

SLSA Tech Talk

Securing the Software Supply Chain:
An In-Depth Exploration of SLSA

October 5, 2023





Welcome!

- Thank you for joining us today! We will begin at 9:02am PT.
- While we wait for everyone to join, please take a moment to do one (or more) of the following:
 - Please add questions using the Zoom Q&A feature
 - Follow us on Twitter: [@openssf](https://twitter.com/openssf), Mastodon: social.lfx.dev/@openssf, & LinkedIn: [OpenSSF](https://www.linkedin.com/company/openssf)
 - Visit our website: <https://openssf.org>
 - Sign up for training: <https://openssf.org/training/courses/>
- This Tech Talk is being recorded



Antitrust Policy Notice

- Linux Foundation meetings **involve participation by industry competitors**, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.



Code of Conduct

- The Linux Foundation and its project communities are **dedicated to providing a harassment-free experience** for participants at all of our events, whether they are held in person or virtually.
- All event participants, whether they are attending an in-person event or a virtual event, **are expected to behave in accordance with professional standards**, with both this Code of Conduct as well as their respective employer's policies governing appropriate workplace behavior and applicable laws.
- <https://openssf.org/community/code-of-conduct/>



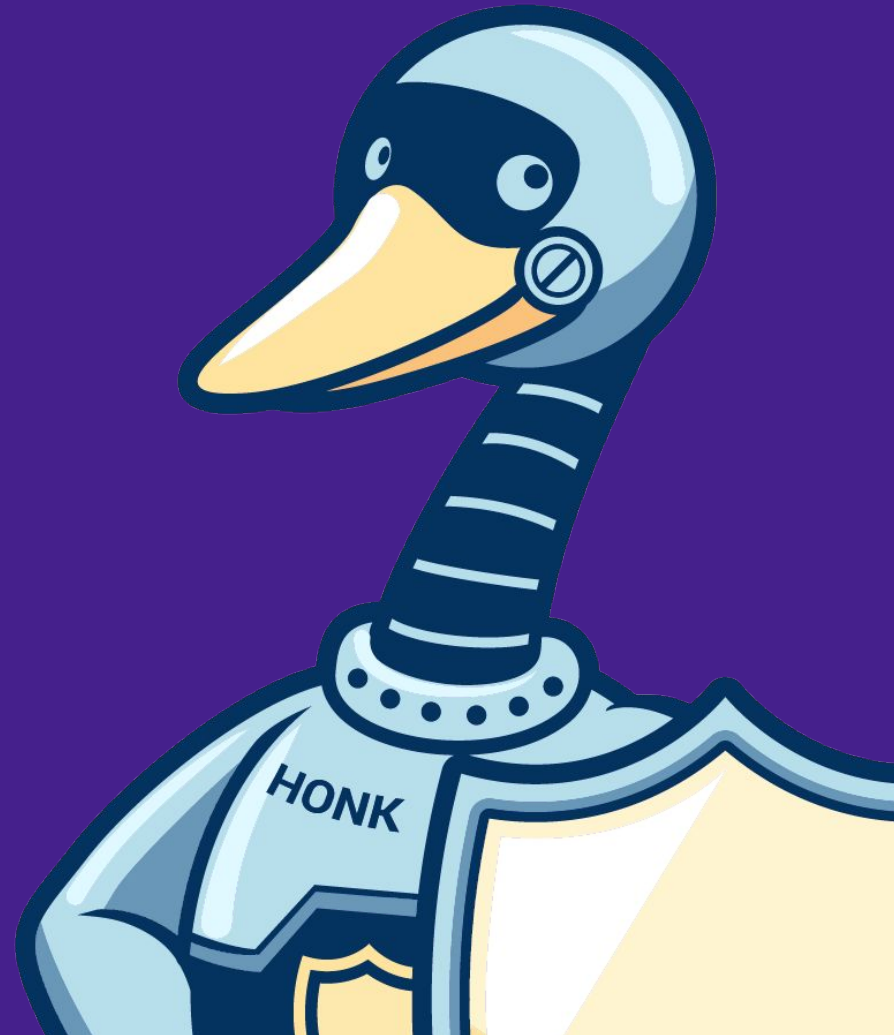
Housekeeping

Please submit your questions during the meeting by using the Q&A feature on Zoom.



