CLOUDNATIVE SECURITYCON NORTH AMERICA 2023

Not all that’s Signed is Secure: Verify the Right Way with TUF and Sigstore

Marina Moore and Zachary Newman

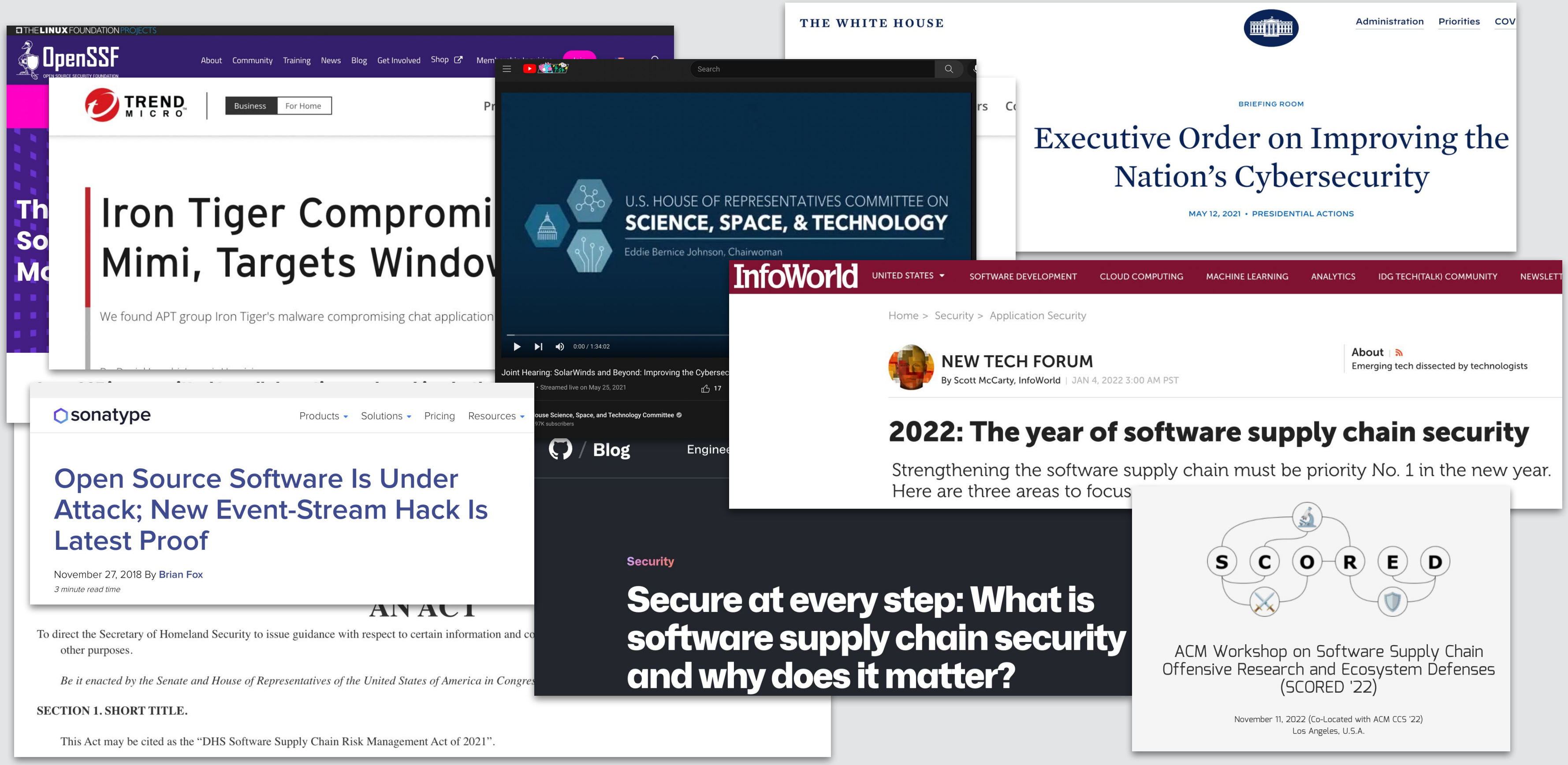
The problem

- Sigstore has more developers signing software
    - So users are more secure, right?
- Signatures only help when **verified correctly**
    - Antipattern: verify software *was* signed, but not *who* signed it

Solution summary

- Enable *flexible*, *smart* policy enforcement
  - *Flexible*: different policies in different settings
  - *Smart*: existing, secure solutions (TUF + in-toto)
- Worked examples:
  - Open source package repositories
  - Internal container registries
  - Everything in between

Software supply chain security


THE LINUX FOUNDATION PROJECTS
OpenSSF
About Community Training News Blog Get Involved Shop
TREND MICRO
Business For Home
THE WHITE HOUSE
Administration Priorities
Search
InfoWorld
UNITED STATES SOFTWARE DEVELOPMENT CLOUD COMPUTING MACHINE LEARNING ANALYTICS IDG TECH(TALK) COMMUNITY
sonatype
Products Solutions Pricing Resources
Blog
Security

