Chainguard.

Supply Chain Security

Dan Lorenc
dlorenc@chainguard.dev

June 8th 2022

Agenda

The Past, Present, and Future of Supply Chain Security

- History of Attacks
  - 1984: The Beginning
  - 1984-2015: The Dark Ages
  - 2015: The Explosion
- Types of Attacks
- Techniques to Mitigate
  - Signatures and Attestations
  - PKI
  - SLSA
  - Reproducible Builds

A Brief History Of Supply Chain Attacks

1984: Reflections on Trusting Trust

TURING AWARD LECTURE

Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

KEN THOMPSON

Part One: Backdoor a Compiler!

1. Add a backdoor to a compiler's source code
   a. The backdoor will cause the compiler to add code to each compiled binary to print the user's password
2. Compile the compiler
3. Hand it to your friends
4. Profit!

Part Two: Backdoor ALL Compilers!

1. Add a **propagating** backdoor to a compiler's source code
   a. The backdoor will cause the compiler to add code to each compiled binary to print the user's password
   b. When compiling another compiler, add the same backdoor
2. Compile the compiler
3. Hand it to your friends
4. **Delete the backdoor source code**
5. Profit, **Over and Over Again!**

Part Three: Leave no Trace!

1. Add a propagating backdoor to a compiler's source code
   a. The backdoor will cause the compiler to add code to each compiled binary to print the user's password
   b. When compiling another compiler, add the same backdoor
   c. When compiling a disassembler, add a backdoor to hide all the other backdoors
2. Compile the compiler
3. Hand it to your friends
4. Delete the backdoor source code
5. Profit, Over and Over Again, **and laugh as your friends are baffled**

The 80s- Late 20teens: The Dark Ages

Gentoo Archives: gentoo-announce

From:	Daniel Robbins <drobbins@g.o>
To:	gentoo-announce@g.o, bugtraq@xxxxxxxxxxxxx.com
Subject:	**[gentoo-announce] GLSA: rsync.gentoo.org rotation server compromised (200312-01)**
Date:	Wed, 03 Dec 2003 12:07:56
Message-Id:	1070474017.15117.101.camel@ht.gentoo.org

Bloke accused of Linux kernel.org hack nabbed during traffic stop

Possible 40 years in the Big House for 2011 infiltration of open-source world's servers

Infrastructure report, 2008-08-22 UTC 1200

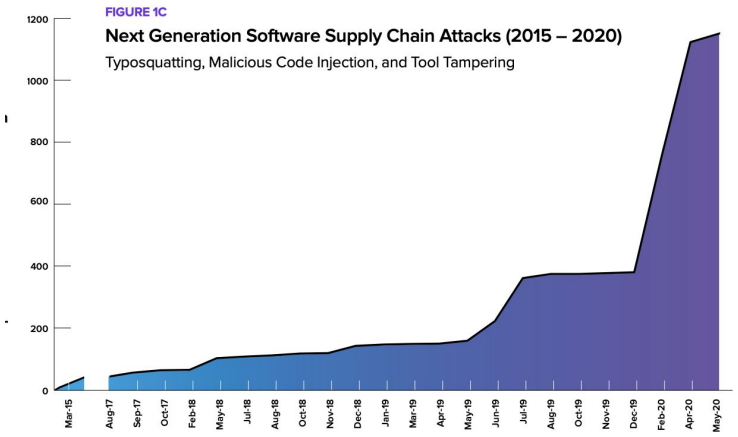
Paul W. Frields stickster at gmail.com
Fri Aug 22 12:00:02 UTC 2008

- Previous message (by thread): Infrastructure status, 2008-08-19 UTC 0200
- Next message (by thread): Fedora Unity releases Fedora 8 Re-Spin
- **Messages sorted by:** [date] [thread] [subject] [author]

```
Last week we discovered that some Fedora servers were illegally
accessed. The intrusion into the servers was quickly discovered, and the
servers were taken offline.
```

2015-Today: The Rise!