Thank you!

Panelists



Michael Lieberman

CTO and Co-Founder, Kusari

Michael Lieberman is a technologist focused on IT transformations. His passion is in applying his expertise to use cases where privacy and security are paramount. He is also highly committed to open-source. Mostly recently he has been focused on work within the software supply chain security space. He is an OpenSSF SLSA steering committee member, tech lead for the CNCF Security Technical Advisory Group (STAG), lead on OpenSSF's FRSCA project and co-lead the CNCF's Secure Software Factory Reference Architecture. He is also a maintainer and architect on the GUAC project along with multiple other open source projects. Michael is also co-author of *Securing the Software Supply Chain* published by Manning.



Marcela Melara

Research Scientist, Intel Corporation

Dr. Marcela Melara is a research scientist in the Security and Privacy Group at Intel Labs. Her current work focuses on developing solutions for high-integrity software supply chains and building trustworthy distributed systems. She has several publications and patents filed related to her research, and leads a number of internal, academic and open-source efforts on software supply chain security. Prior to joining Intel, she received her PhD in Computer Science from Princeton University and did her undergraduate studies at Hobart and William Smith Colleges. She is a Siebel Scholar, a mentor for Científico Latino, and her research on CONIKS was awarded the Caspar Bowden PET Award. Outside of work, Marcela is an avid hiker, gardener, creative writer and bookworm.





Loreli Cadapan

CPO, ActiveState

Loreli's passion lies in solving challenges faced by security leaders and developers where application security is crucial. With years of experience in the enterprise software industry, successfully having worked from startups to enterprise and everything in between, she has held roles from coding, architecture, to product. Her focus is within the software supply chain security and securing open source, and is CPO of ActiveState, building products to power the world's software development teams and to accelerate their application security solutions.



