Independent Container Updates

How to have auto-updated container images in Dangerzone

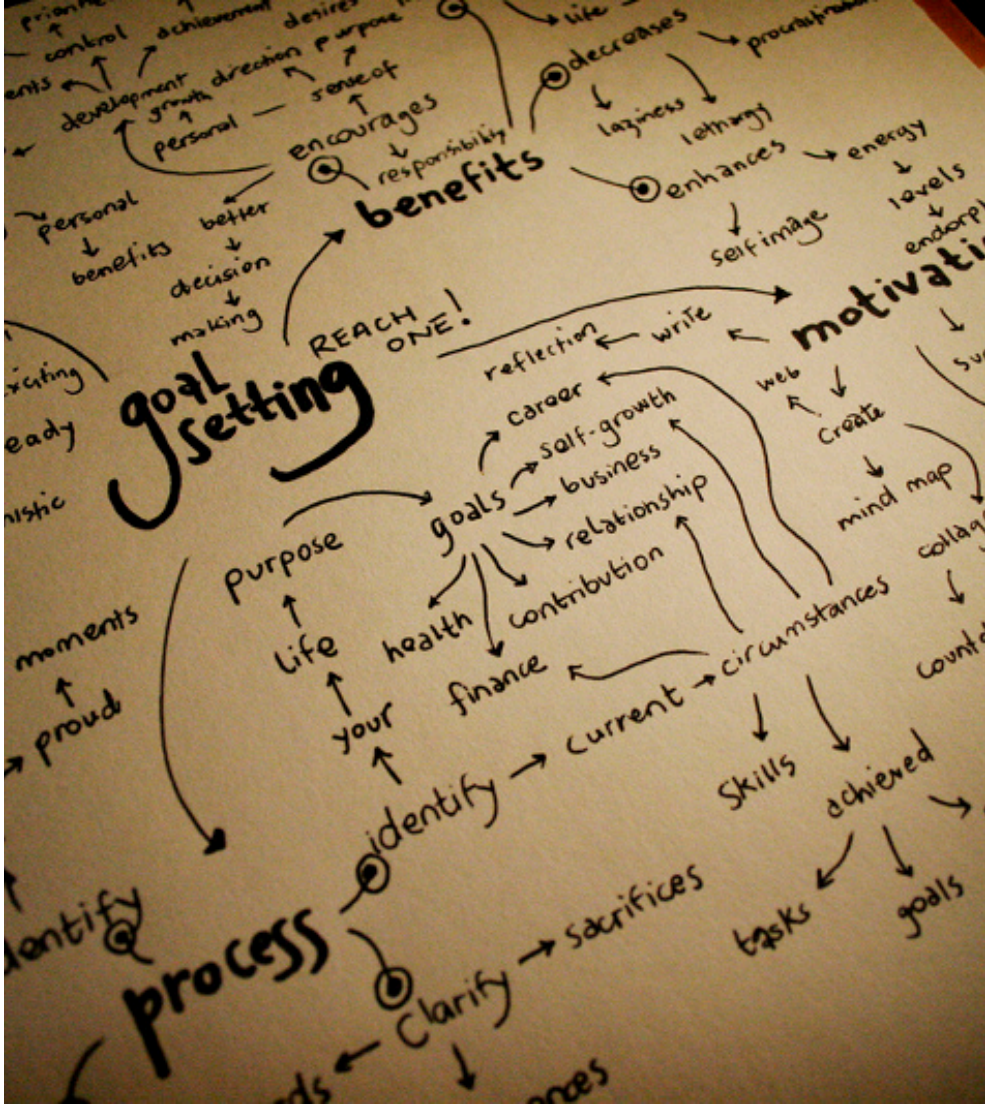
Why independent container updates?