Iron Tiger Compromi
Mimi, Targets Windo

We found APT group Iron Tiger's malware compromising chat application

U.S. HOUSE OF REPRESENTATIVES COMMITTEE ON SCIENCE, SPACE, & TECHNOLOGY
Eddie Bernice Johnson, Chairwoman

0:00 / 1:34:02

Joint Hearing: SolarWinds and Beyond: Improving the Cybersec
Streamed live on May 25, 2021

BRIEFING ROOM

Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

Home > Security > Application Security

NEW TECH FORUM
By Scott McCarty, InfoWorld | JAN 4, 2022 3:00 AM PST

About
Emerging tech dissected by technologists

2022: The year of software supply chain security

Strengthening the software supply chain must be priority No. 1 in the new year. Here are three areas to focus

Open Source Software Is Under Attack; New Event-Stream Hack Is Latest Proof

November 27, 2018 By Brian Fox
3 minute read time

AN ACT

To direct the Secretary of Homeland Security to issue guidance with respect to certain information and co
other purposes.

Be it enacted by the Senate and House of Representatives of the United States of America in Congres

SECTION 1. SHORT TITLE.

This Act may be cited as the "DHS Software Supply Chain Risk Management Act of 2021".

Security

Secure at every step: What is software supply chain security and why does it matter?

S C O R E D

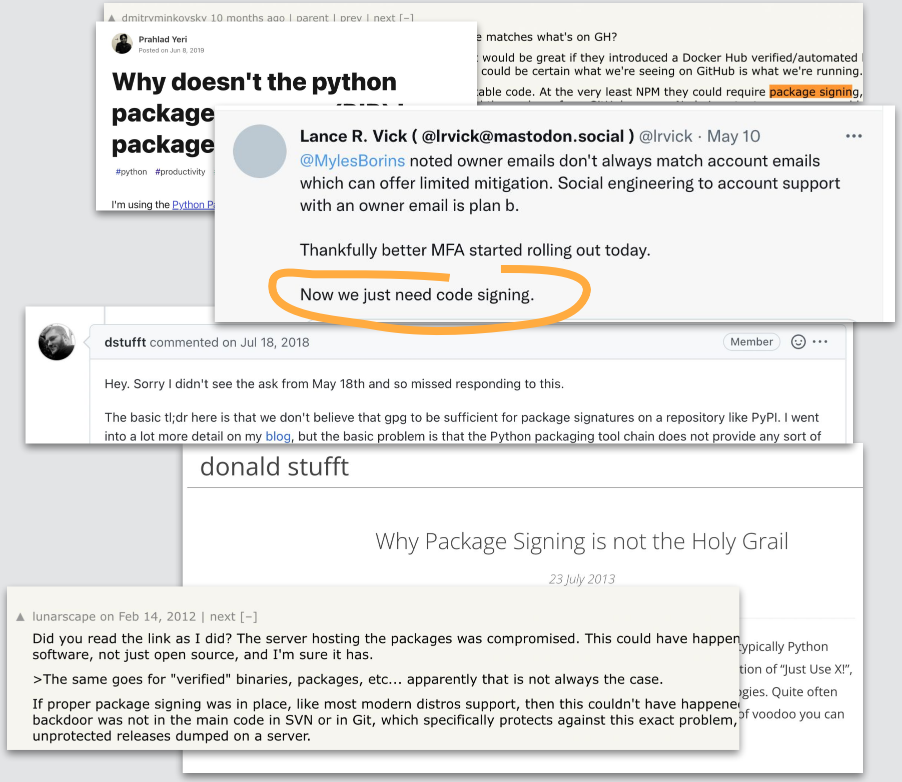
ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses (SCORED '22)

November 11, 2022 (Co-Located with ACM CCS '22)
Los Angeles, U.S.A.

Why sign software?

Part of the solution.

- You download software from the *right* place, but it's not what the owner intended
  - Compromised account
  - Compromised build process
  - Compromised package repository



▲ dmitryminkovsky 10 months ago | parent | prev | next [–]

e matches what's on GH?

would be great if they introduced a Docker Hub verified/automated
could be certain what we're seeing on GitHub is what we're running.

able code. At the very least NPM they could require package signing,

Prahlad Yeri
Posted on Jun 8, 2019

Why doesn't the python package
package
package

#python #productivity

I'm using the Python Pa

Lance R. Vick (@lrvick@mastodon.social) @lrvick · May 10

@MylesBorins noted owner emails don't always match account emails which can offer limited mitigation. Social engineering to account support with an owner email is plan b.

Thankfully better MFA started rolling out today.

Now we just need code signing.

dstufft commented on Jul 18, 2018 Member

Hey. Sorry I didn't see the ask from May 18th and so missed responding to this.

The basic tl;dr here is that we don't believe that gpg to be sufficient for package signatures on a repository like PyPI. I went into a lot more detail on my blog, but the basic problem is that the Python packaging tool chain does not provide any sort of

donald stufft

Why Package Signing is not the Holy Grail

23 July 2013

ypically Python
tion of "Just Use X!",
gies. Quite often
of voodoo you can

▲ lunarscape on Feb 14, 2012 | next [–]

Did you read the link as I did? The server hosting the packages was compromised. This could have happen
software, not just open source, and I'm sure it has.