Joshua Lock

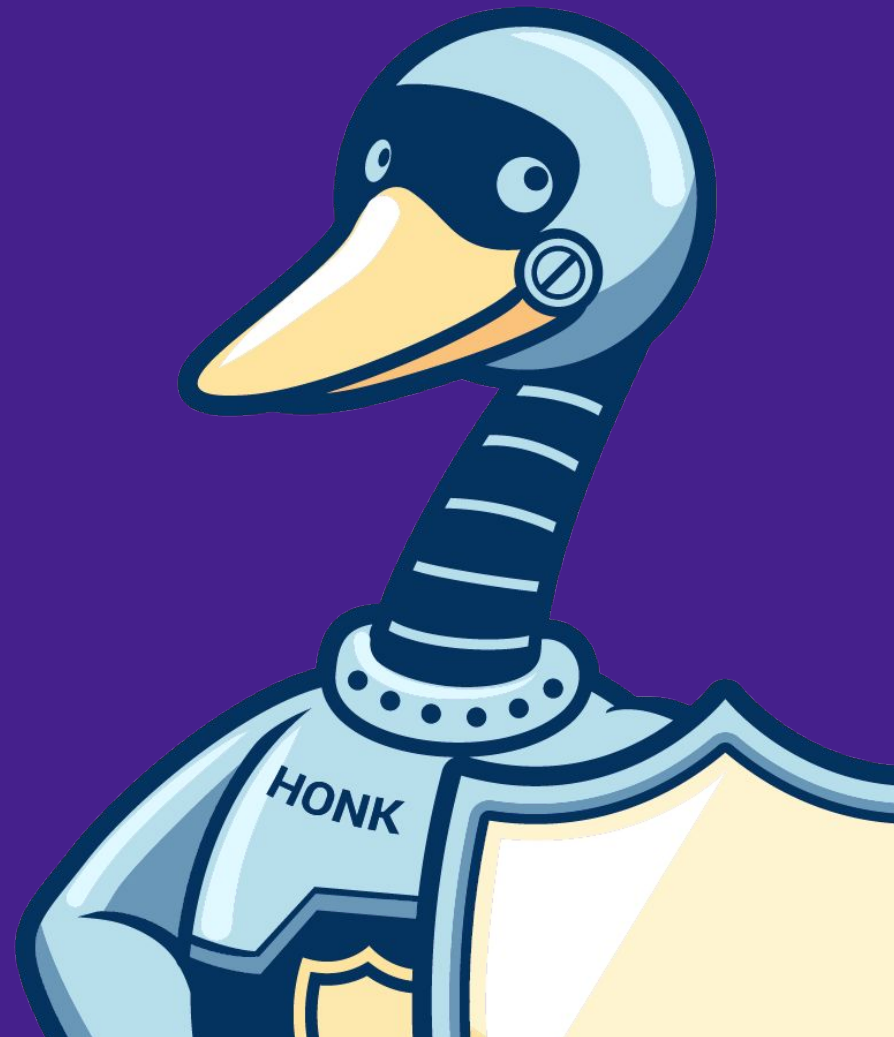
Distinguished Engineer, Verizon

Joshua is a software engineer with deep experience in the domains of software supply chain security and build systems. He is Open Source Architect at Verizon and actively engaged in upstream secure software supply chain projects and their integrations into open source ecosystems. He is a maintainer/editor of The Update Framework (TUF) and a steering committee member/maintainer of the Supply-chain Levels for Software Artifacts (SLSA) framework, as well as contributing to several other projects and communities.



