# project sigstore

## software signing for the masses!

Luke Hinds

# Me: Luke Hinds

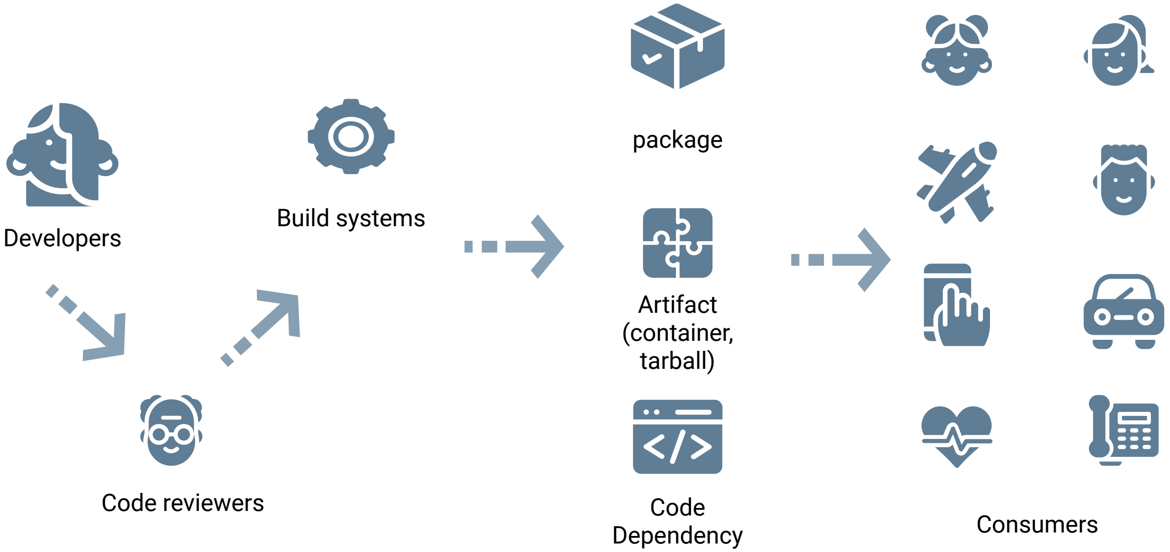
- Security Engineering Lead (Emerging Tech, CTO, Red Hat)
- Kubernetes Security Response Team
- OpenSSF TAC / Confidential Computing Board Member
- Software engineer

🐦 @decodebytes

# Today’s Talk

- Introduction to supply chain attacks
- Some example attacks
- Introduction to sigstore
- Quick demonstration of sigstore

# What is a software supply chain?



# What is a software supply chain?

Git Code Repos Held to Ransom – Thousands Hacked

by

Many pri

demandi

September 6, 2019

**CirclCI data breach exposed customer GitHub and Bitbucket logins**

The sc

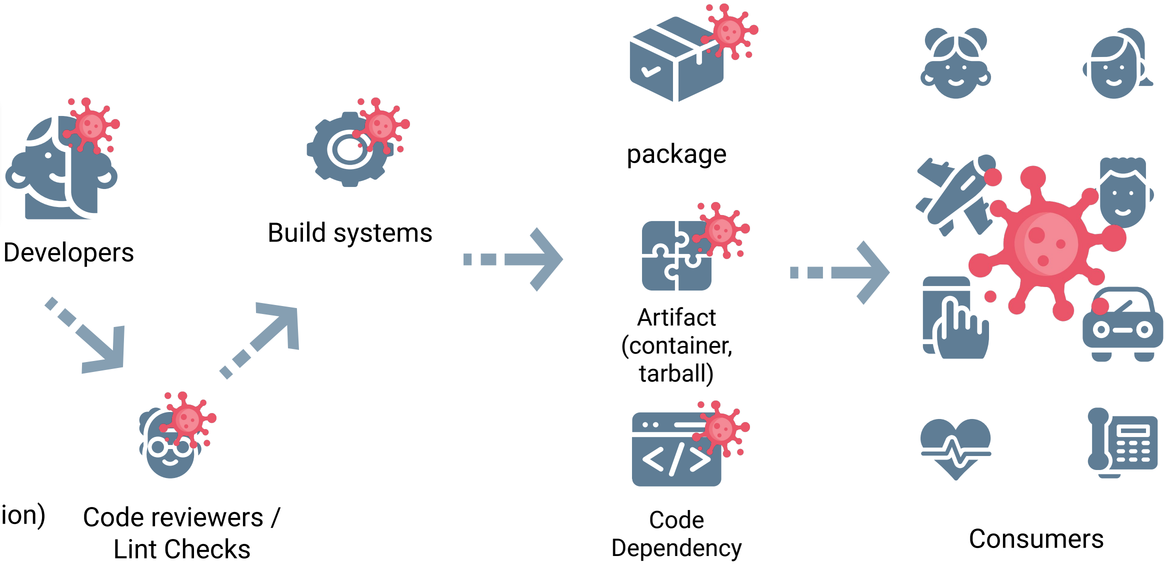
incide

**GitHub hacked, millions of projects at risk of being modified or deleted**

By Se

A new type of supply-chain attack with serious consequences is flourishing

New dependency confusion attacks take aim at Microsoft, Amazon, Slack, Lyft, and Zillow.



Developers

Code reviewers / Lint Checks

Build systems

package

Artifact (container, tarball)