- Attacks are on the rise across every stage of the development lifecycle
- 650% increase in 2022 (Sonatype)
- **$100 billion** in damages from Solarwinds attack alone (Senate report)
- **45%** of organizations will experience an attack by 2024 (Gartner) **a 3X increase from 2021.**



Why Now?

We've gotten good enough at basic infrastructure security that supply chain attacks are now the easiest path for threat actors.

Types of Attacks Today

Next Generation Supply Chain Attacks

The Software Supply Chain “Problem” is Many Problems

Integrity

Quality

Malicious Maintainers

The Software Supply Chain “Problem” is Many Problems

Integrity

- Solarwinds/Sunburst
- Codecov
- Wordpress Theme attack
- Linux Kernel server compromise

Quality

- Heartbleed
- Log4Shell
- Shellshock
- Psychic Signatures

Malicious Maintainers

- Leftpad
- Protestware
- Hypocrite commits
- Typosquatting
- Coin mining

Mitigation Techniques

The Quality Problem

All code has bugs. Whether it's yours or someone else's.

Sometimes these bugs cause security problems…

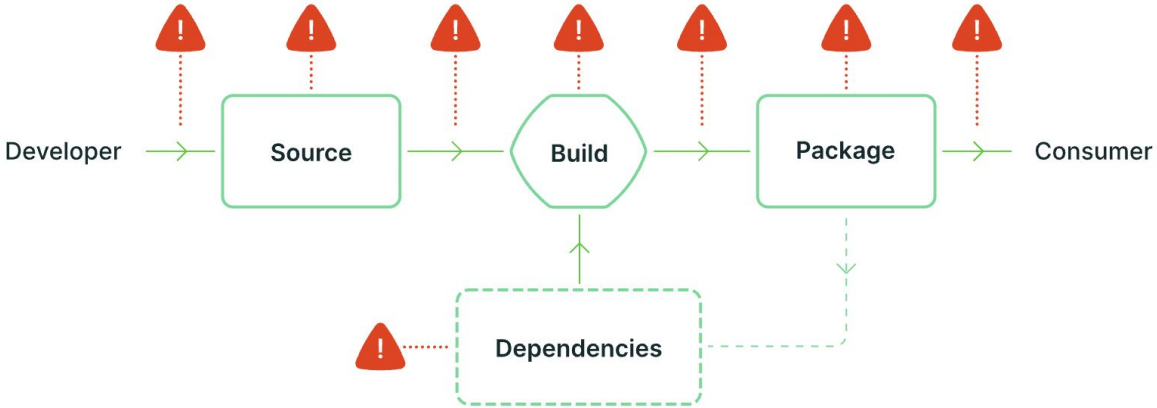
- Apply more care in what you use!
- Basic hygiene goes a long way:
  - Fuzzing
  - Testing (SAST/DAST)
  - **Regular updates to pickup fixes**
- Inventory Management!
  - SBOMs are a key component here.
- Pin dependencies for visibility and patch/update them regularly!

The Integrity Problem

Build systems, artifact managers, source repositories, and distribution systems are all attack points.

And they've been neglected traditionally.

- **Treat build systems like production systems.**
- Use responsible package installation techniques (no curl | bash where avoidable).
- Ask your vendors to do so as well.
- The SLSA Framework helps map threats to mitigations.