Introduction to SLSA

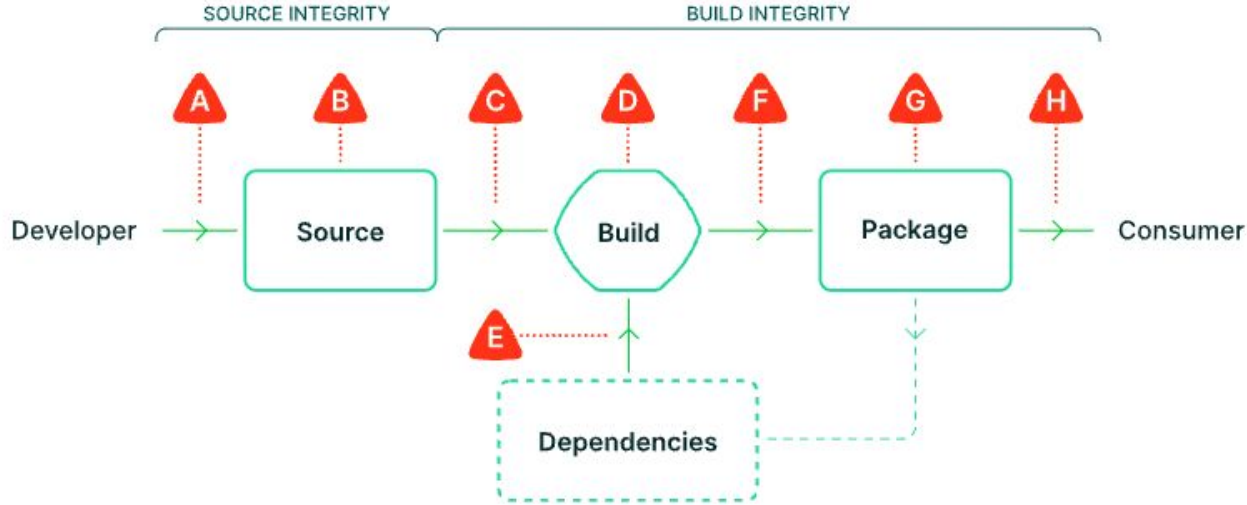
Michael Lieberman





SLSA

The Breadth of the Problem



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

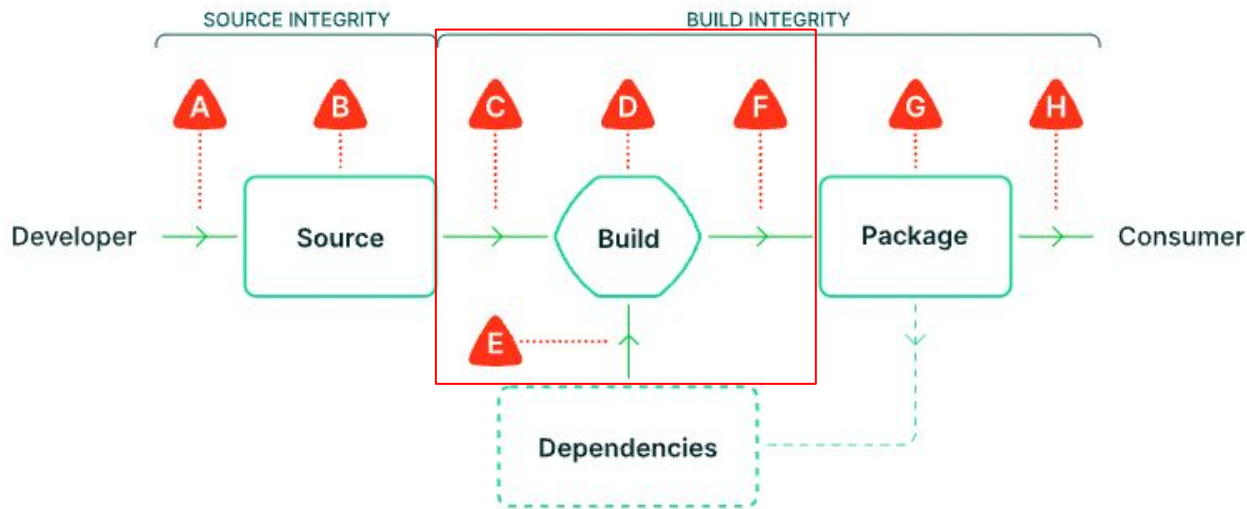
E Use compromised dependency

F Upload modified package

G Compromise package repo

H Use compromised package

The Breadth of the Problem



A Submit unauthorized change

B Compromise source repo

C Build from modified source

D Compromise build process

E Use compromised dependency

F Upload modified package

G Compromise package repo

H Use compromised package

SLSA 1.0 Build Track Focus

- Record the source the build ingests
- Record the dependencies the build ingests
- Record what the build generates
- Record the parameters to the build

The Depth of the Solution



Policy is here!
slsa-verifier

Policy and insight

Automation, risk management, and compliance throughout the SDLC.
Governance, developer assistance, and policy shifted left.

GUAC is here

Aggregation and synthesis

Smart aggregation turning data into meaning. Intelligent linking of project, resource, developer, artifact, repo, toolchain.

SLSA is here!

Software attestations

Schemas and sources for rich security metadata. SBOM, SLSA provenance, VEX, OSV, security scorecards, developer reputation, plus proprietary data.

Sigstore is here!

Trust foundation

A decentralized, flexibly anchored trust fabric. Signatures, strong identities, distributed timestamping, federation.

Yet Another Introduction to SLSA



SLSA is a Supply Chain Security Framework

SLSA Build 1.0 Has 3 Levels

SLSA is focused on the producer

SLSA is split into tracks

First released track is the build track

SLSA has a provenance statement format in in-toto attestation format

Easy to generate and consume JSON format

Why the Build Track?

There's a lot to software supply chain security

The build is critical to the foundation

Provenance is generally missing

How do we track from source to build artifact?



What SLSA Isn't

SLSA 1.0 Build Track

Can't prevent malware from being built... Just makes it easier to detect.