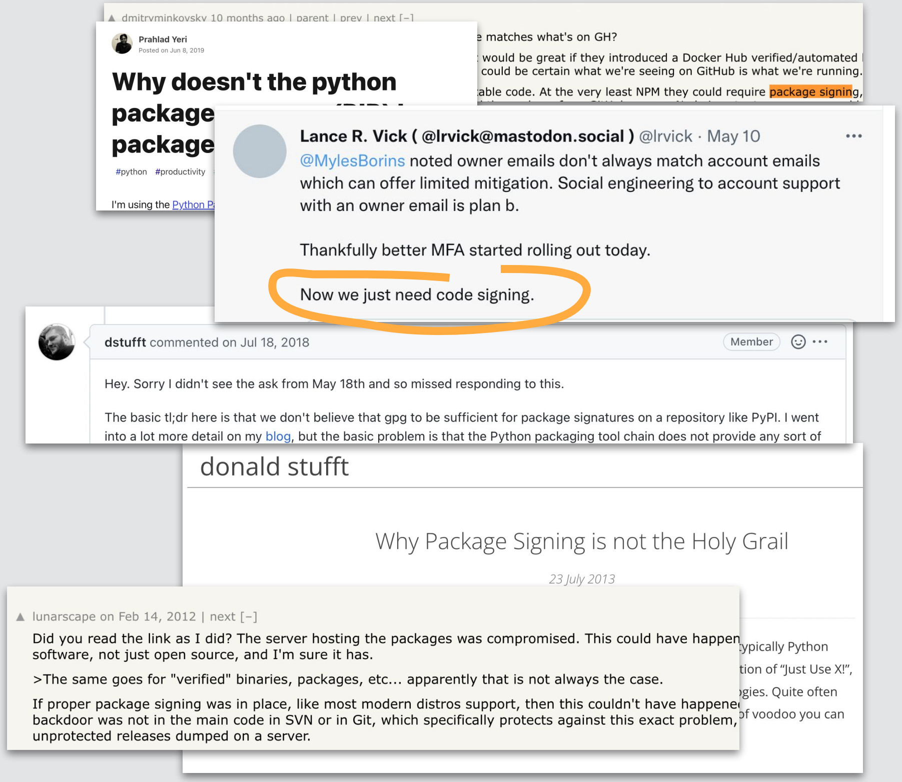
>The same goes for "verified" binaries, packages, etc... apparently that is not always the case.

If proper package signing was in place, like most modern distros support, then this couldn't have happene
backdoor was not in the main code in SVN or in Git, which specifically protects against this exact problem,
unprotected releases dumped on a server.

Why sign software?

Part of the solution.

- Not all attacks!
  - Normal vulnerabilities
  - Underhanded PRs
  - Blackmailing authors
- *If* you know who's supposed to sign a package, signing helps.
  - Big "if;" will revisit later



▲ dmitryminkovsky 10 months ago | parent | prev | next [–]

e matches what's on GH?

would be great if they introduced a Docker Hub verified/automated
could be certain what we're seeing on GitHub is what we're running.

able code. At the very least NPM they could require package signing,

Prahlad Yeri
Posted on Jun 8, 2019

Why doesn't the python package
package

#python #productivity

I'm using the Python P

Lance R. Vick (@lrvick@mastodon.social) @lrvick · May 10

@MylesBorins noted owner emails don't always match account emails which can offer limited mitigation. Social engineering to account support with an owner email is plan b.

Thankfully better MFA started rolling out today.

Now we just need code signing.

dstufft commented on Jul 18, 2018 · Member

Hey. Sorry I didn't see the ask from May 18th and so missed responding to this.

The basic tl;dr here is that we don't believe that gpg to be sufficient for package signatures on a repository like PyPI. I went into a lot more detail on my blog, but the basic problem is that the Python packaging tool chain does not provide any sort of

donald stufft

Why Package Signing is not the Holy Grail

23 July 2013

typically Python
tion of "Just Use X!",
gies. Quite often
of voodoo you can

▲ lunarscape on Feb 14, 2012 | next [–]

Did you read the link as I did? The server hosting the packages was compromised. This could have happen
software, not just open source, and I'm sure it has.

>The same goes for "verified" binaries, packages, etc... apparently that is not always the case.