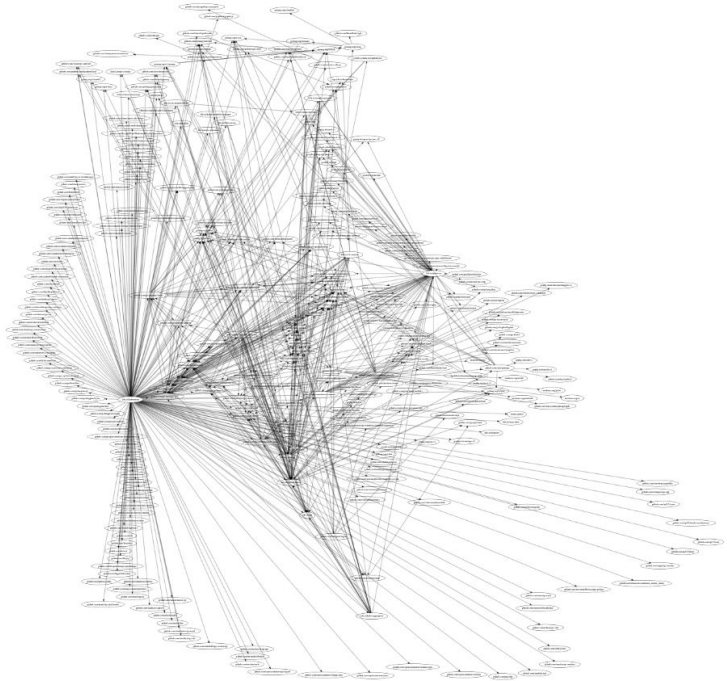


Malicious Developers

Open Source is great because anyone on the internet can contribute!

Until you remember that not everyone on the internet is a nice person…

- Also applies to innersource/insider-risk models.
- Can’t always stop this, but you can require signatures and non-repudiation to limit blast radius.

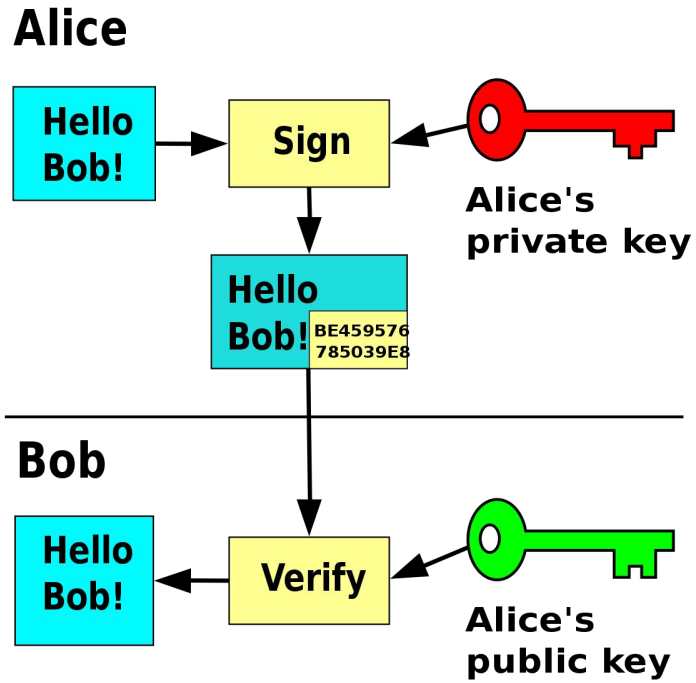


Deep Dives

Signatures/Attestations, PKI, Artifact Integrity

What is a Signature?

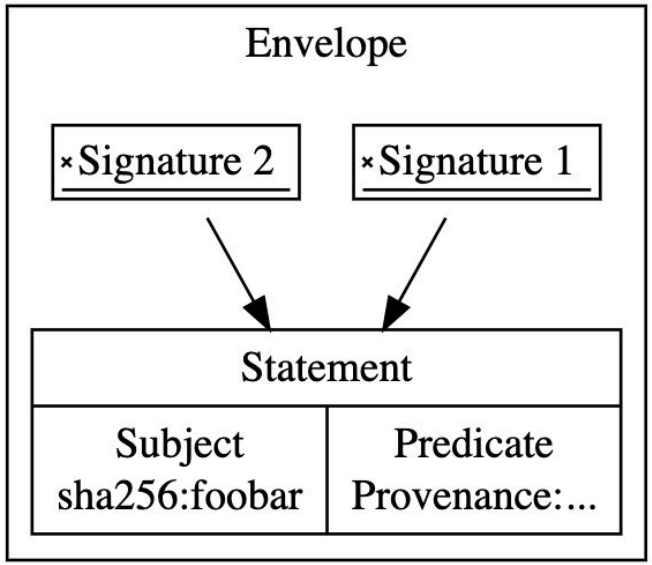
Signatures are a tool and a primitive, **not** an answer or a solution!