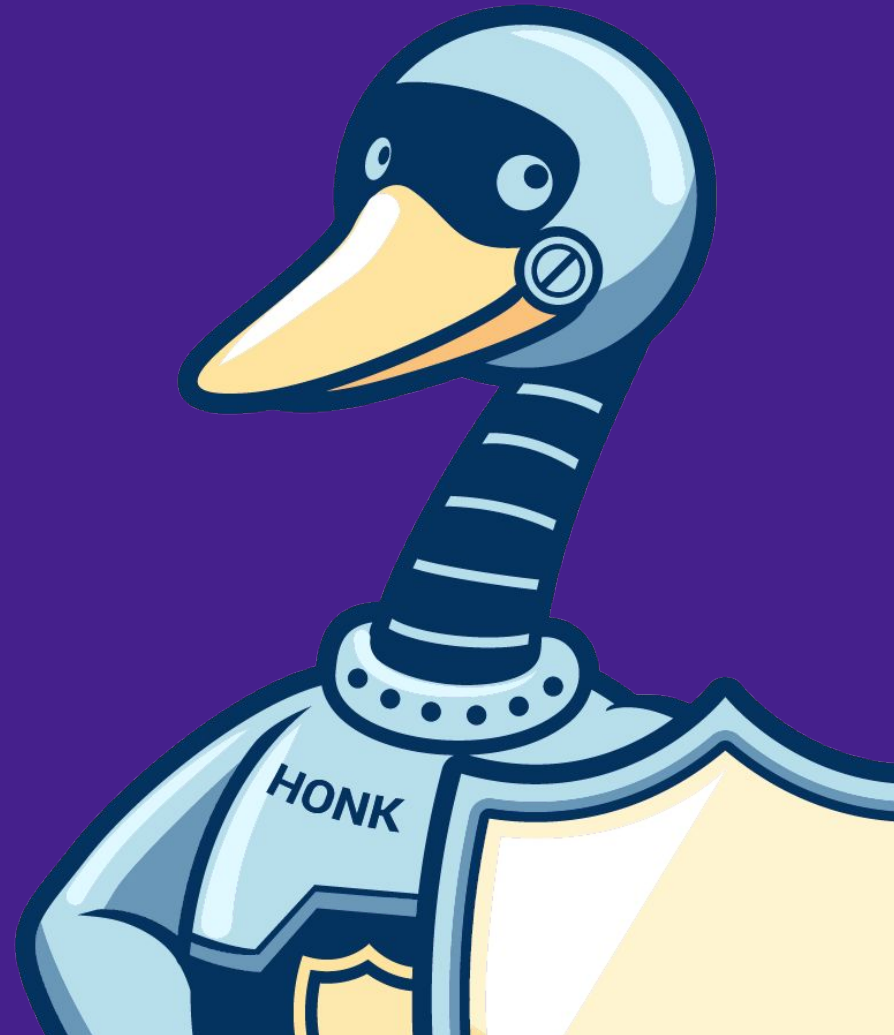
SLSA is not a comprehensive set of rules for ingestion

See the OpenSSF companion project Secure Supply Chain Consumption Framework (S2C2F)