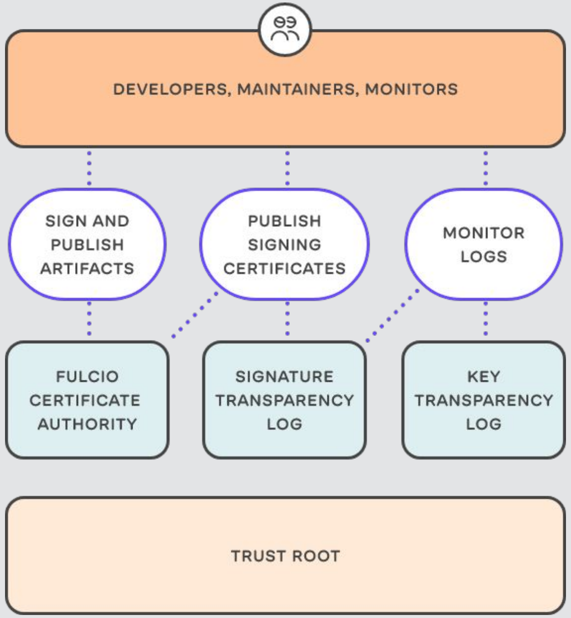
If proper package signing was in place, like most modern distros support, then this couldn't have happene
backdoor was not in the main code in SVN or in Git, which specifically protects against this exact problem,
unprotected releases dumped on a server.

Sigstore

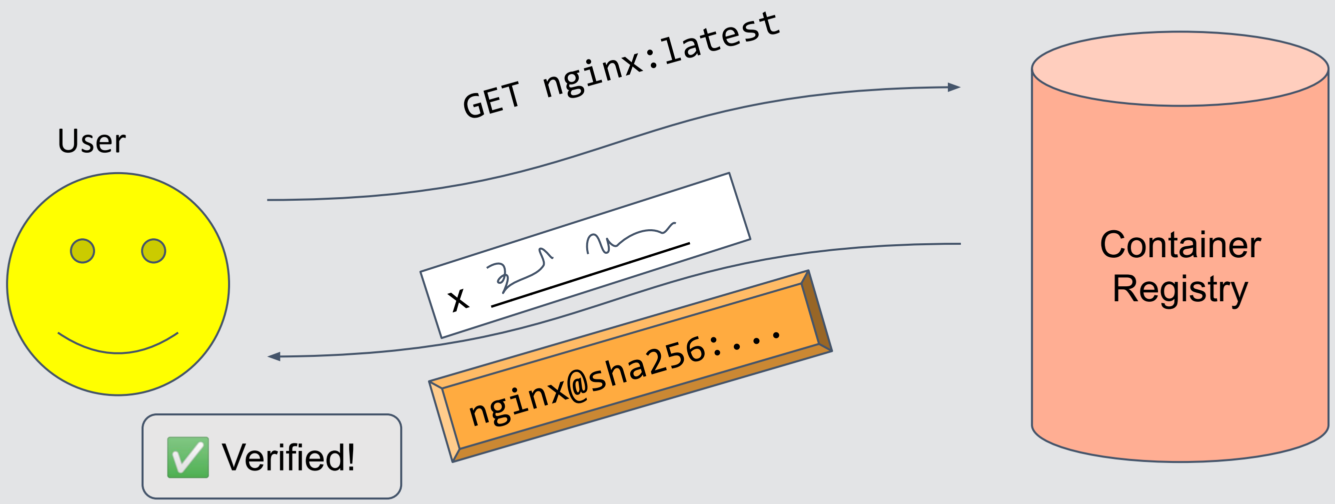
- Easy signing for containers and more
- No key management:
  - Sign with SSO
  - Sign with machine identity
- Transparency: detect misbehavior

Sigstore

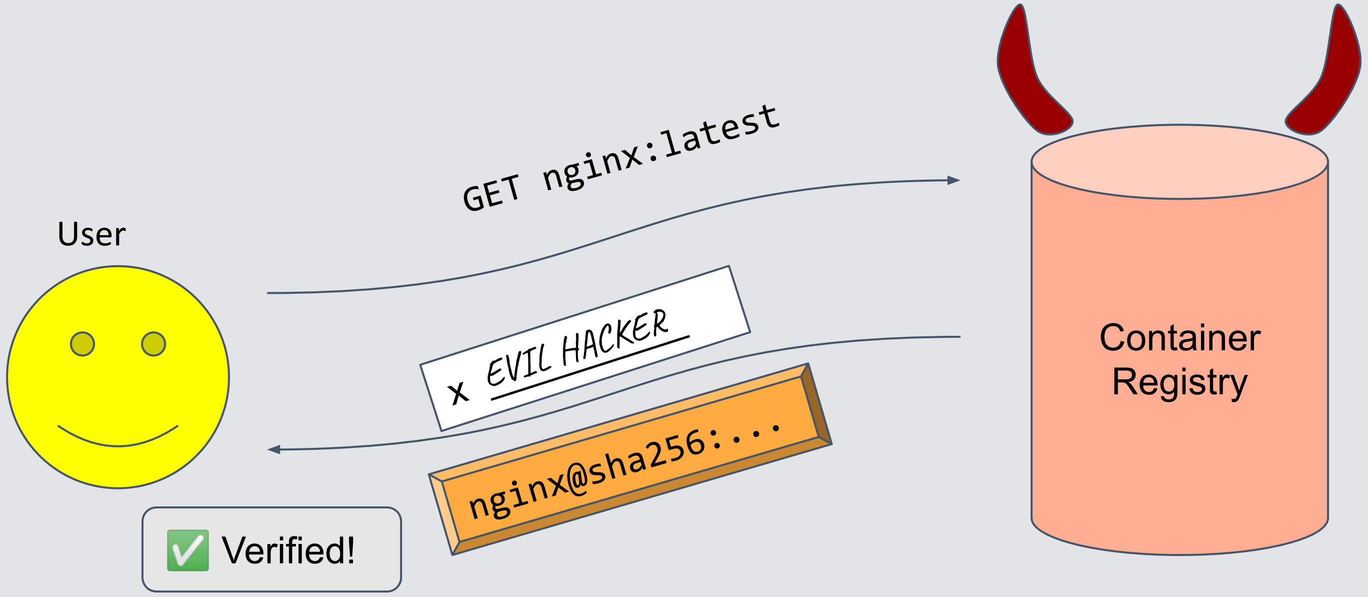
- Fulcio (CA): issues short-lived certificates for OIDC credentials ("login with Facebook")
- Rekor (log): timestamps signatures, record metadata
- Cosign: stick signatures in OCI registries


