The Honest Agenda

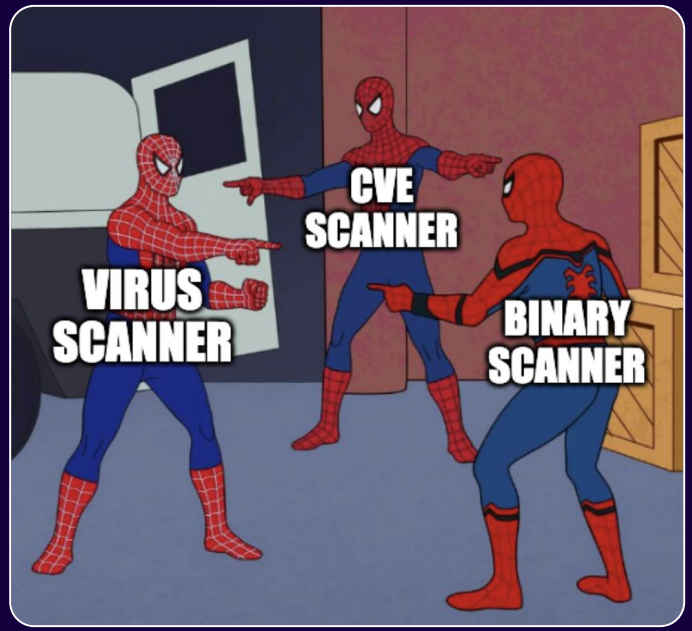
- Some Talk around (CVE) Scanners and Containers and bit of Security
- Some Quality Memes (mostly DIY!)
- 4 Live Demos (hope it works!)

- Q&A

Scanners & the Upfront Trust Problem

From Virus Scanners → CVE Scanners

- Virus scanners:
  - Binary signatures
  - File Patterns
  - Heavier on the resources
- CVE scanners:
  - Package DBs
  - Metadata analysis
  - Lighter on resources
- Different Scanning stages:
  - Build-time
  - Registry
  - Runtime
- Human factor:
  - Conditioned to trust authority (parents, teachers, society)
  - Extend this upfront trust to scanners → believe results blindly



Demo #1: Scanner Comparison (1)

Base Image Scanning:

- 3 base images: `debian:latest, ubuntu:22.04, alpine:3.19`
- Trivy vs. Grype Vulnerability Scanners
- Script `compare.sh`: fetches the images, scans and summarizes results
- Observations and Questions:
  - *(do the demo now before clicking!)*
  - Result discrepancies - why?
  - Why are CVEs without a fix not being displayed for Alpine using Trivy?
  - (Why) Chainguard prefers Grype?
  - *Duration: ~14s runtime for 3 small base images (we'll need that later, ignore it for now)*

Demo #1: Scanner Comparison (2)

Another example - `python:3.13.3-alpine3.22`

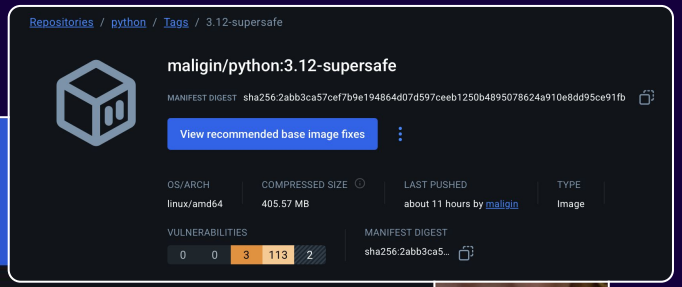
- Trivy vs. Grype
- Script `compare.sh --image-file images.txt`:
  - Fetches the images, scans and summarizes results
- Observations:
  - *(do the demo now before clicking!)*
  - Result discrepancies:
    - 2 Critical, 4 High, 5 Medium (etc.) - w.o. a fix
    - Trivy doesn't display Alpine (and Wolfi) CVEs without a fix (yet)