Code Dependency

Consumers

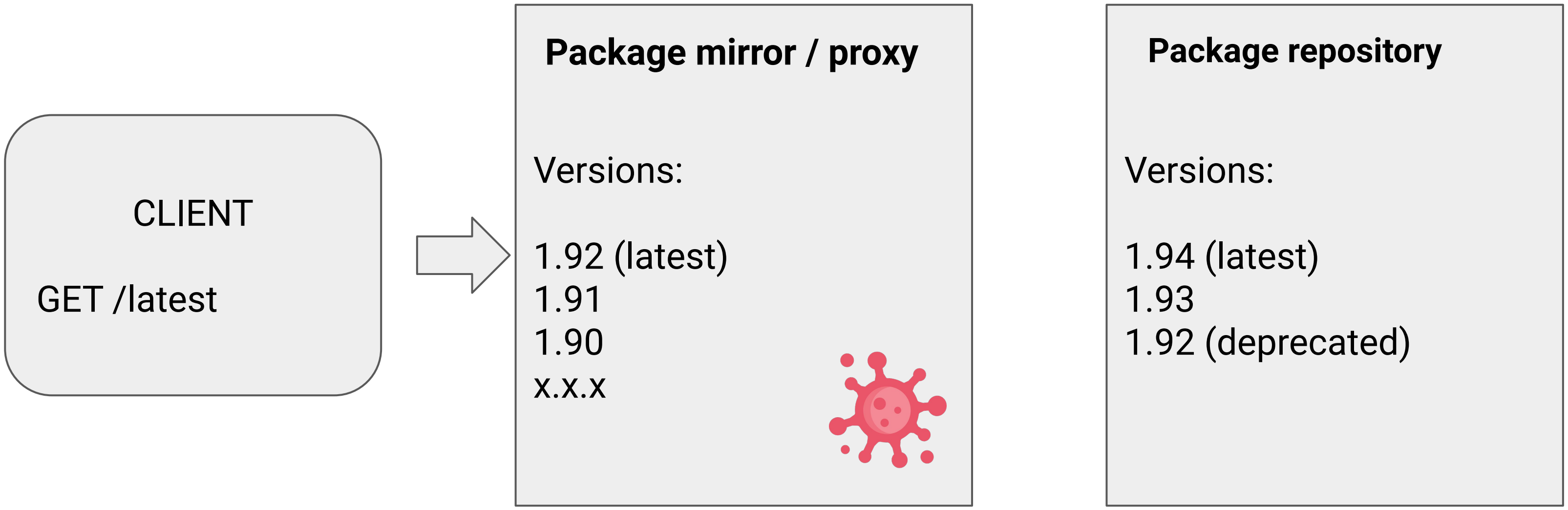
- Replay / Rollback attacks
- Compromised keys
- Account Compromise
- Swapped hashes
- Compromise of build systems
- Unwanted Payloads
- Easy reconnaissance (open configuration)
- Typosquatting

* “In **2021** the world witnessed a **650% increase** in software supply chain **attacks**, aimed at exploiting weaknesses in **upstream open source ecosystems.**”

* https://www.sonatype.com/resources/state-of-the-software-supply-chain-2021

# Common supply chain attacks

# Common attacks: Rollback / Replay attacks



CLIENT

GET /latest

**Package mirror / proxy**

Versions:

1.92 (latest)
1.91
1.90
x.x.x

**Package repository**

Versions:

1.94 (latest)
1.93
1.92 (deprecated)

Version **1.92** contains: CVE-2021-12345

# Common attacks: Key Compromise

**CLIENT**

GET kernel.org/linux-5.14.7.tar.xz
GET kernel.org/linux-5.14.7.tar.sign

**TOFU**

I am Linus Torvalds and I signed this!

- Hard to detect!

# Common attacks: Key Compromise

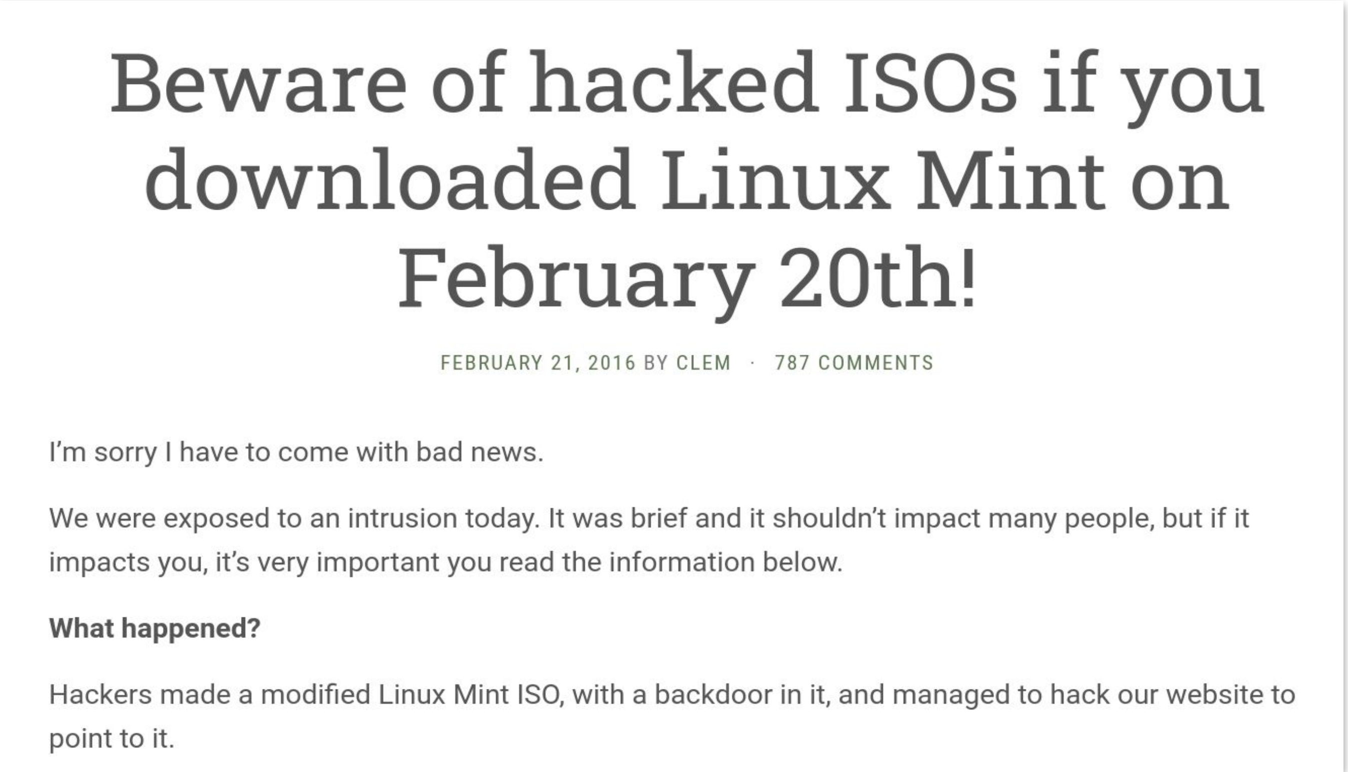
- Malicious code was inserted into update system plug-in called ***SolarWinds.Orion.Core.BusinessLayer.dll***

- This compromised dll was signed by a seemingly valid, but compromised SolarWinds certificate.