What is an Attestation?

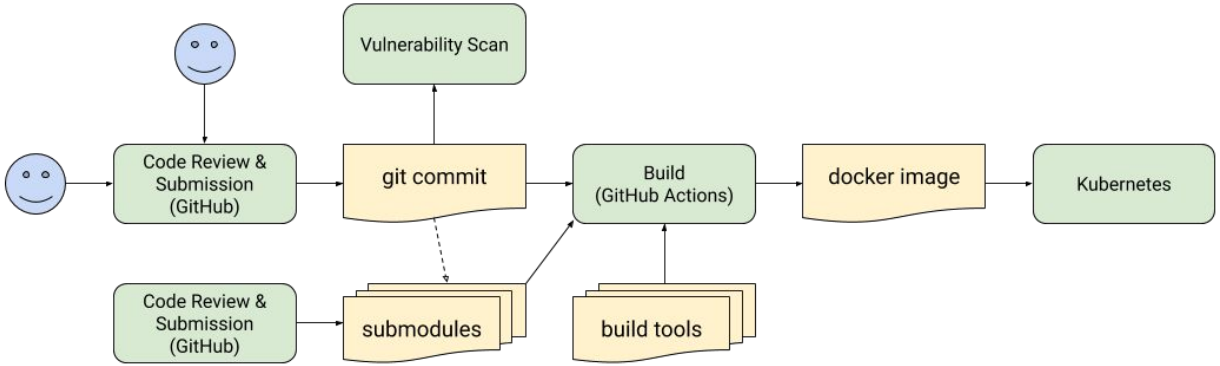
An Attestation is a Signature + Some Context

- **Envelope:** Handles authentication and serialization.
- **Statement:** Binds the attestation to a particular subject and unambiguously identifies the types of the predicate.
- **Predicate:** Contains arbitrary metadata about the subject, with a type-specific schema.
- **Bundle:** Defines a method of grouping multiple attestations together.

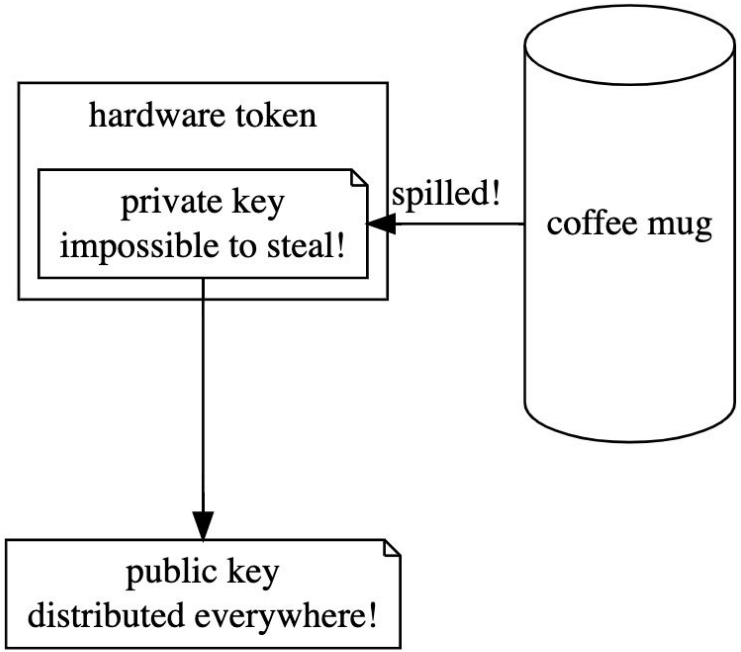


Attestation Data Types

- Provenance (SLSA)
- Vulnerability Scan
- Code Review
- SBOMs, more!