DEVELOPERS, MAINTAINERS, MONITORS
SIGN AND PUBLISH ARTIFACTS
PUBLISH SIGNING CERTIFICATES
MONITOR LOGS
FULCIO CERTIFICATE AUTHORITY
SIGNATURE TRANSPARENCY LOG
KEY TRANSPARENCY LOG
TRUST ROOT

Verification Policies



Verification Policies



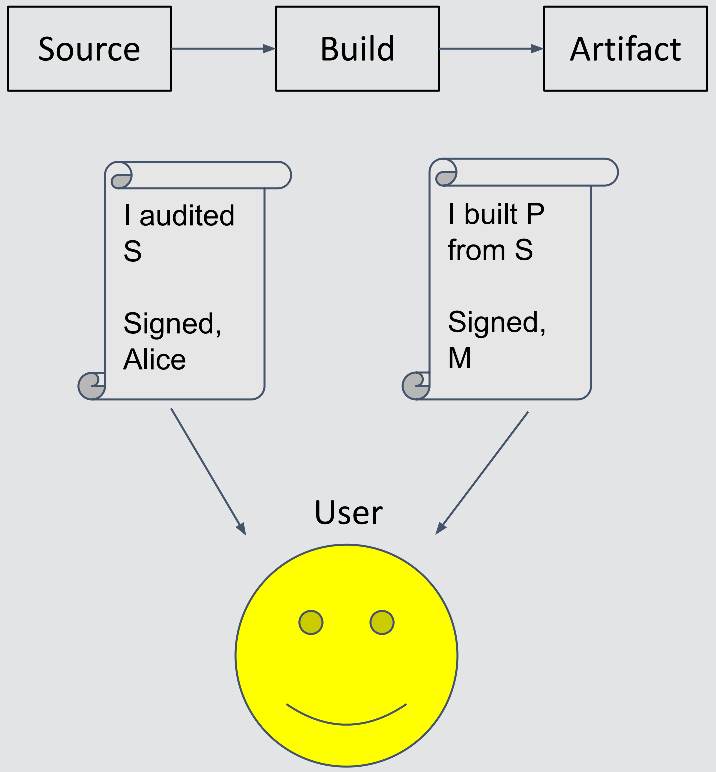
Verification Policies

- Verification policies help us **interpret signatures**.
  - What do I *mean* when I sign something?
  - Did I look at every byte in the binary?
  - We can attach *specific* meanings to signatures (claims)