*US agencies — including parts of the Pentagon, the Department of Homeland Security, the State Department, the Department of Energy, the National Nuclear Security Administration, and the Treasury — were attacked*

solarwinds

# Common Attacks: Swapped hashes / artifacts



## Beware of hacked ISOs if you downloaded Linux Mint on February 20th!

FEBRUARY 21, 2016 BY CLEM · 787 COMMENTS

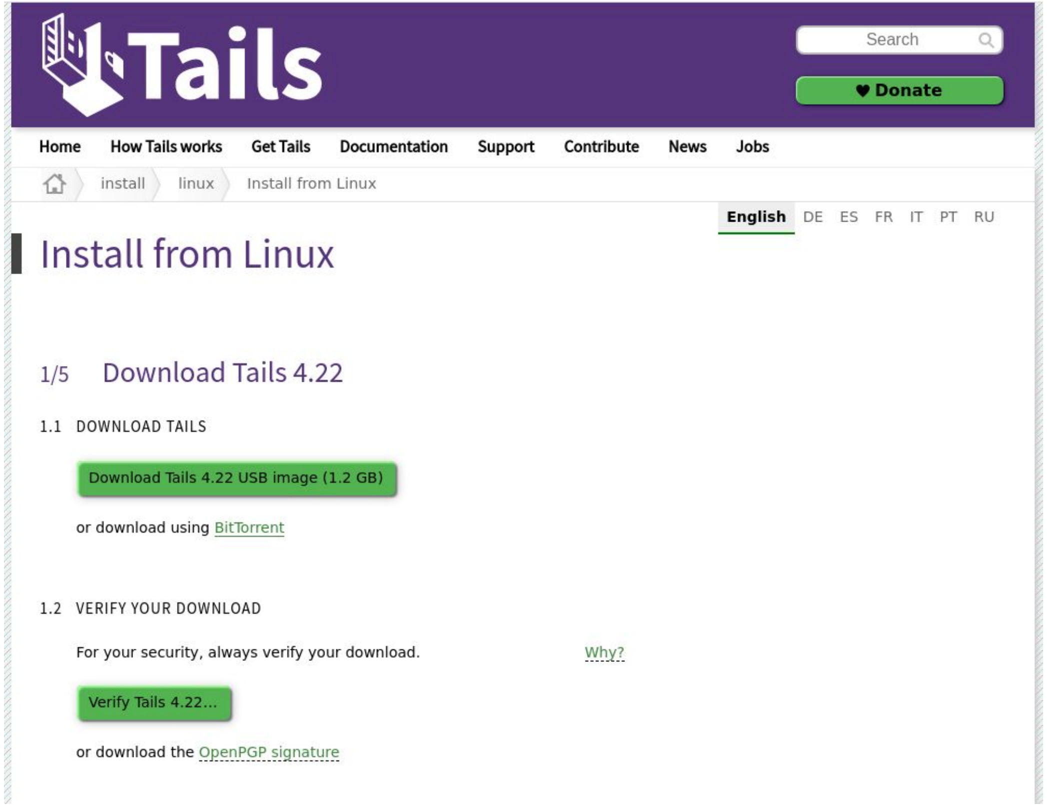
I'm sorry I have to come with bad news.

We were exposed to an intrusion today. It was brief and it shouldn't impact many people, but if it impacts you, it's very important you read the information below.

**What happened?**

Hackers made a modified Linux Mint ISO, with a backdoor in it, and managed to hack our website to point to it.

# Common Attacks: Swapped hashes / artifacts



Tails
Search
♥ Donate
Home How Tails works Get Tails Documentation Support Contribute News Jobs
install › linux › Install from Linux
English DE ES FR IT PT RU

## Install from Linux

### 1/5 Download Tails 4.22

1.1 DOWNLOAD TAILS

Download Tails 4.22 USB image (1.2 GB)

or download using BitTorrent

1.2 VERIFY YOUR DOWNLOAD

For your security, always verify your download. Why?

or download the OpenPGP signature

# Common attacks: Compromise of build systems

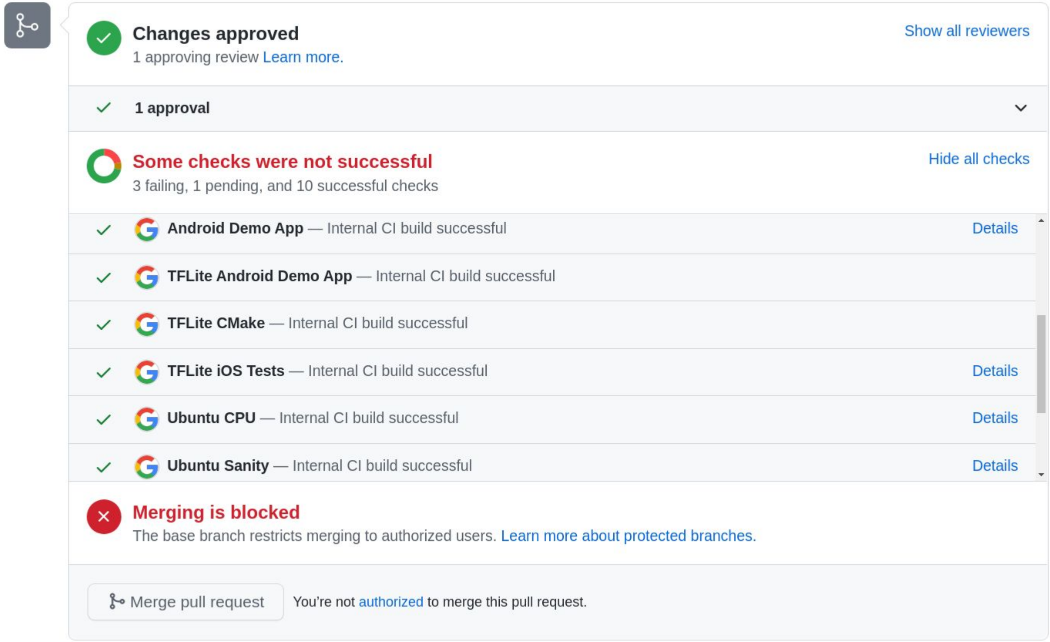
- **codecov** is run in hundreds of CI systems (Kubernetes, HashiCorp, Twilio,Rapid7, Monday.com, and e-commerce giant Mercari.)
- An attacker replaced an bash uploader script to CI to leak secrets
- * The attack successfully run a huge amount of customer networks

**  US investigators examining the case told Reuters on Tuesday that the attackers responsible for the hack managed to exploit not only Codecov software, but also potentially used the organization as a springboard to compromise a huge number of customer networks."*

https://about.codecov.io/security-update/

# Common attacks: Resource hijacking

```
git push origin btc-miner
```



**Changes approved**
1 approving review Learn more.

Show all reviewers

✓ **1 approval**

**Some checks were not successful**
3 failing, 1 pending, and 10 successful checks

Hide all checks

✓ **Android Demo App** — Internal CI build successful Details

✓ **TFLite Android Demo App** — Internal CI build successful

✓ **TFLite CMake** — Internal CI build successful

✓ **TFLite iOS Tests** — Internal CI build successful Details

✓ **Ubuntu CPU** — Internal CI build successful Details

✓ **Ubuntu Sanity** — Internal CI build successful Details

**Merging is blocked**
The base branch restricts merging to authorized users. Learn more about protected branches.

Merge pull request You're not authorized to merge this pull request.