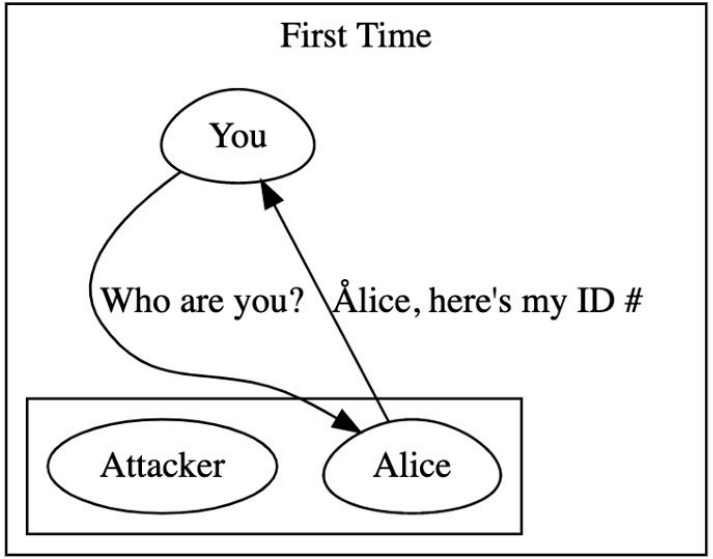


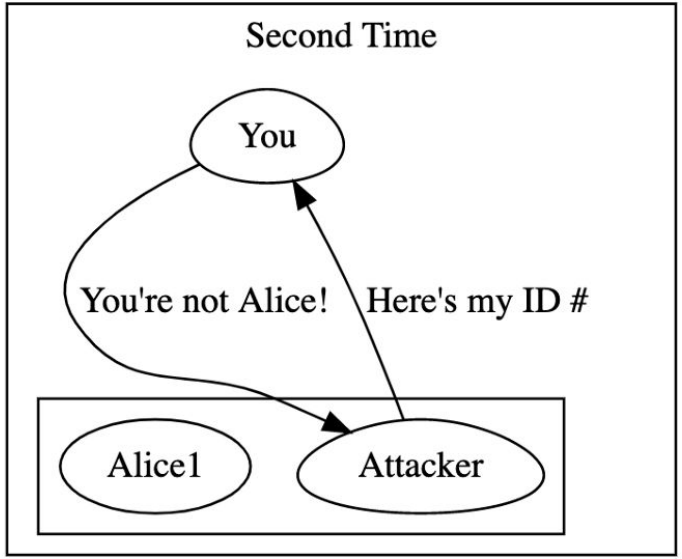
What is PKI?



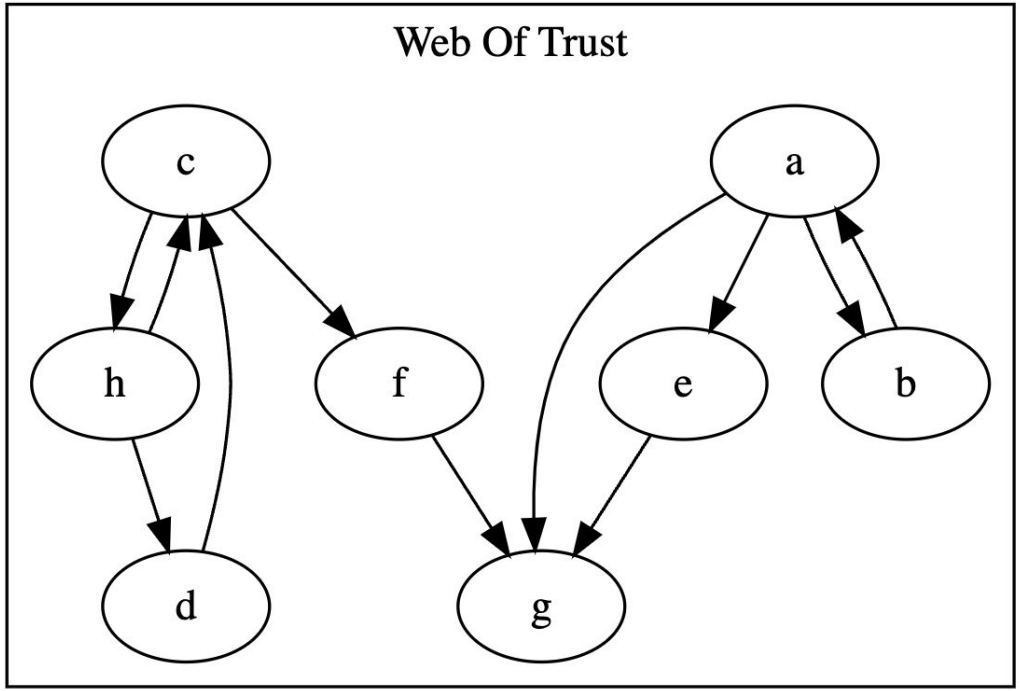
Now what?

