#RSAC

RSAConference2022

San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **OST-T01**

Sigstore, the Open Source Software Signing Service

Luke Hinds

Security Engineering Lead, Office of the CTO
Red Hat

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

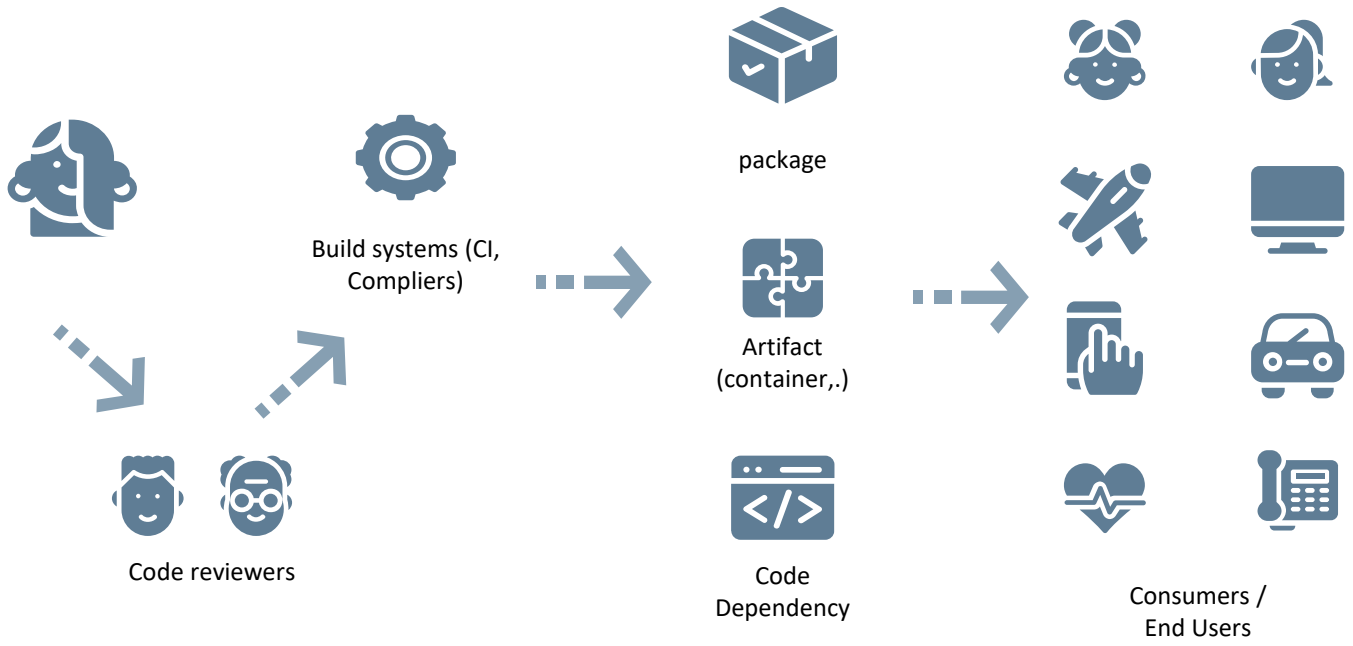
Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