# Easy reconnaissance

- Build system configuration is open to scrutiny.
- Attacks can look to leverage integration tests as means to back door code.
- Far too many instances of stuff like…

```
curl https://example.com/install.sh | sudo bash
```

# Typosquatting

jellyfish | jellyflsh

# Typosquatting and a quick quiz ?

```
python3-dateutils
```

✗

```
dateutils
```

✓

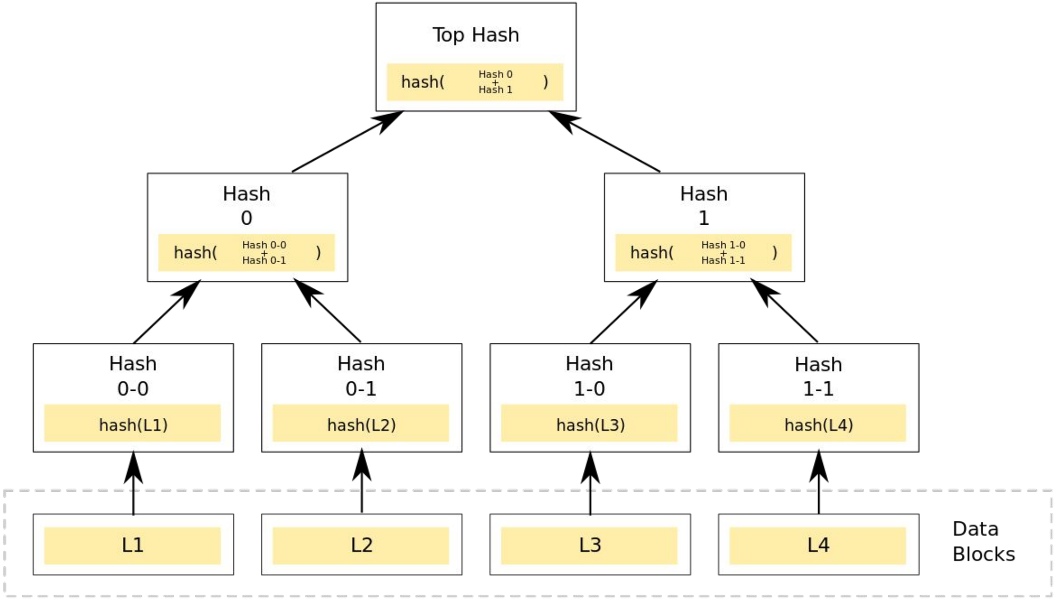
# So what we need is..

- Transparency (detect key compromise)
- Non repudiation (authenticity), not TOFO
- Tamper resistance (protect integrity)
- Time stamping (rollback / replay attacks)

What existing projects / technologies might be leveraged here?

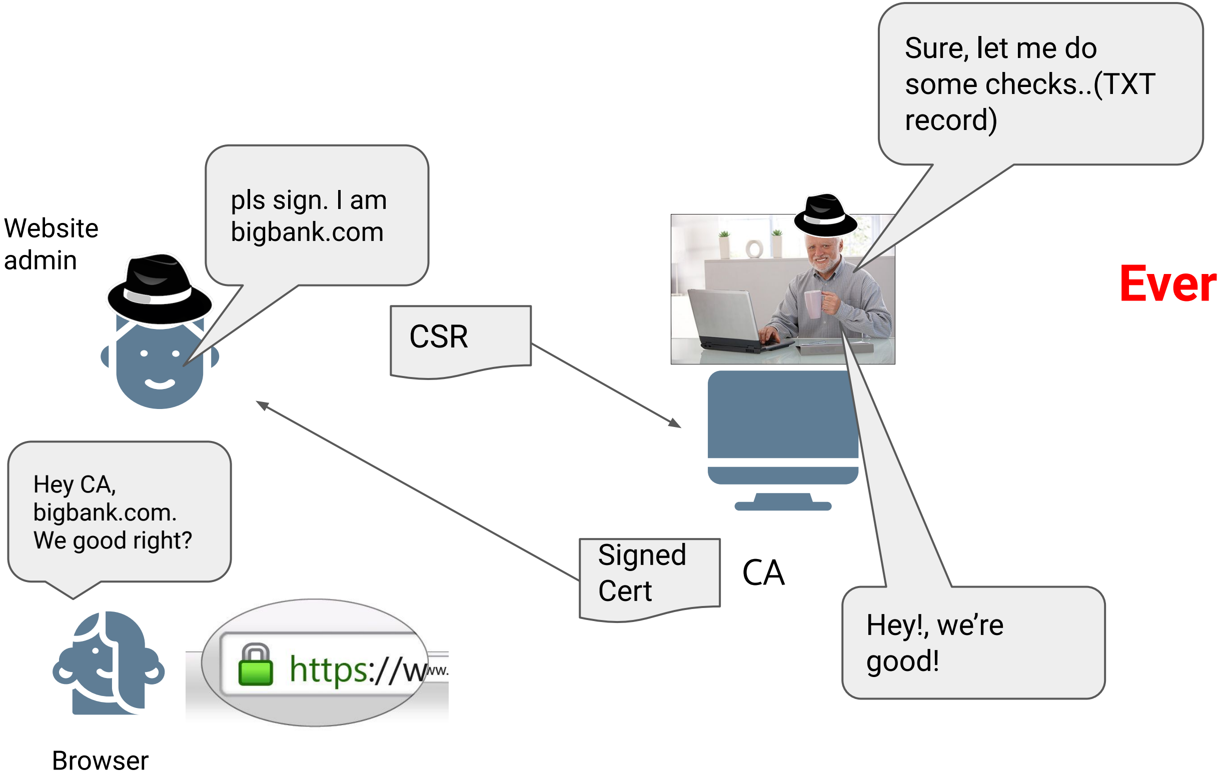
# Transparency logs….

- **Merkle trees** - used in Git, blockchain, and certificate transparency systems
- Append-only, “immutable”
- Tamper-evident: changing a leaf breaks the whole structure
- Hashing (sha265)  is relatively computationally inexpensive



Example use of a transparency log.....**certificate transparency**

# Before Certificate Transparency ...



Website admin

pls sign. I am bigbank.com

CSR

Sure, let me do some checks..(TXT record)

CA

Signed Cert

Hey!, we're good!