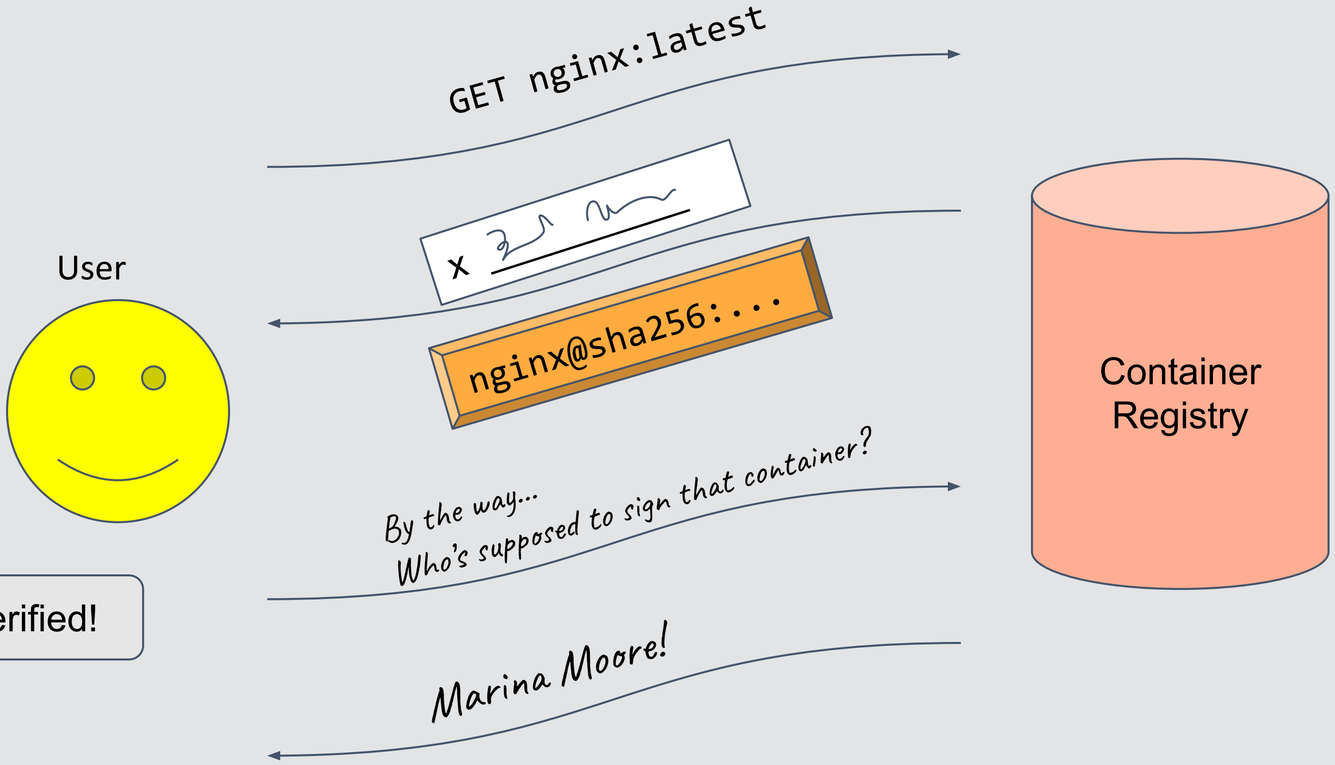
I claim <X>
Signed,
<Party>

Verification Policies

- Simple: universal signer.
  - Signature == “this binary is good”
- Ownership: package P came from Alice
- Build integrity: machine M built this artifact
- Combination: BOTH
  - Machine M: “I built package P from source code S”
  - AND Alice: “I audited S”

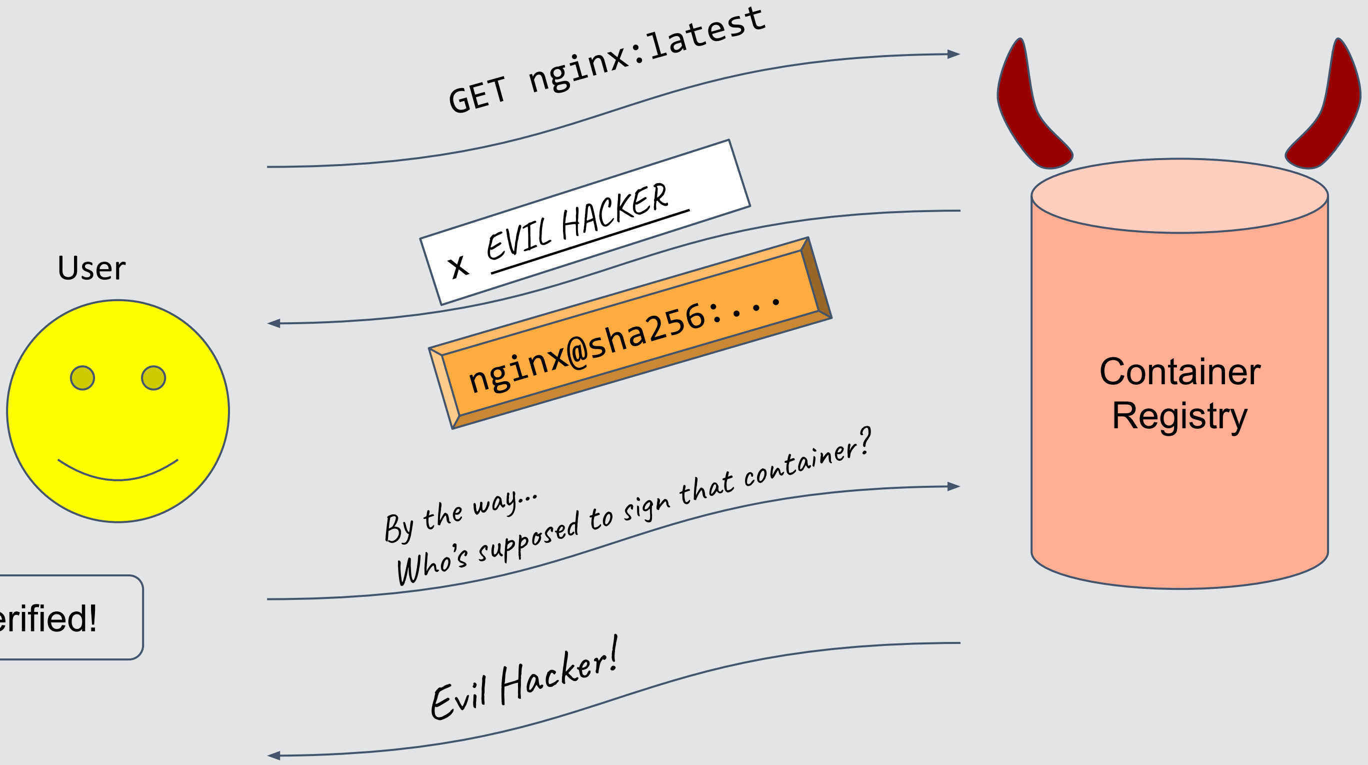


Getting a Policy Securely



✅ Verified!