Speaker Intro

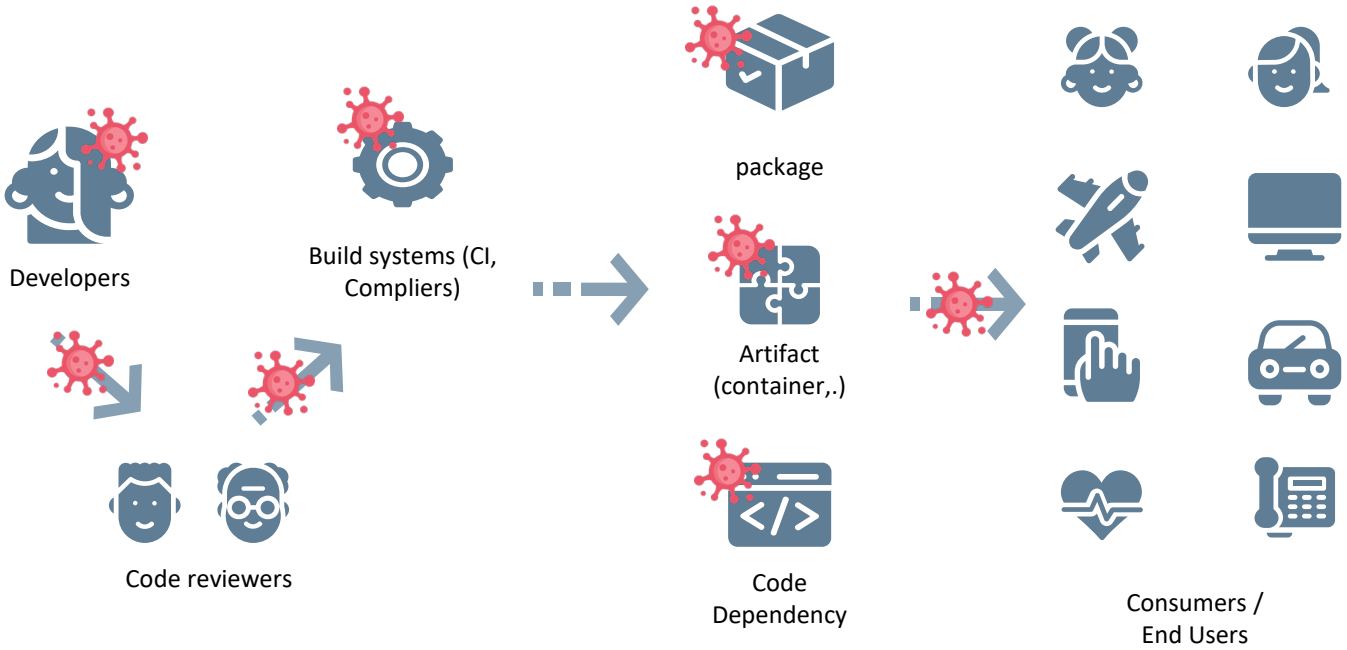
Luke Hinds

- Founder of sigstore
- OpenSSF TAC Member
- Confidential Computing Board
- Bug bounty programs (kubernetes)

What is a software supply chain??



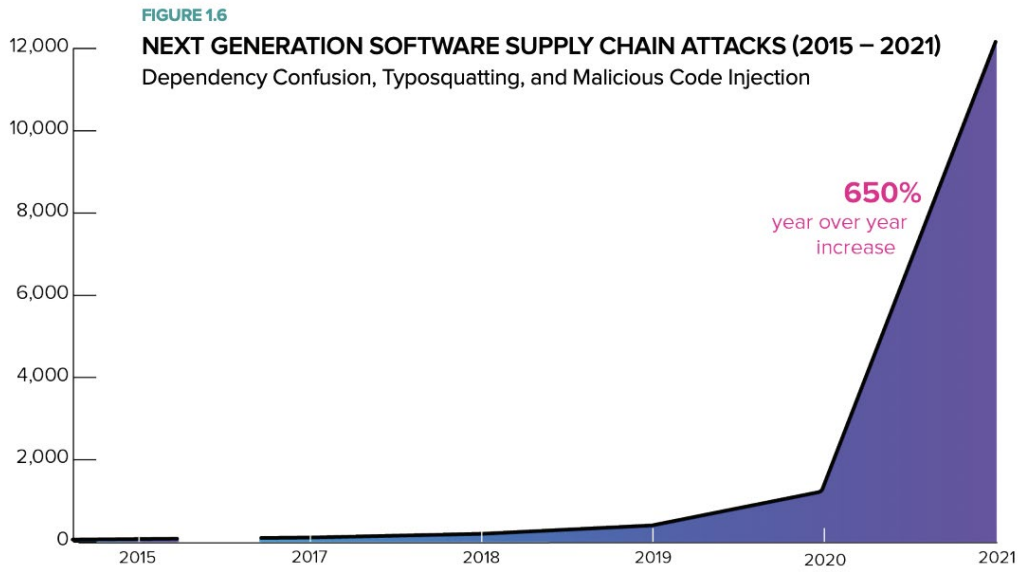
Software Supply Chain Attacks



- Replay / freeze attacks
- Compromised keys
- Account Compromise
- Swapped hashes
- Compromise of build systems
- Easy reconnaissance (open configuration)
- Typosquatting
- Developer Burnout ‘act out’

sigstore

Software Supply Chain Attacks



650%

Increase in supply chain attacks in 2021

Sonatype's State of the Software Supply Chain

What can be done?