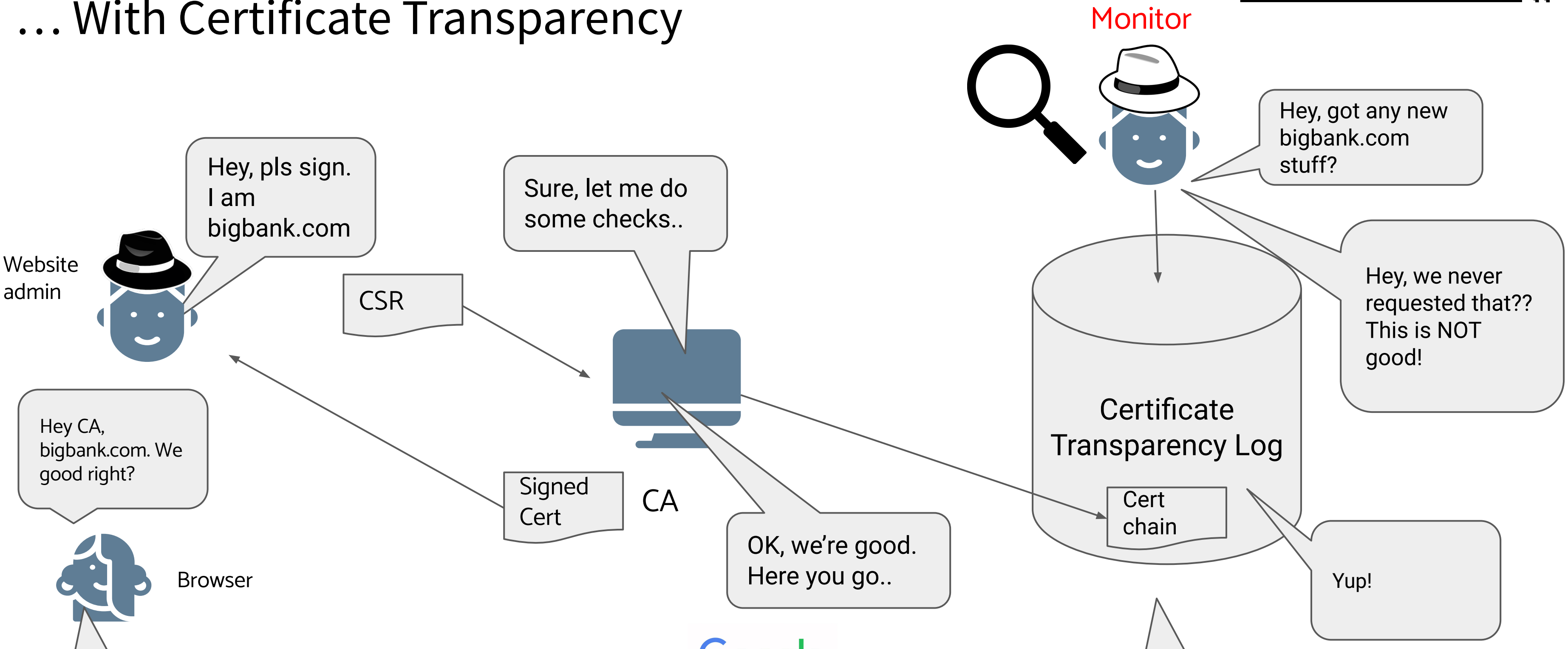
Hey CA, bigbank.com. We good right?

https://www.

Browser

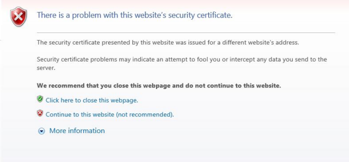
**Eugh!**
**Everyone is trusting the CA!**

# … With Certificate Transparency


Monitor
Hey, got any new bigbank.com stuff?
Hey, we never requested that?? This is NOT good!
Website admin
Hey, pls sign. I am bigbank.com
Sure, let me do some checks..
CSR
Signed Cert
CA
OK, we're good. Here you go..
Certificate Transparency Log
Cert chain
Yup!
Hey CA, bigbank.com. We good right?
Browser

Hey CTL, you got bigbank.com right? (**Expect-CT**)

Nope!


There is a problem with this website's security certificate.

The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

**We recommend that you close this webpage and do not continue to this website.**

Click here to close this webpage.

Continue to this website (not recommended).

More information

Google

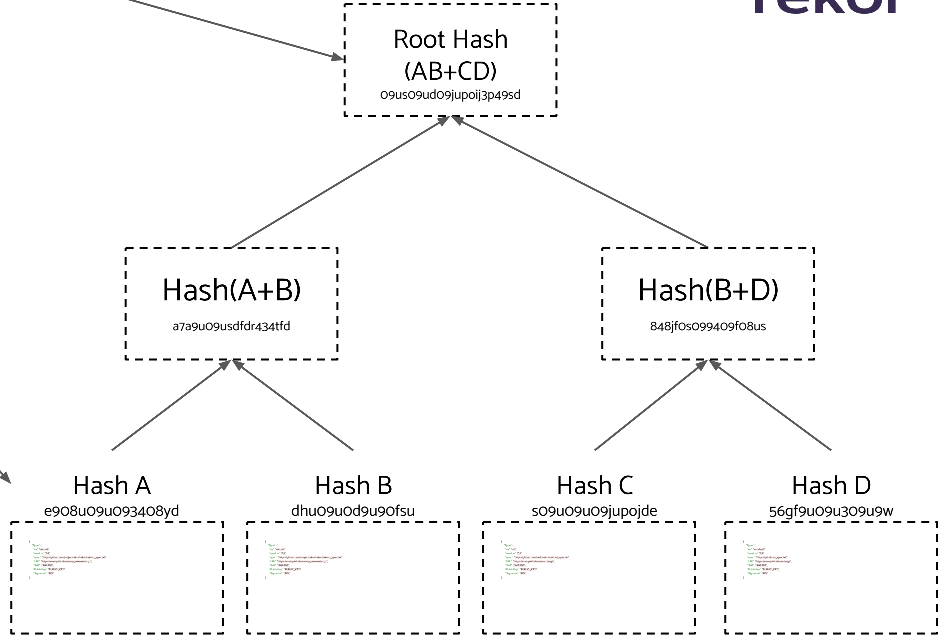
Let's Encrypt

CLOUDFLARE

Could we have Software signing transparency?

