long-lived credentials! Use Octo-STS to replace PATs with short-lived tokens
  - use rulesets to stop tags being updated
- Dev use digests instead of tags! Digests are immutable. This is also valid for container images.

Trending Threat Models: Local Dev Environment

Attacker injects malicious code in popular ecosystem library; code is triggered in automated execution (such as pre-install hook) at the developer's host, may download additional payload to steal secrets in ENV variables and configuration files.

How to stay safe

- Maintainer all previous precautions
- Dev containerize everything! The risk is immense if you're running your dev environment directly on your host machine. Use safe base images to mitigate risk of container escape. Use safe package sources to mitigate risk of build-time tampering.

Mitigating Risks

Recap: strategies to mitigate software supply chain risks as a developer

Containers are the New Runtime

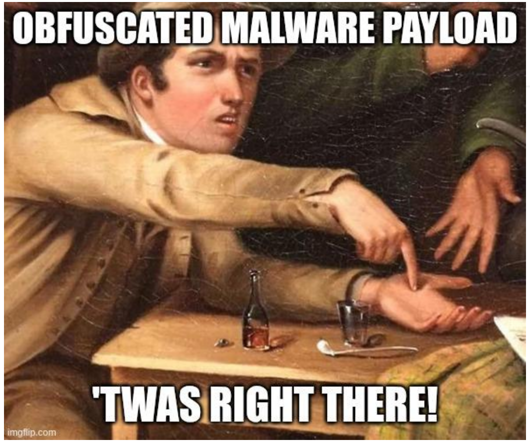
- Everything is containers now!
  - Still, containers are not safe by default
- Pain points with generic images:
  - Bloated images, persistent CVEs, random "official" images, insecure defaults
  - Developers will literally run any base image
- Chainguard Containers:
  - Minimal base images with low-to-zero CVEs
  - Secure-by-default (non-root, locked-down)
- Dockerfile swap → smaller images, fewer vulns

Handling Dependency Chaos

- Most risk lives in libraries / transitive dependencies
- Public registries = uncurated, unpredictable, hard to audit
- Chainguard Libraries:
  - Eliminate threats at build and distribution
  - Curated, continuously rebuilt package feeds (Python, Java, and Javascript)
  - Signed artifacts, strong provenance; compatible with existing tooling (pip, npm, etc.)
  - Same dev UX; safer default sources for dependencies
  - Prevention of pre and post install scripts

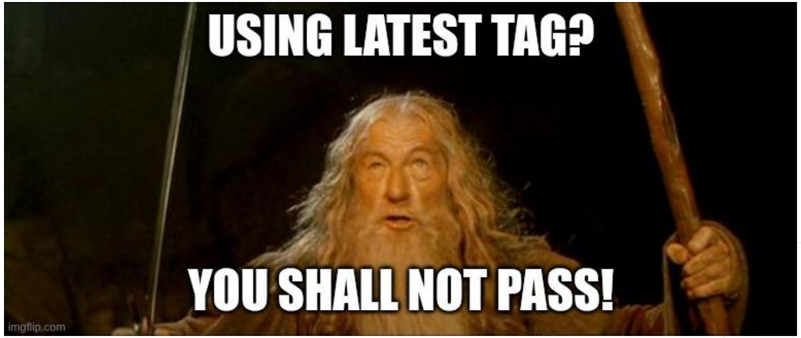
Locking Down CI / CD

- Automated workflows introduce obfuscated risks
  - Pin GitHub Actions and container images to a digest instead of a tag
  - Ban PATs from your organization
  - Use malware scanners when appropriate
  - Establish rulesets and other controls in your repository to protect branches and tags
- Containerized environments need frequent updates
  - Use a tool such as Digestabot or Renovate to update digests and dependencies



Protecting your Cluster with Policy as Code

- Move from opaque security gates → clear, codified rules
- Policy engines ensure only trusted workloads actually run
- Use OSS policy engines (e.g., OPA/Gatekeeper, Kyverno) to:
  - Restrict registries/sources
  - Disallow :latest tag
  - Require signatures/labels tied to SBOMs
- Start in audit/monitor mode; later enforce



Concrete Dev Actions This Sprint

What you can do now with the least amount of friction

Chainguard

Concrete Dev Actions This Sprint

- Audit your containerized workloads
  - Grype scan for CVEs
  - Check image size / dependencies (attack surface)
  - Look for insecure defaults (image runs as root, outdated builds, not pinned by digest)
- Choose one workload / Dockerfile and:
  - Use DFC to migrate to a Chainguard Image
- Capture:
  - Image size change
  - CVE reduction

Demo

Migrating to Chainguard Containers

JANUARY 6 @ 1PM ET

15-Minute Live Demo of Chainguard Libraries

Speakers: Ross Gordon & Angela Zhang

How Chainguard Libraries Protects You From Shai-Hulud and the Next Wave of Open Source Malware

JANUARY 14 @ 1PM ET

Proactive Open Source Library Management

Speaker: Manfred Moser

Join a deep dive into how attackers are inserting thrates into open source libraries and how Chainguard Libraries prevents them before they enter your environment.

Q&A

chainguard.dev

Chainguard

Thank you!

chainguard.dev

Chainguard