Who is signing?

What gaps are present?

1. A significant lack of code signing adoption

1. A lack of credible & trustworthy provenance

1. Key management still a challenge!

Digital Signatures

So what does software signing get us?

So what does software signing get us?

Verifies **integrity** of content (signature cannot be verified if even 1 bit is altered)

So what does software signing get us?

Non-repudiation (i.e. entity that possesses the private key can not state that they did not sign the artifact)

So what does software signing get us?

Authentication: if a private key is conceptually bound to an identity, the sender of signed messages can be assumed

So what does software signing get us?

If signature includes a (third-party signed) **timestamp**, consumers can have greater assurances of when the artifact was signed

Who is signing (Critical Projects)?

System	Signing Tools	Trust Model
Linux Kernel	PGP	Mostly TOFU (trust on first use)
Node.js Core	PHP	PKs in git repo (insecure)
Kubernetes	sigstore	sigstore
Python	PGP	Keys on website (insecure)
OpenSSL	PGP	Keys on website (insecure)

Who is signing (Package Managers)?

System	Signatures	Cert Systems	In Use
PyPi	Optional	PGP	Rarely
NPM	PHP	NA	0%
Maven Central	Required	PGP/x509	100%
Containers	PGP	PGP/x509	Rarely
Go	N/A		
Ruby	Optional	x509	Rarely
Crates.io	No	No	No

So why are we not signing?

So why are we not signing!?

⊗ Managing security of private keys is difficult and expensive

So why are we not signing!?

⊗ Handling key rotation and key compromise

So why are we not signing!?

⊗ Fear of Key compromise

So why are we not signing!?

⊗ Costs!

So why are we not signing!?

Tooling is cumbersome to use and has not been modernised…

MOTHERBOARD
TECH BY VICE

People Are Freaking Out That PGP Is 'Broken'— But You Shouldn't Be Using It Anyway

ars TECHNICA — BIZ & IT | TECH | SCIENCE | POLICY | CARS | GAMING & CULTURE

OPSEC IS STILL HARD —

Op-ed: I'm throwing in the towel on PGP, and I work in security

Schneier on Security

Blog | Newsletter | Books | Essays | News | Talks | Academic | About Me

Home > Blog

Giving Up on PGP

Filippo Valsorda wrote an excellent essay on why he's giving up on PGP. I have long believed PGP to be more trouble than it is worth. It's hard to use correctly, and easy to get wrong. More generally, e-mail is inherently difficult to secure because of all the different things we ask of it and use it for.

Search
Powered by DuckDuckGo
Go
Blog Essays Whole site

AMIT KATWALA | SECURITY | 17.05.2018 12:21 PM

We're calling it: PGP is dead

The EFail vulnerability threatened to punch a hole in PGP's security.