# Rekor, append-only, verifiable transparency log

```
{
   "type": "rekord",
   "apiVersion": "0.0.1",
   "spec": {
            "signature": {
                 "format": "pgp",
                 "signature": "sodiui9i9sdkpoklkldd…",
                 "publicKey": {  "url": "-----BEGIN PGP PUBLIC KEY---asdc.."  },
            },
            "data": {
                 "url": "https://example/release/my_release.tar.gz",
                 "hash": { "algorithm": "sha256", "value": "83jfj8we89903uhejw88…"  }
            }
      }
}
o
```



Entry can be validated by "inclusion proof" using signed tree hash)

# Rekor Transparency log

- Transparency log is publicly audible
  - https://rekor.sigstore.dev
- Tamper resistant:
  - Protects against targeted attacks
  - Allows early insight into key compromise
  - Acts as a public ledger to provide non repudiation

For a transparency log to be useful, we need folks to **sign things!**

And they **are not…!**

# Who is signing today. Critical projects?

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

# Who is signing today. Package managers...

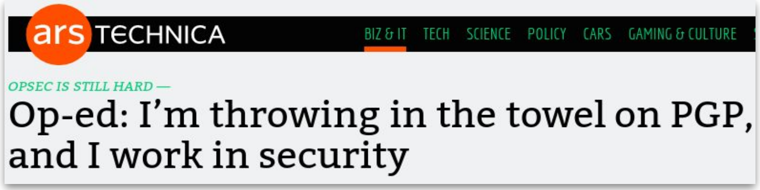
| System | Signatures | Cert Systems | In Use |
|---|---|---|---|
| PyPI | Optional | PGP | Rare |
| NPM | No | No | No |
| Maven Central | Required | PGP/x509 | 100% (keys stored centrally) |
| Containers | Optional | PGP/x509 | Rare |
| Ruby | Optional | x509 | Rare |
| Crates.io (rust) | No | No | No |

# Users are not adopting current signing tools

- Users find singing tools such as PGP cumbersome to use.
- They fear key compromise, need expensive hardware to protect
- Hard to trust keys, challenge to use in CI / ephemeral work loads
- Consensus is... its broken..

MOTHERBOARD
TECH BY VICE

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



OPSEC IS STILL HARD —

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

Schneier on Security

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.

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 existing projects / technologies might be leveraged here?

# OpenID Connect?

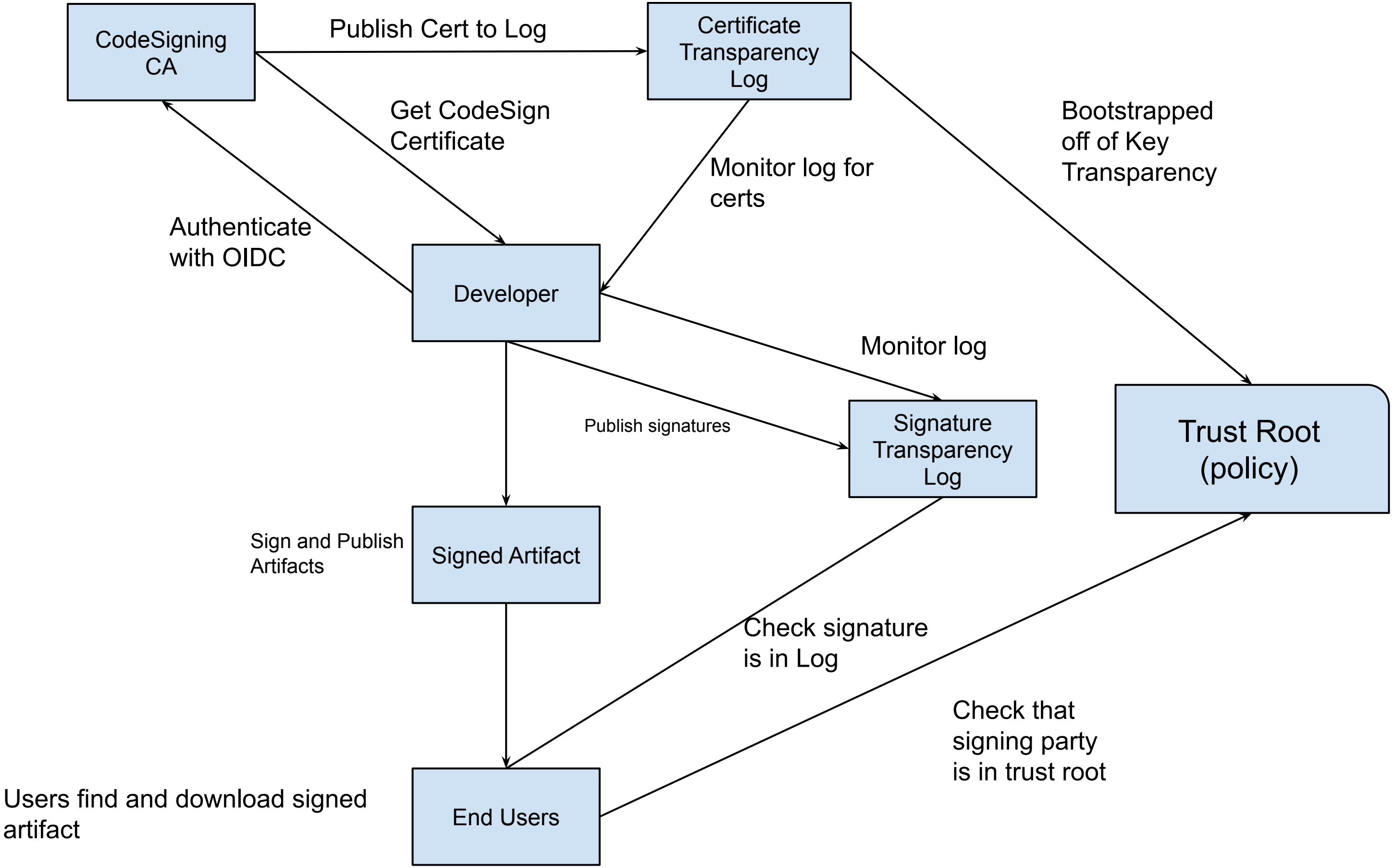
- Users can have a third party attest their identity
- Two Factor Authentication (much easier to use)?