PKI Deep Dive - TOFU

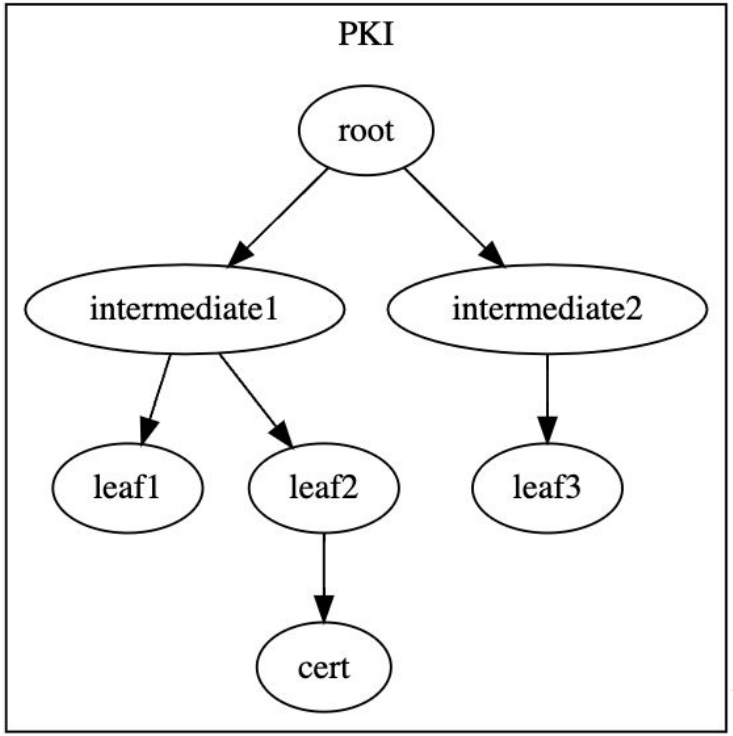




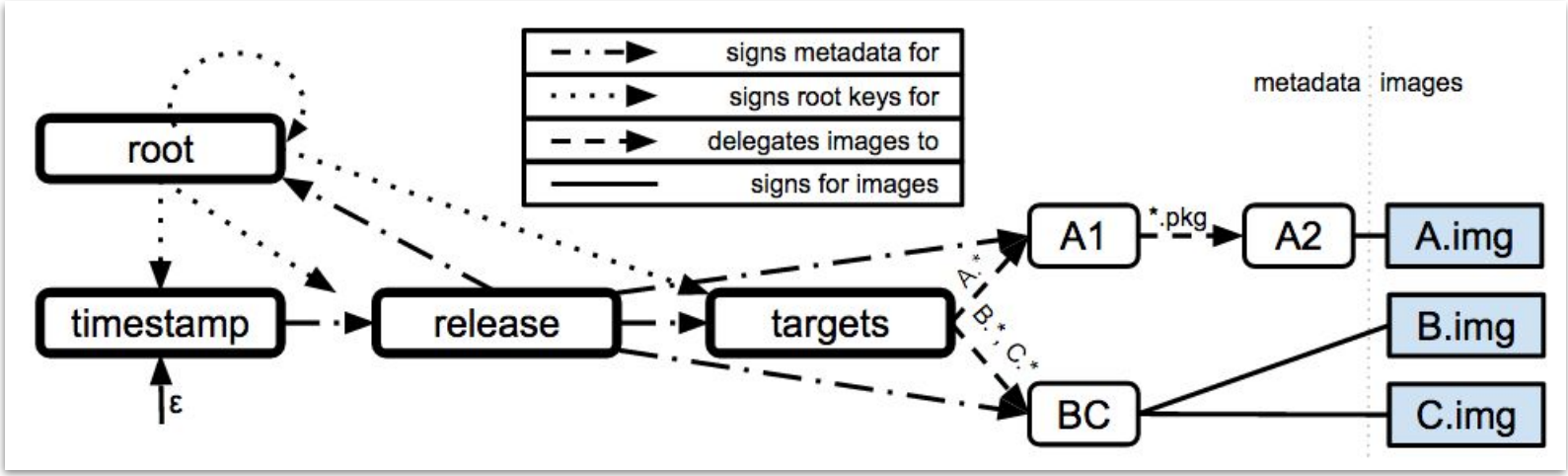
PKI Deep Dive - Web of Trust