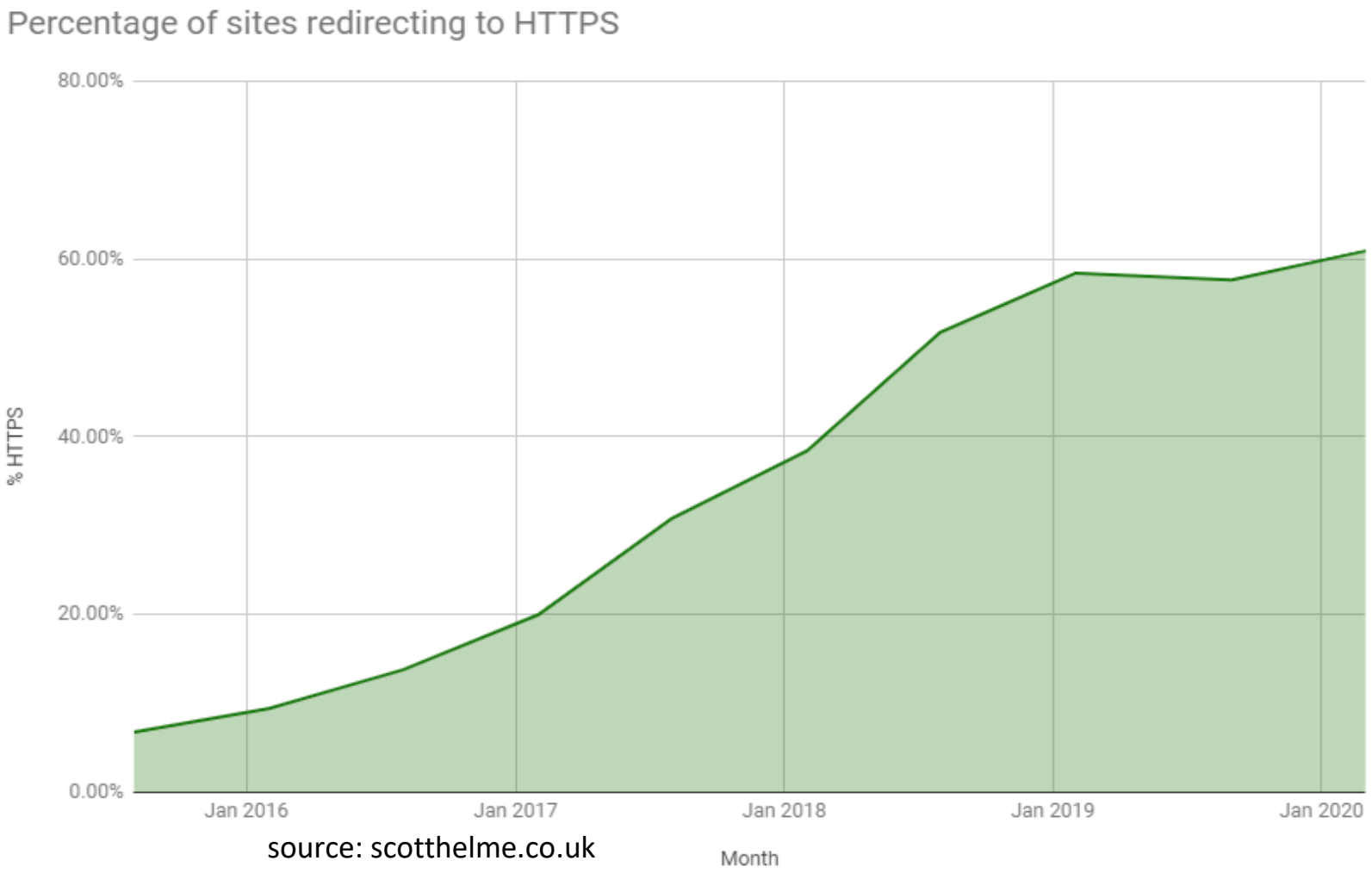
What if signing and key management were greatly simplified and provided for free to all?

Why reinvent the wheel?

Case Study:

HTTPS

HTTPS based websites 2015 - 2020



source: scotthelme.co.uk

Month	% HTTPS
Mar 2020	60.93%
Sep 2019	57.66%
Feb 2019	58.44%
Aug 2018	51.78%
Feb 2018	38.42%
Aug 2017	30.78%
Feb 2017	19.96%
Aug 2016	13.76%
Feb 2016	9.39%
Aug 2015	6.71%