Getting a Policy Securely



✅ Verified!

Solution: TUF and in-toto

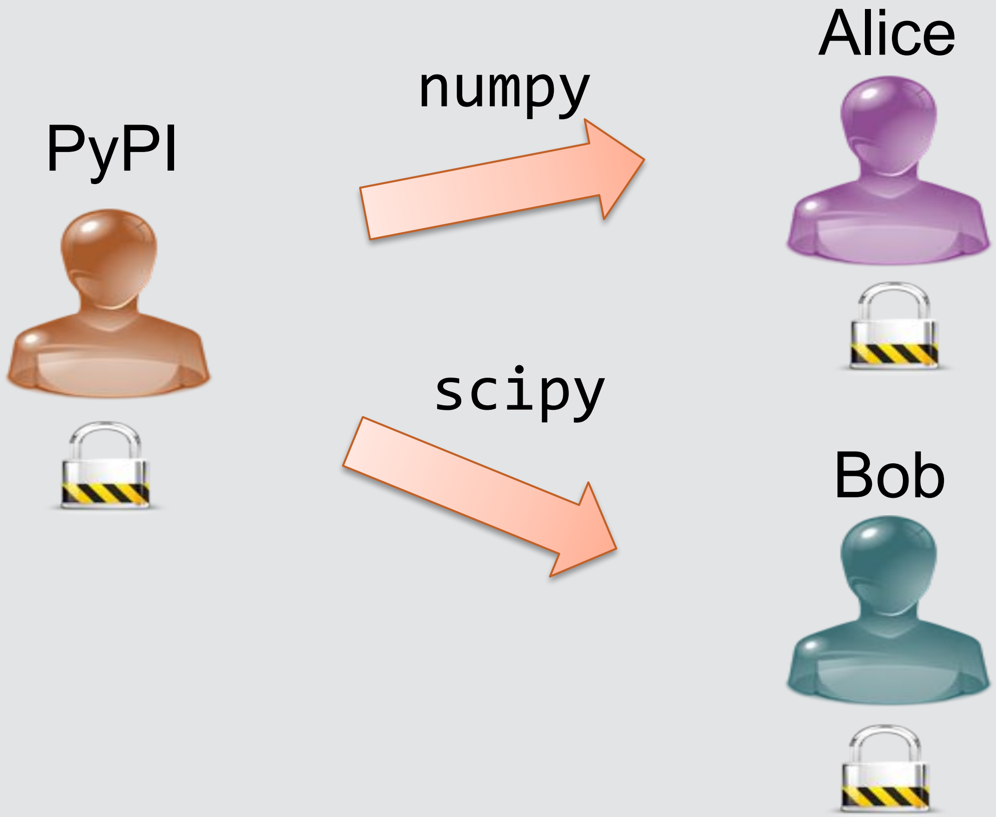
CLOUDNATIVE SECURITYCON NORTH AMERICA 2023

- You have to know what you’re running. There’s a *context* for software.
- The Update Framework (TUF) does secure distribution
  - WHO uploaded, WHAT did they upload, WHY you trust them
  - Compromise resilient: secure even when a repository or signer is compromised
- In-toto does “combinations”
  - Beyond distribution: who built, tested, etc.

TUF

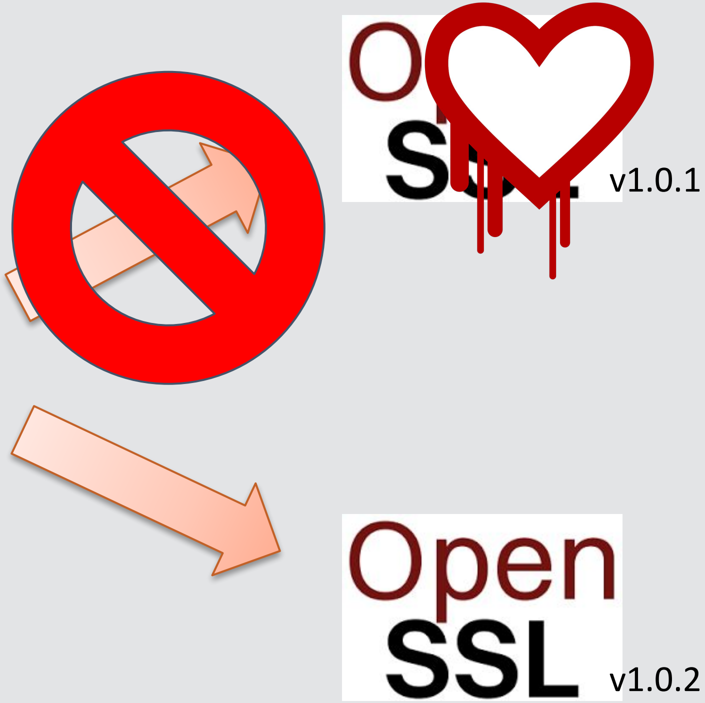
- CNCF Graduated Project
  - Based on peer-review academic research
  - Used by Fuschia, Datadog, automotive industry, ...
- TUF principles:
  - Separation of responsibilities
    - Minimize consequence of any one compromise
  - Multi-signature trust
  - Explicit and implicit revocation
  - Secure recovery from a compromise
- Full talk: TUF-en Up Your Signatures (KubeCon NA 2022)