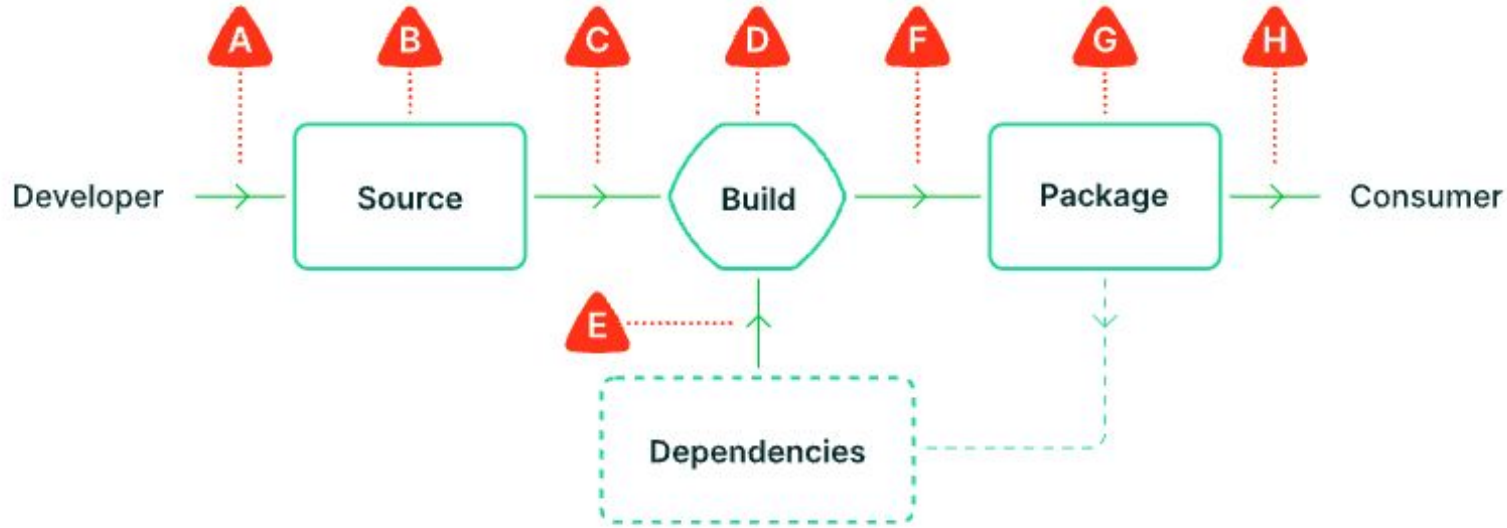


Trustworthiness and Transparency

Marcela Melara



The Breadth of the Problem: Many Possible Attack Points



What does it *really* mean to trust software?

Who wrote this software?

Who built this software?

Who released this software?

What components make up this software?

How was the software built?

What platform was the software built on?

Was the build compromised?

Was a legitimate version of gcc used?

Does the software contain buffer overflow vulnerabilities?

Does this software conform to regulatory requirements?

etc.



The Depth of the Solution

Trust?

What?
How?

Who?
When?

Trust foundation

A decentralized, flexibly anchored trust fabric. Signatures, strong identities, distributed timestamping, federation.



The Depth of the Solution

Trust?

Policy and insight

Automation, risk management, and compliance throughout the SDLC. Governance, developer assistance, and policy shifted left.

slsa-verifier,
in-toto Layouts



What?
How?

Aggregation and synthesis

Smart aggregation turning data into meaning. Intelligent linking of project, resource, developer, artifact, repo, toolchain.

GUAC

Software attestations

Schemas and sources for rich security metadata. SBOM, SLSA provenance, VEX, OSV, security scorecards, developer reputation, plus proprietary data.

SLSA, SBOM, etc.
+ in-toto

Who?
When?

Trust foundation

A decentralized, flexibly anchored trust fabric. Signatures, strong identities, distributed timestamping, federation.