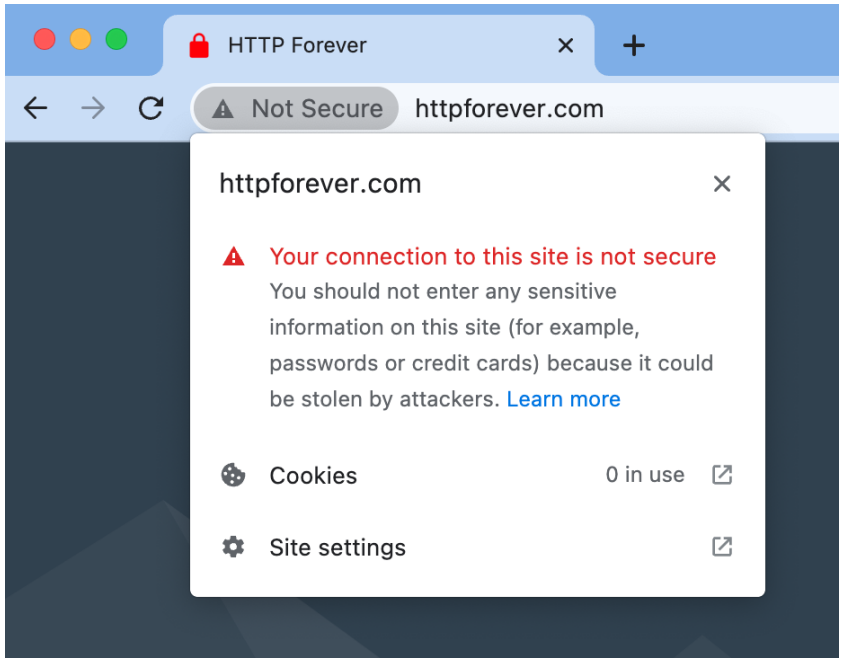
What happened in 2015 and beyond?

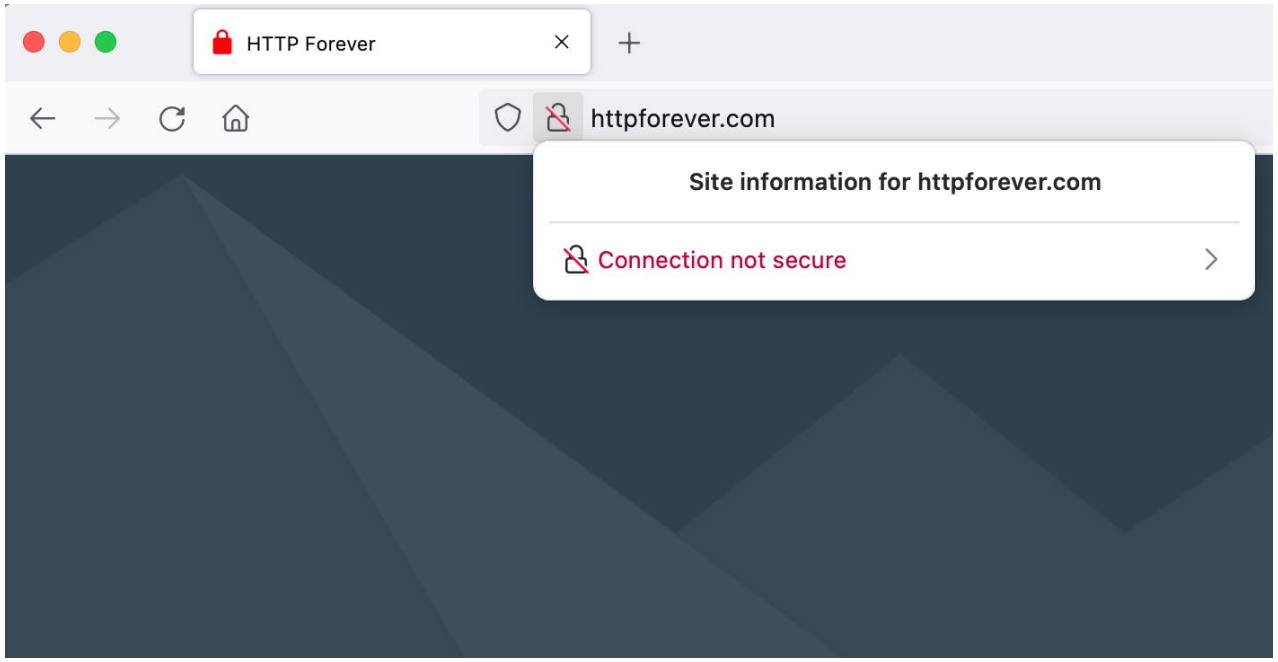
What happened in 2015 and beyond?