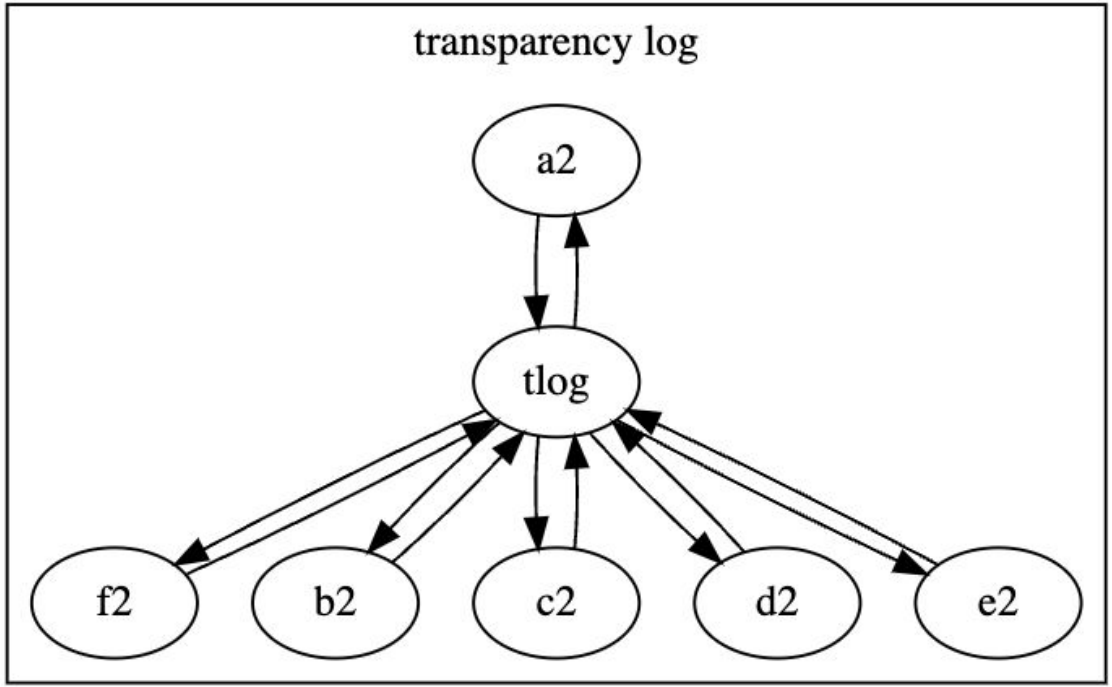
PKI Deep Dive - Traditional x509



PKI Deep Dive - TUF



PKI Deep Dive - Transparency!