1. Split container images updates from releases

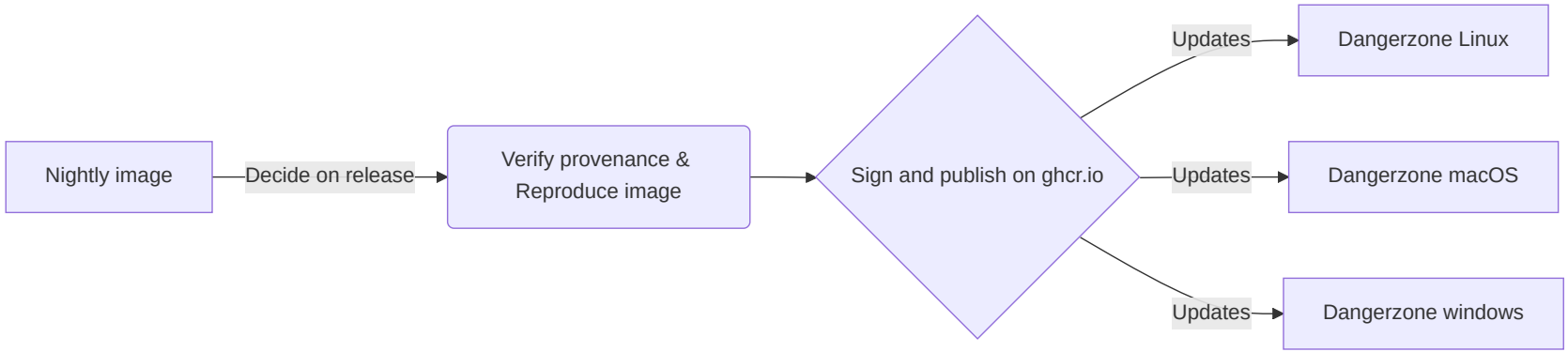
   e.g. avoid issuing a new release for each container image update

2. Quickly patch security holes in container images

   Dangerzone releases take a non-trival amount of work



Overview


Nightly image
Decide on release
Verify provenance & Reproduce image
Sign and publish on ghcr.io
Updates
Dangerzone Linux
Updates
Dangerzone macOS
Updates
Dangerzone windows

Introductory concepts

- provenance
- sigstore/cosign
- reproducible builds

Provenance / Attestations

We want to have some proof that images are built in a specific way.

```
./dev_scripts/dangerzone-image attest-provenance ghcr.io/freedomofpress/dangerzone/dangerzone
  --repository freedomofpress/dangerzone
  --commit "e67fbc1e72ca35a05bf103711e790ef43f1b0978"
  --branch test/image-publication-cosign

🎉 Successfully verified image 'ghcr.io/freedomofpress/dangerzone/dangerzone' and its associated claims:
- ✅ SLSA Level 3 provenance
- ✅ GitHub repo: freedomofpress/dangerzone
- ✅ GitHub actions workflow: .github/workflows/multi_arch_build.yml
- ✅ Git branch: test/image-publication-cosign
- ✅ Git commit: e67fbc1e72ca35a05bf103711e790ef43f1b0978
```

The attestations are stored **in the ghcr.io** registry

Sigstore

A set of tools to sign assets and verify them.

Cosign

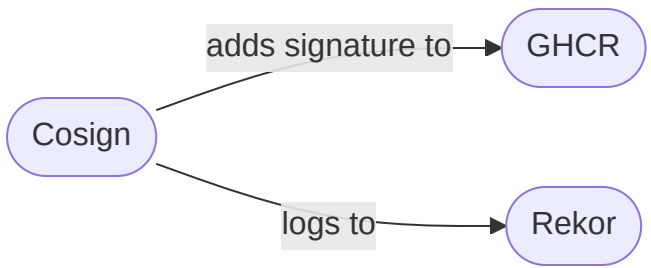
Sign and verify artifacts and containers, with storage in an OCI registry

Rekor

Append-only, auditable transparency log service

GHCR

The Github Container Registry, where images are published



Sigstore (2)

Publishes signatures and attestations to the container registry, as blobs,

at `ghcr.io/freedomofpress/dangerzone/dangerzone`

- Attestations at `sha256-<digest>.att` example logs
- Signatures at `sha256-<digest>.sig` example logs
- Rekor logs allow to tie an identity (Github in this case) to signatures / attestations, and also allows to audit it.

Reproducible builds

- We do not want to trust images built by the Github CI runners blindly
- But we can verify we can reproduce the same containers locally before signing them!

More info on https://reproducible-builds.org/

Public / Private keys?

- Bundle a public key we trust with each release
- Sign the containers we want to distribute with this key
- We're using a yubikey for this

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEoE0CXLGff79fR8KyPnSvOY74UBkt
2sLi+aVFUzS1Qwt4wosxHhcDN2B6QSsLlvgsH82q6qcA6PL2SdS/p4jWGA==
-----END PUBLIC KEY-----
```

Demo time !

Future work

- Use rekor log index to ensure updates are only going upwards
- Let users decide if they want to auto-update or not (opt-out? opt-in?)