September, 2015	First **Let's Encrypt** Certificate issued
October, 2015	Trusted by all major browsers
July, 2018	Chrome v68 (non HTTPS "insecure")
November, 2020	Firefox 83 introduces HTTPS-Only Mode

Month	% HTTPS
Mar 2020	60.93%
Sep 2019	57.66%
Feb 2019	58.44%
Aug 2018	51.78%
Feb 2018	38.42%
Aug 2017	30.78%
Feb 2017	19.96%
Aug 2016	13.76%
Feb 2016	9.39%
Aug 2015	6.71%

Browsers Close in…





What if we could do the same for software?

What is sigstore?

- Under the OpenSSF (open source software foundation)
- Provides software signing as a public good service
- Combination of services and clients
- Can be deployed privately / internal network

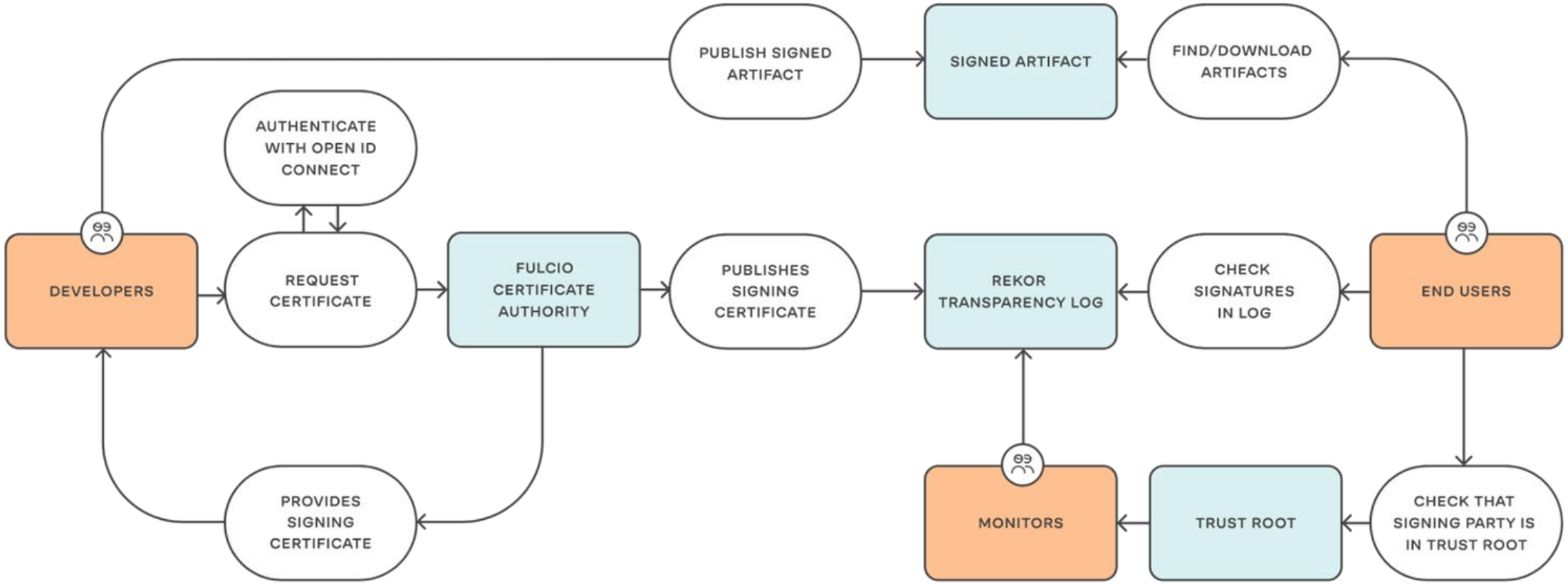
sigstore projects

- **Fulcio**: CA issues code signing certificates based on OIDC identity
- **Rekor**: signature transparency log - append-only, immutable
- **Cosign**: container signing tool
- **Many other clients**: maven, rust, ruby gems, python..

