Parochial Suggestions on Open Source Developer Tools for Security

Helping NIST Consider Tools for a Proof-of-Concept DevSecOps Project

John Speed Meyers
Monday, September 19, 2022
I have not performed an exhaustive analysis of all open source security tools related to DevSecOps. (Yikes!)

So what is this based on?

- R&D at IQT Labs on open source software security
- R&D at Chainguard, a software supply chain security company
- My experience as an open source software developer of no particular repute

My Mission

Make suggestions about what mature open source tools NIST should consider for its proof-of-concept and explain why.
Areas of Focus

- secure container base images
- signing software
- static analysis during continuous integration
secure container base images
The FROM command in a Dockerfile is the ultimate import statement.

```bash
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Source: Docker, “Best Practices for Writing Dockerfiles,” 2022,
https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
Popular open source base images can have tens or hundreds of known vulnerabilities.

secure container base images

apko offers an alternative method of building minimal, secure-by-default base images

https://github.com/chainguard-dev/apko
Apko-built images can have dramatically reduced CVE counts

Source: Daily trivy scan of nginx:latest and distroless.dev/nginx:latest
Sigstore: code signing is for everyone
In most* programming language ecosystems, hardly anyone** signs software artifacts.

Why?
- PGP was designed for super-users
- Storing long-lived private keys is a pain
- Acquiring code signing certificates can be onerous and expensive
- among others…

* Maven is an exception.
** < 5 percent

Sigstore simplifies signing

Source: sigstore.dev
Sigstore usage is growing!

from the
user’s
perspective

Involves a command line tool or GitHub Action

https://github.com/sigstore/cosign

$ COSIGN_EXPERIMENTAL=1 cosign sign user/demo

This will open a browser window to authenticate your credentials for the signature.
static analysis during continuous integration
scan all code going into your codebase

scan

*static analysis is cheap and cheerful*

all code

*at least the code you write*

*but scanning dependencies too is nice*
golangci-lint is a combination of linters for Go

https://github.com/golangci/golangci-lint
What it practically means:
checks pass before a commit to main gets merged
Parochial Suggestions on Open Source Developer Tools for Security

Helping NIST Consider Tools for a Proof-of-Concept DevSecOps Project