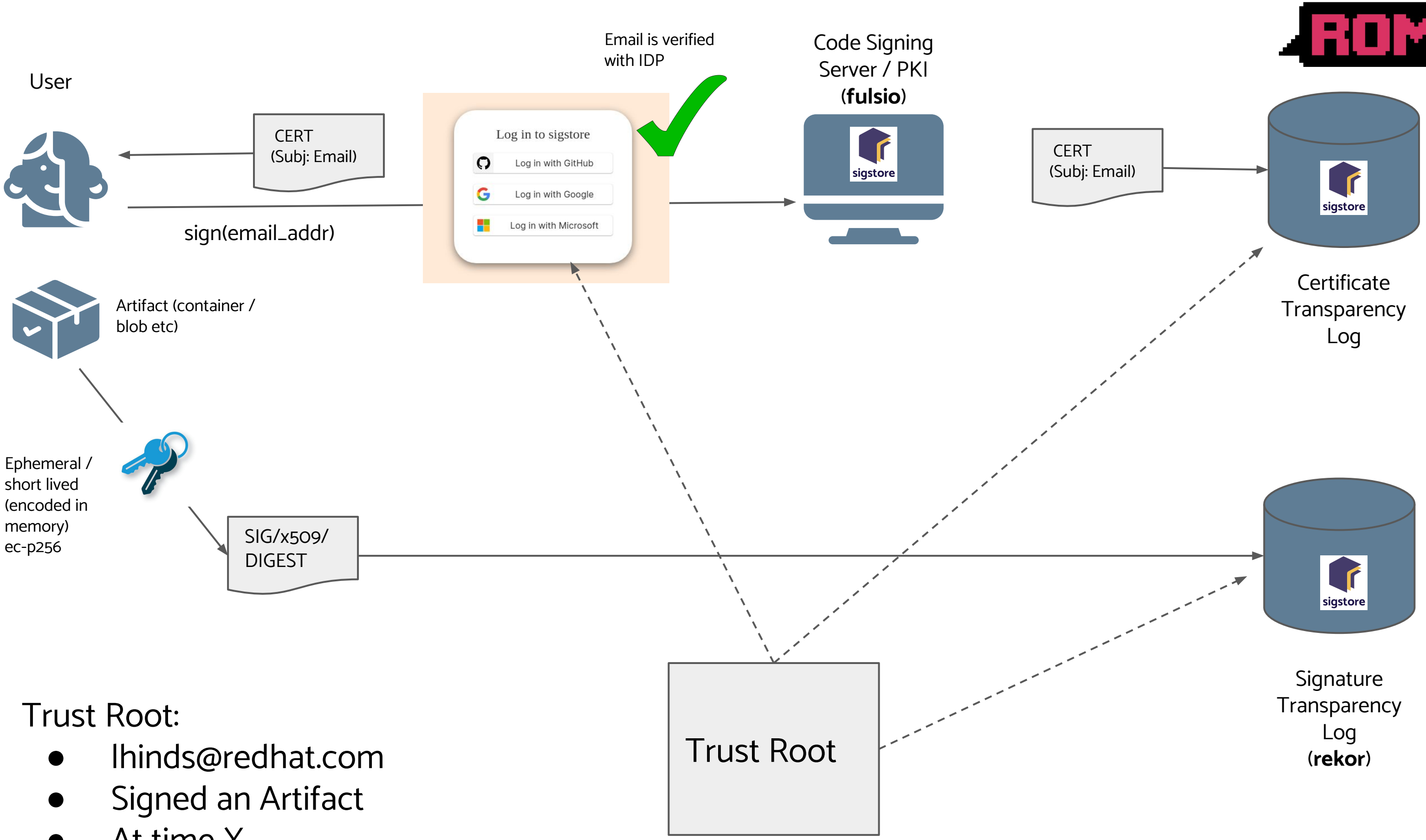
Could someone sign with their OpenID Connect account?

OpenID

Could this be coupled with Transparency logs?

~~rekor~~ / sigstore


CodeSigning CA
Publish Cert to Log
Certificate Transparency Log
Get CodeSign Certificate
Monitor log for certs
Bootstrapped off of Key Transparency
Authenticate with OIDC
Developer
Monitor log
Publish signatures
Signature Transparency Log
Trust Root (policy)
Sign and Publish Artifacts
Signed Artifact
Check signature is in Log
Check that signing party is in trust root
Users find and download signed artifact
End Users



Trust Root:

- lhinds@redhat.com
- Signed an Artifact
- At time X
- With cert chain to sigstore CA

# Build consensus - multiple signers

```
lhinds@redhat.com, dlorenc@gmail.com, bcallaway@redhat.com
between: xx/xx/xxx > xx/xx/xxxx
sha256:c8f9d3ac002cf17d6caeaf315648d9ac5f6c08308bd58a05a028b6e16b4
```

# We do cater to the security geek still..

PGP / minisign / SSH sign / X509

Back end support:

- pkcs11 (e.g Yubikey / HSM)
- Various key management systems (aws, GCP, azure)

# Client tooling - cosign

- Container signing tool
- OCI Registries

# Client tooling - sget (secure get)

- Safe bash script retrieval
- No more `curl https://site.com | sudo bash`

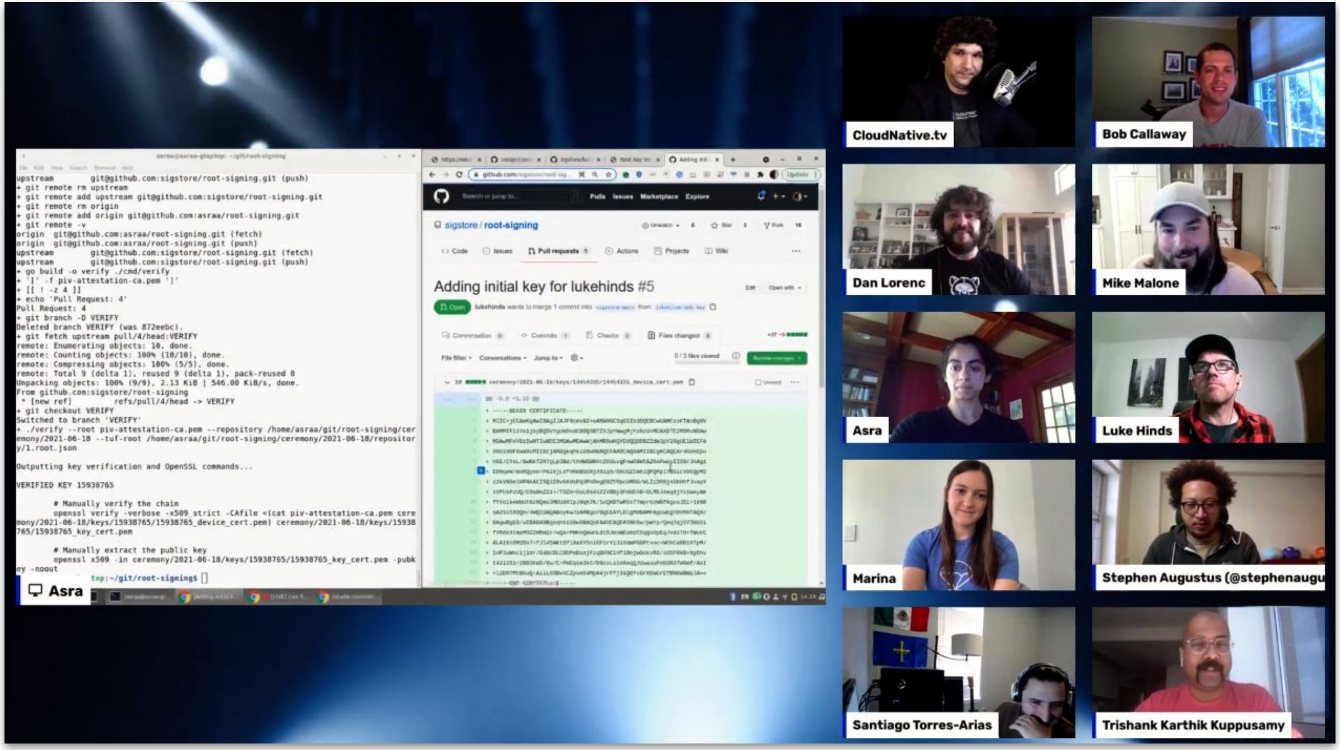
# Client tooling - other clients

- Ruby Gems client
- Commit signing
- Maven Plugin
- Python Implementation in planning / prototype