Other formats supported

- OpenID Connect Signing
- KMS (AWS, Azure, GCP, Vault)
- PKCS11 (YubiKey, HSM)
- Algs RSA, ECDSA, Ed25519, GPG

sigstore OIDC signing



sigstore

sigstore ODIC signing

GitHub Action:

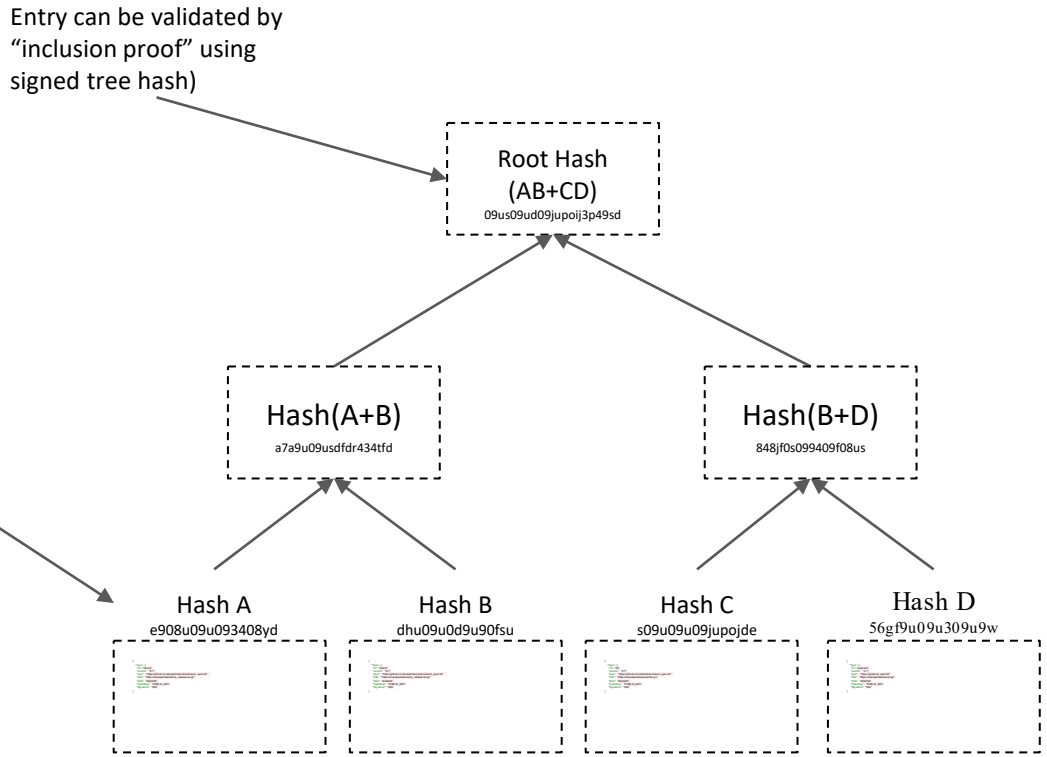
```
"Subject Alternative Name":
"https://github.com/lukehinds/widgets/.github/workflows/docker-publish.yml@refs/heads/main"
```