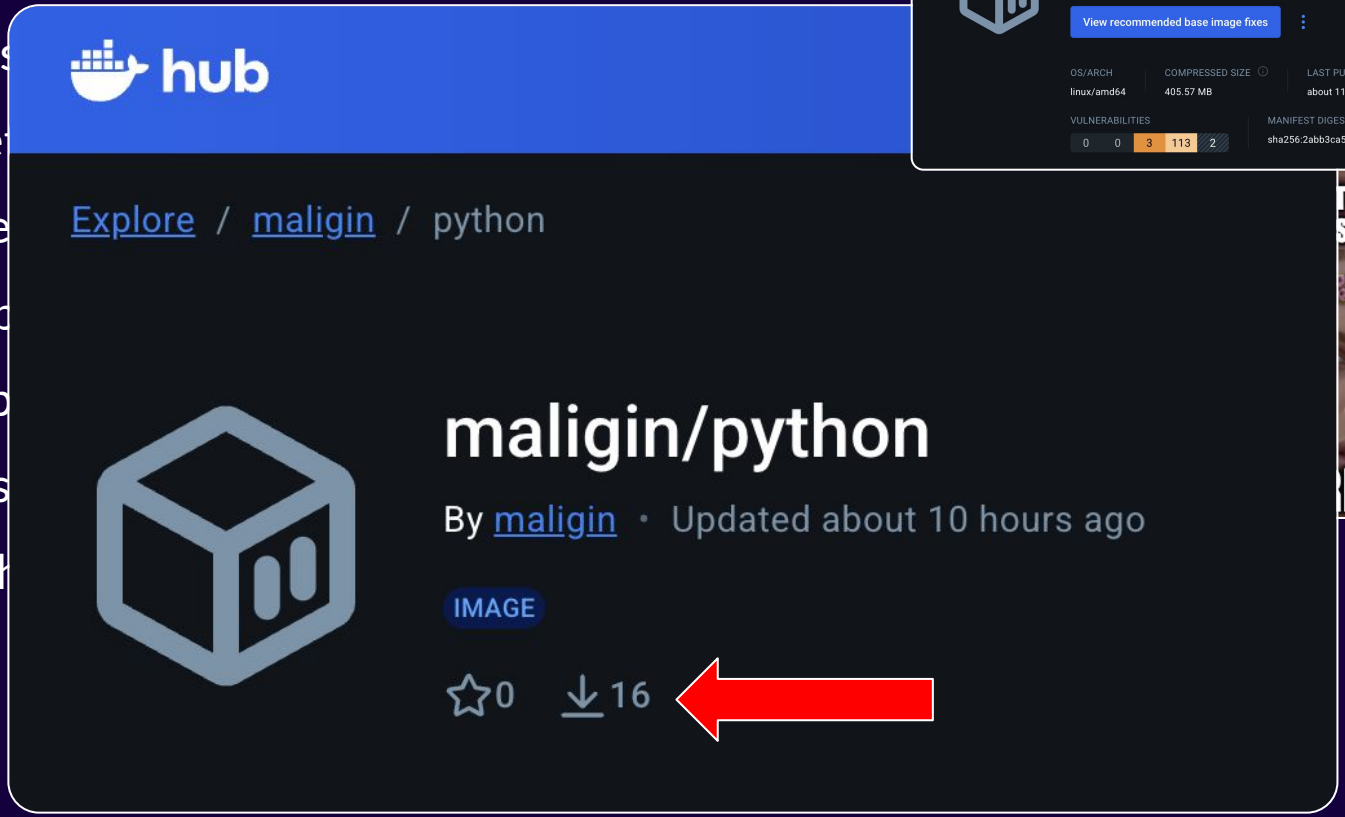
Feature	Supported
Unfixed vulnerabilities	-
Dependency graph	✓

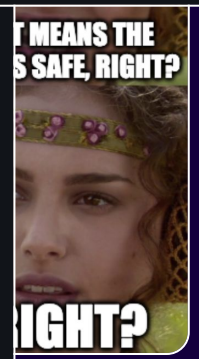
The Manipulation Reality

How Eas

- Me
- File
- Pub
- Exp
- Res
- So h







Demo #4: Sigstore in Action

Container Signing & Verification: `sigstore-demo.sh`

- Sign our Python image
- Push to local registry
- Manipulate metadata (again, similar to our metadata fix)
- Results *(do the demo now before clicking!)*:
  - Verify signature → fails ❌
  - Proof: tampering caught ✅

Real-World Proof: Antropic Misuse, npm, xz utils...

- XZ Utils Backdoor (CVE-2024-3094)
- Antropic Threat Intelligence Report 2025
- Shai-Hulud npm worm — shock download numbers:
  - Infects 147 npm packages
  - Over 2 million weekly downloads

The XZ Utils backdoor (CVE-2024-3094): Everything you need to know, and more

April 3, 2024

'Vibe hacking': how cybercriminals used Claude Code to scale a data extortion operation

Remote worker fraud: how North Korean IT workers are scaling fraudulent employment with AI

A terrifying, self-replicating malwaere has infected npm packages with over 2 million downloads per week - here's how to stay safe

News By Sead Fadilpašić published September 17, 2025

SLSA Framework

Levels of Build Assurance (SLSA Level) and What It Means in a Nutshell:

- 0 - No guarantees
- 1 - Build process documented
- 2 - Hosted build service
- 3 - Non-falsifiable provenance

Security at the Source == Safe Source for Open Source

Where Supply Chain Security Starts - True Security begins at Build Time:

- Built from Source (Wolfi & Chainguard OS)
  - Generate SBOM during the Build Process (apko & melange)
- Full Provenance (sigstore):
  - Declarable and Reproducible Builds = Root of Trust
- Framework: SLSA (Level 3)

Chainguard's Open Source Projects + Useful Tools

- Apko:
  - Apk-based OCI Image Builder
  - GitHub: https://github.com/chainguard-dev/apko
- Melange:
  - Apk Packages Builder using declarative pipelines.
  - GitHub: https://github.com/chainguard-dev/melange
- Chainctl:
  - Used to interact with the Chainguard Console - https://console.chainguard.dev
  - How to Install: https://edu.chainguard.dev/chainguard/chainctl-usage/how-to-install-chainctl/
- Grype:
  - Chainguard's standard OSS Vulnerability Scanner
  - How to install: https://edu.chainguard.dev/chainguard/chainguard-images/staying-secure/working-with-scanners/grype-tutorial/
- Docker File Converter (DFC):
  - Converts existing Dockerfile to Chainguard
  - How to Install and Use: https://edu.chainguard.dev/chainguard/migration/dockerfile-conversion/
- Digestabot:
  - Updates a Image Digest when using the **tag+digest** Pattern
  - GitHub: https://github.com/chainguard-dev/digestabot

Q&A Session (if we’re still on time!)

https://images.chainguard.dev

Thank you!

Peter Andersson, Sales Engineer

images.chainguard.dev