Sigstore

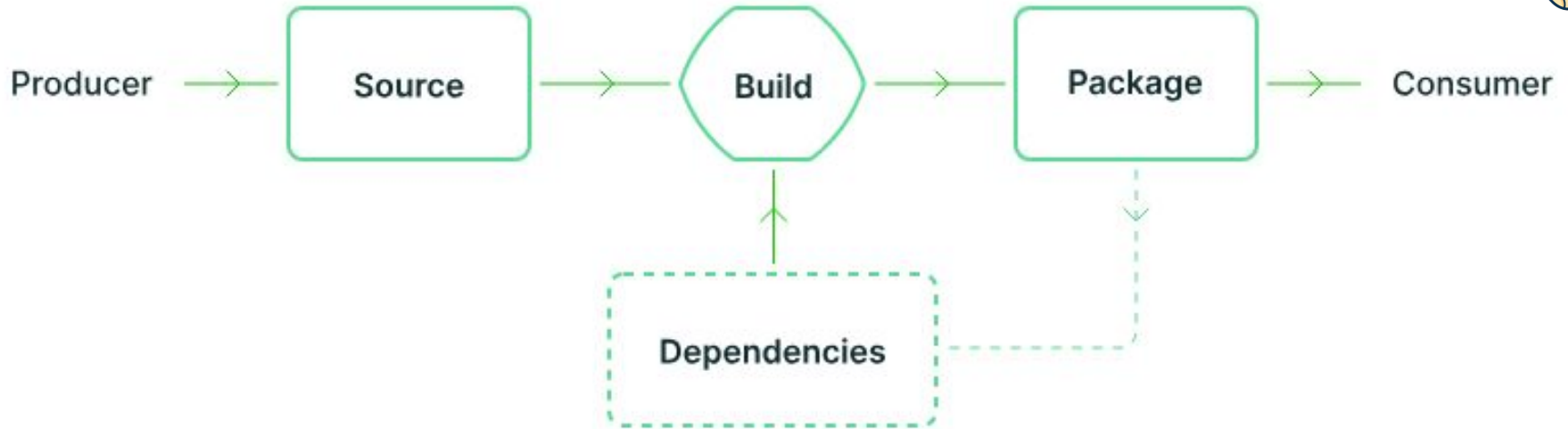


Software Attestations & Transparency...

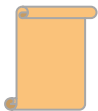
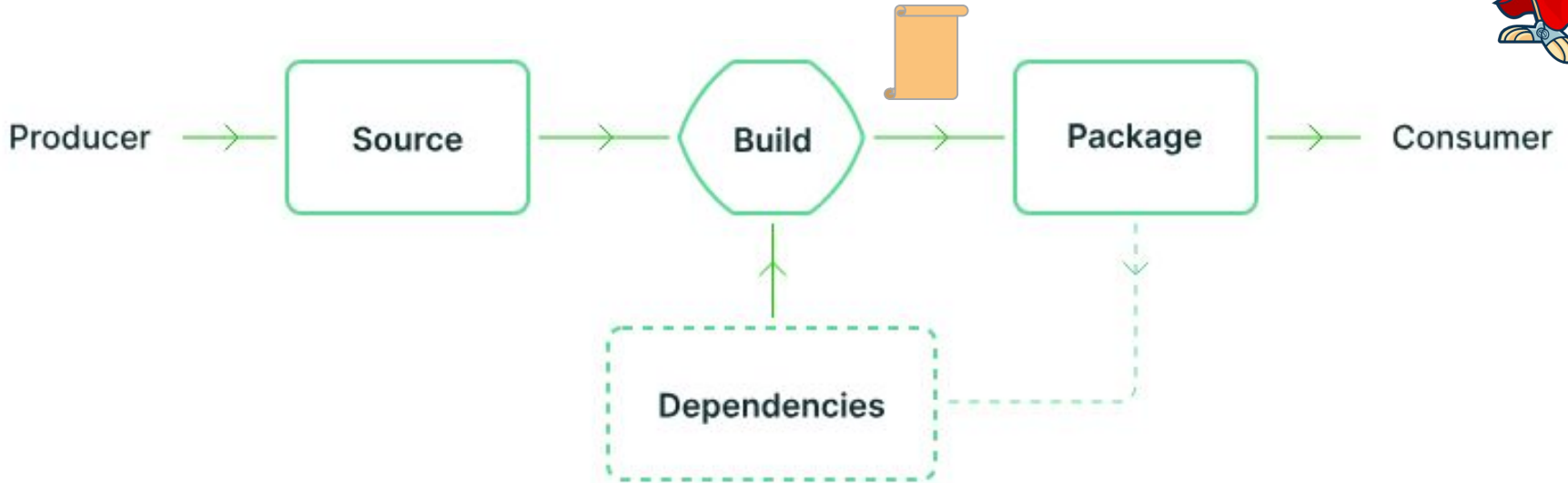


- ... capture information about any aspect of the SW supply chain.
- ... enable the verification of properties of software artifacts and their build.
- ... reduce the risk of security problems going undetected.