Email:

```
"Subject Alternative Name": "lhinds@redhat.com"
```

sigstore public transparency log

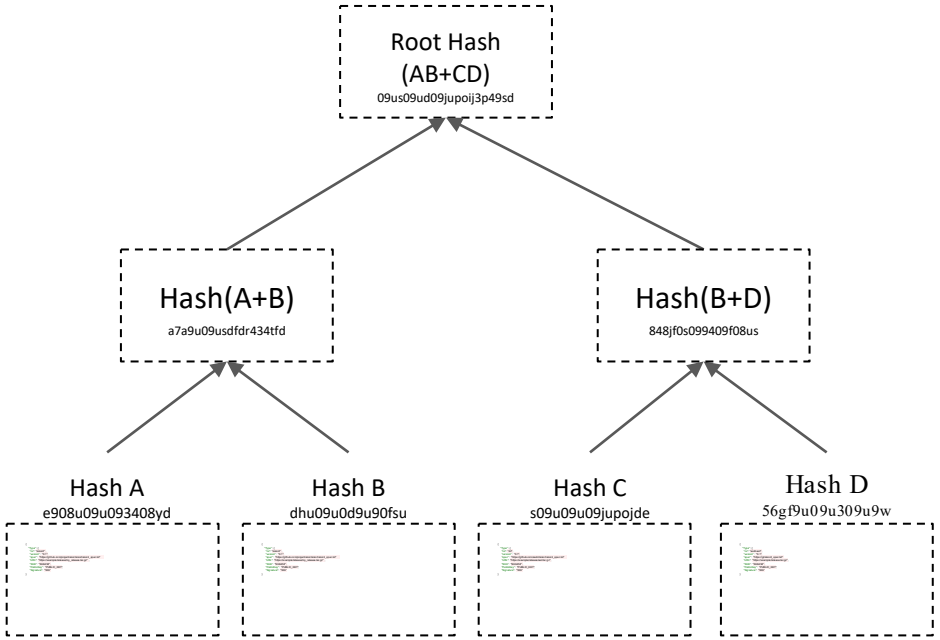

```
{
  "type": "rekord",
  "apiVersion": "0.0.1",
  "spec": {
                "signature": {
        "format": "pgp",
        "URL": "https://example/release/my_release.tar.gz.sig",
        "publicKey": { "url": "https://example/keys/public_key.pgp" },
                },
                "data": {
        "url": "https://example/release/my_release.tar.gz",
        "hash": {"algorithm": "sha256", "value": "83jfj8we89903uhejw88..!" }
        }
    }
}
```

sigstore

logs are publically transparent?

- Publicly verifiable
- Has my key been used?
- Has my OIDC been used?
- What is the blast radius of a key compromise?
- Who has signed X digest?



Open Source Adoption

GitHub Actions

Blog

Engineering Community Product Security Open Source Enterprise Education Company Changelog

Product Security

Safeguard your containers with new container signing capability in GitHub Actions

GitHub has partnered with the OpenSSF and Project Sigstore to add container image signing to our default "Publish Docker Container" workflow.

Kubernetes

ZDNet

Trending Innovation Security Business Finance Education Home & Office More

MUST READ: Misinformation needs tackling and it would help if politicians stopped muddying the water

Kubernetes taps Sigstore to thwart open-source software supply chain attacks

The Kubernetes project takes a step forward in shielding users from supply chain attacks on its users.

sigstore

Maven Central

Log4Shell Help for Central Publishers (**Last Update 2022-01-03**). Click for more info

All Systems Operational

The Central Repository Documentation

Search

WELCOME PUBLISH CONSUME FAQ CENTRAL SEARCH SUPPORT WHO IS SONATYPE? CHANGELOG STATUS & NEWS

STATUS & NEWS

Central Status

Latest News

News Archive

Maven Central and Sigstore

Table of contents

What's Next?

As custodians of the Maven Central registry, it's important to us here at Sonatype to ensure Central remains accessible, secure and modern for users and publishers.

With this in mind, over the past few years we have been investing heavily in Maven Central with the goal of modernizing the platform, improving the security of publishing and consumption and providing the developer experience consistent with expectations of contemporary software registries. This is a wide ranging effort that is expected to improve upon nearly every aspect of the platform.

As we work through design and planning activities, the emergence of sigstore as a solution to address provenance concerns that are critical to software supply chains is particularly exciting to us.

Many More

Other projects onboarding

Ruby Gems

PyPi

Rust Crates

JReleaser

Alpine Linux

Npm

Nuget

Maven Central

Vitess

How to find us.

Project Website

https://sigstore.dev

code

https://github.com/sigstore

Slack

https://sigstore.slack.com

twitter

https://twitter.com/projectsigstore