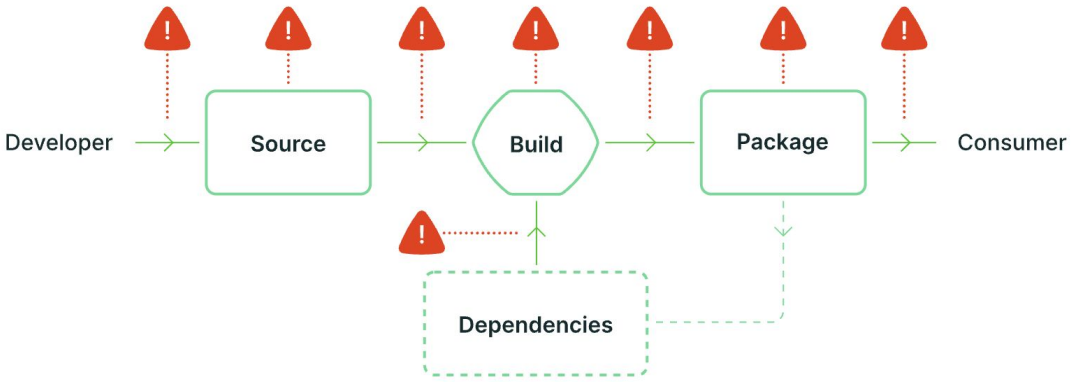


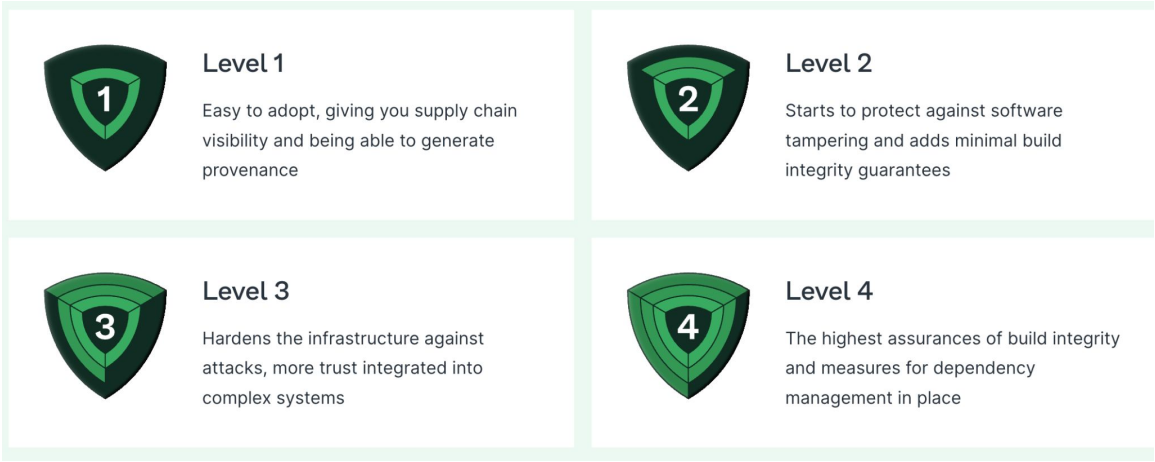
SLSA/SSDF

An artifact/signature is only as secure as the infrastructure that produced it*.

SLSA and the SSDF are frameworks for mapping threats to controls.

TREAT BUILD SYSTEMS LIKE PRODUCTION SYSTEMS





Reproducible Builds

A shortcut to build integrity?

- **Same inputs == same outputs**
- Can be used to detect backdoors
- Diverse Double Compiling can even solve Trusting Trust!
- Complex in practice, and requires someone to actually check.
- Two lower trusted build systems can still be simpler than one highly secured one in some cases.

Questions?