TUF: Delegations



TUF: Explicit Revocation

- Respond to new information:
  - Vulnerabilities
  - New versions
- Timeliness: client never gets revoked/out-of-date packages

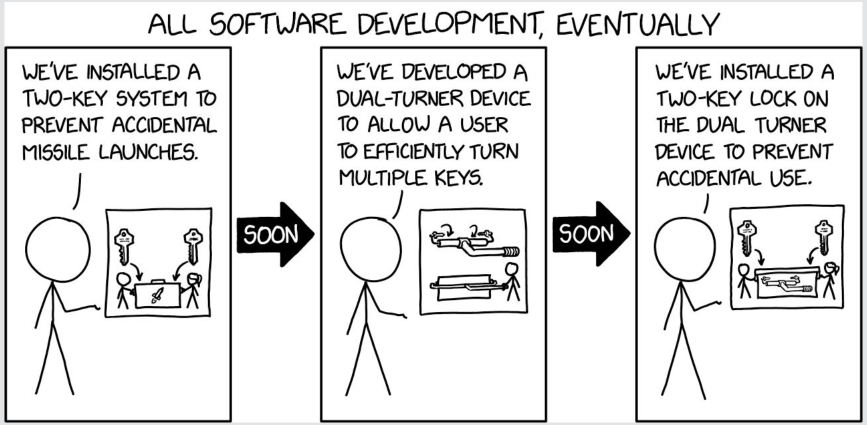


TUF Implicit Revocation

- All keys expire
- Helps with undetected compromises
    - Ensure all keys are current

TUF signature thresholds

- Require multiple signatures for the *same* package
- Developer team AND security team signed a package


ALL SOFTWARE DEVELOPMENT, EVENTUALLY
WE'VE INSTALLED A TWO-KEY SYSTEM TO PREVENT ACCIDENTAL MISSILE LAUNCHES.
SOON
WE'VE DEVELOPED A DUAL-TURNER DEVICE TO ALLOW A USER TO EFFICIENTLY TURN MULTIPLE KEYS.
SOON
WE'VE INSTALLED A TWO-KEY LOCK ON THE DUAL TURNER DEVICE TO PREVENT ACCIDENTAL USE.

Undetected key compromise

- Remaining issues with using TUF:
  - Detecting when your key is used by an attacker
  - Are you seeing the same signatures as everyone else?
- We also need *auditability*

TUF + Sigstore

CLOUDNATIVE SECURITYCON NORTH AMERICA 2023

- We get auditability with Sigstore!
- Use Sigstore’s transparency with TUF for:
    - User auditing of key usage
    - Global consistency

TUF/Sigstore Internal Containers

- Store signatures + TUF metadata in OCI
- Fixed policy:
    - Dev team must sign every image (using SSO)
    - Image built by GitHub Actions (using workload ID)
- For free: revocation, key rotation, freshness
- Enforced by Kubernetes admission controller

TUF/Sigstore Package Repository

CLOUDNATIVE SECURITYCON NORTH AMERICA 2023

- Delegate to every uploader to the repository
- Uploaders can use key pair *or* Sigstore identity
- Default policy: All packages signed by correct uploader
- For paranoid users: allowlist trusted uploaders
- Enforced by package manager
- For free: revocation, key rotation, freshness, protection from repository compromise

TUF + Sigstore other uses

- App store
  - Trusted developers
- Curated package repository
  - Additional signatures from analysis/security teams
- Single product updater
  - Mitigate compromise of distribution server (Mimi)

Details / learn more / future work

- Revocation
- Scalability
- In-band key rotation
- Post quantum
- Source signing

- Simplifying setup of TUF repositories
  - Federation
  - Share TUF roots
- in-toto

Get involved

- TUF
  - Specification: theupdateframework.github.io/specification/latest/
  - python-tuf: github.com/theupdateframework/python-tuf
  - go-tuf: github.com/theupdateframework/go-tuf
  - rust-tuf: github.com/theupdateframework/rust-tuf
  - CNCF slack
- Sigstore
  - Github: github.com/sigstore
  - Home page: sigstore.dev/
  - Sigstore Slack

CLOUDNATIVE SECURITYCON NORTH AMERICA 2023

Please scan the QR Code above
to leave feedback on this session