# Public Service

- Sigstore will be a **non-profit** organization, **free to use** by anyone
- Run under the Linux Foundation
- Code developed in the Open, by a community
    - [https://github.com/sigstore](https://github.com/sigstore)

# Open Source Root CA



```
upstream        git@github.com:sigstore/root-signing.git (push)
+ git remote rm upstream
+ git remote add upstream git@github.com:sigstore/root-signing.git
+ git remote rm origin
+ git remote add origin git@github.com:asraa/root-signing.git
+ git remote -v
origin  git@github.com:asraa/root-signing.git (fetch)
origin  git@github.com:asraa/root-signing.git (push)
upstream        git@github.com:sigstore/root-signing.git (fetch)
upstream        git@github.com:sigstore/root-signing.git (push)
+ go build -o verify ./cmd/verify
+ '[' -f piv-attestation-ca.pem ']'
+ [[ ! -z 4 ]]
+ echo 'Pull Request: 4'
Pull Request: 4
+ git branch -D VERIFY
Deleted branch VERIFY (was 872eebc).
+ git fetch upstream pull/4/head:VERIFY
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0
Unpacking objects: 100% (9/9), 2.13 KiB | 546.00 KiB/s, done.
From github.com:sigstore/root-signing
 * [new ref]         refs/pull/4/head -> VERIFY
+ git checkout VERIFY
Switched to branch 'VERIFY'
+ ./verify --root piv-attestation-ca.pem --repository /home/asraa/git/root-signing/cer
emony/2021-06-18 --tuf-root /home/asraa/git/root-signing/ceremony/2021-06-18/repositor
y/1.root.json

Outputting key verification and OpenSSL commands...

VERIFIED KEY 15938765

        # Manually verify the chain
        openssl verify -verbose -x509_strict -CAfile <(cat piv-attestation-ca.pem cere
mony/2021-06-18/keys/15938765/15938765_device_cert.pem) ceremony/2021-06-18/keys/15938
765/15938765_key_cert.pem

        # Manually extract the public key
        openssl x509 -in ceremony/2021-06-18/keys/15938765/15938765_key_cert.pem -pubk
ey -noout
      top:~/git/root-signing$ 
```

Asra
sigstore / root-signing
Pulls Issues Marketplace Explore
Code Issues Pull requests Actions Projects Wiki
Adding initial key for lukehinds #5
Open
Conversation Commits Checks Files changed
-----BEGIN CERTIFICATE-----
CloudNative.tv
Bob Callaway
Dan Lorenc
Mike Malone
Asra
Luke Hinds
Marina
Stephen Augustus (@stephenaugu
Santiago Torres-Arias
Trishank Karthik Kuppusamy

https://github.com/sigstore/root-signing

# sigstore vision

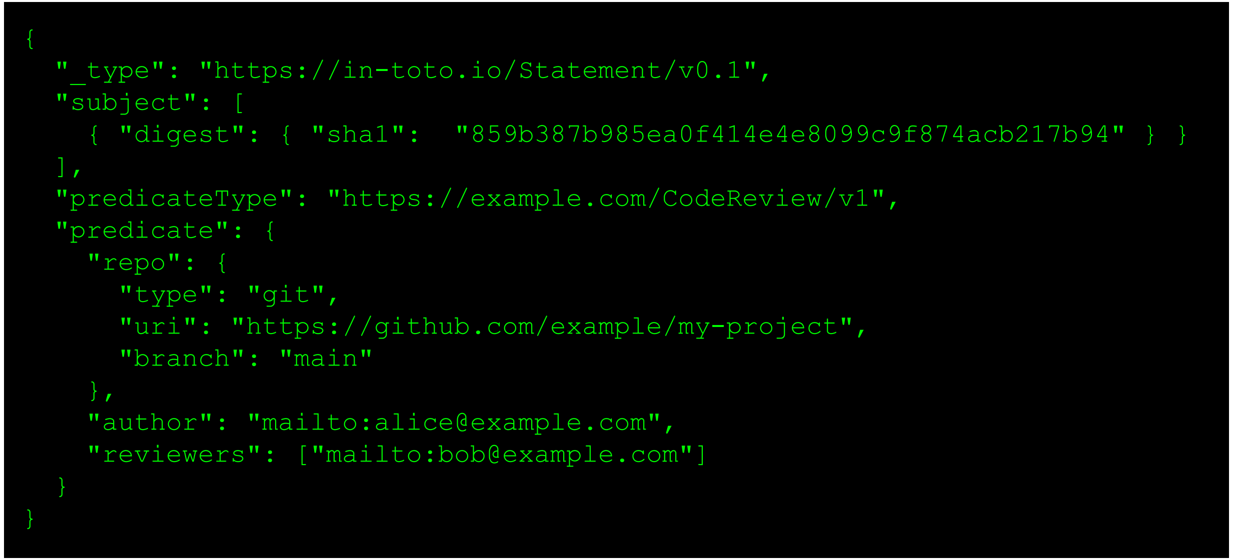
**“To be to software signing and provenance, what Let’s encrypt were to HTTPS”**

# What’s next

# Community on-boarding

fedora
archlinux
kubernetes

# Community collaboration (SBOM)



```
{
  "_type": "https://in-toto.io/Statement/v0.1",
  "subject": [
    { "digest": { "sha1":  "859b387b985ea0f414e4e8099c9f874acb217b94" } }
  ],
  "predicateType": "https://example.com/CodeReview/v1",
  "predicate": {
    "repo": {
      "type": "git",
      "uri": "https://github.com/example/my-project",
      "branch": "main"
    },
    "author": "mailto:alice@example.com",
    "reviewers": ["mailto:bob@example.com"]
  }
}
```

SBOM (Secure Bill of Materials)

# Work with you...

- Integrate systems to audit our public transparency log
- Help you sign things!
- Welcome code contributions , documentation, kick the tyres

# Find out more..

https://sigstore.dev

https://github.com/sigstore