Step 1: Supply Chain Transparency with SLSA

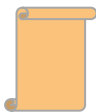
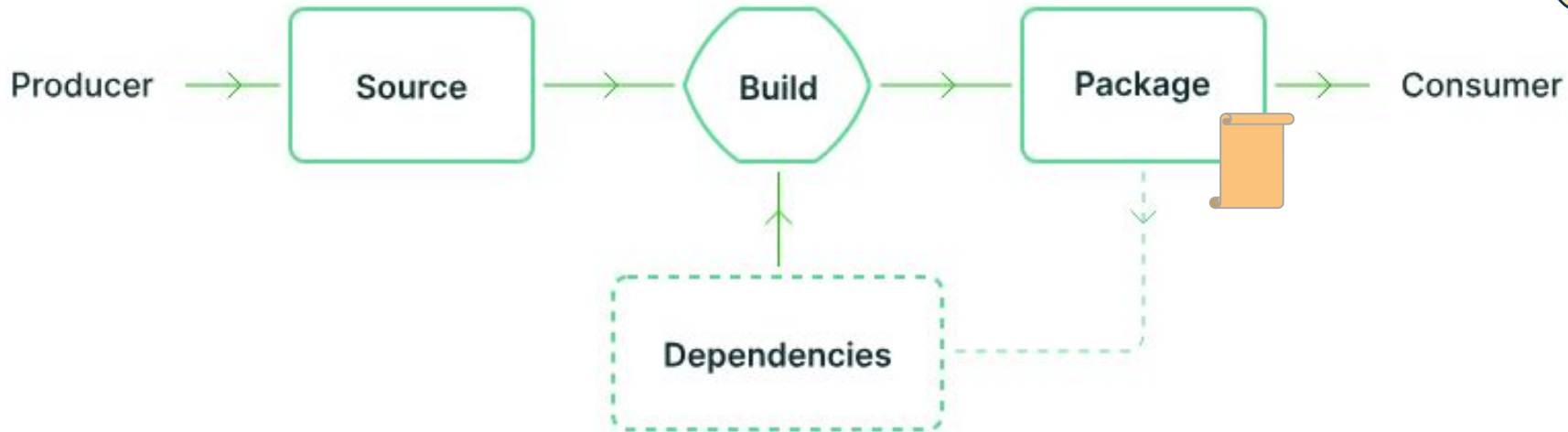


Step 1: Supply Chain Transparency with SLSA



in-toto attestation containing SLSA Provenance, authenticated by the producer

Step 2: Enabling Automated Trust Verification



in-toto attestation containing SLSA Provenance, authenticated by the producer

SLSA's Guiding Principles

- 1. Trust (a small number of) platforms, focus on artifacts.**
Why? Hardened artifact production can be scaled out.



SLSA's Guiding Principles



- 1. Trust (a small number of) platforms, focus on artifacts.**
Why? Hardened artifact production can be scaled out.
- 2. Trace software back to source code, not individuals.**
Why? Reduce risk of undetected tampering by trusted person/credentials.

SLSA's Guiding Principles



1. Trust (a small number of) platforms, focus on artifacts.

Why? Hardened artifact production can be scaled out.

2. Trace software back to source code, not individuals.

Why? Reduce risk of undetected tampering by trusted person/credentials.

3. Prefer attestations over inferences.

Why? Create a verifiable record across the supply chain.

Security Levels of SLSA

Loreli Cadapan



Security Levels of SLSA

- Organized into a series of levels
- Increasing integrity guarantees

How Does SLSA 1.0 Help?



Implementer	Requirement	Degree	L1	L2	L3
Producer	Choose an appropriate build platform		✓	✓	✓
	Follow a consistent build process		✓	✓	✓
	Distribute provenance		✓	✓	✓

Organization and Projects - e.g. Kubernetes, NPM packages, Hello World, etc.

How Does SLSA 1.0 Help?

Implementer	Requirement	Degree	L1	L2	L3
Build platform	Provenance generation	Exists	✓	✓	✓
		Authentic		✓	✓
		Unforgeable			✓
	Isolation strength	Hosted		✓	✓
		Isolated			✓

Build Systems - e.g. Tekton, Github Actions, Gitlab CI, Jenkins, etc.

Provenance Generation

- Exists
- Authentic
- Unforgeable



Isolation Strength

- Hosted
- Isolated



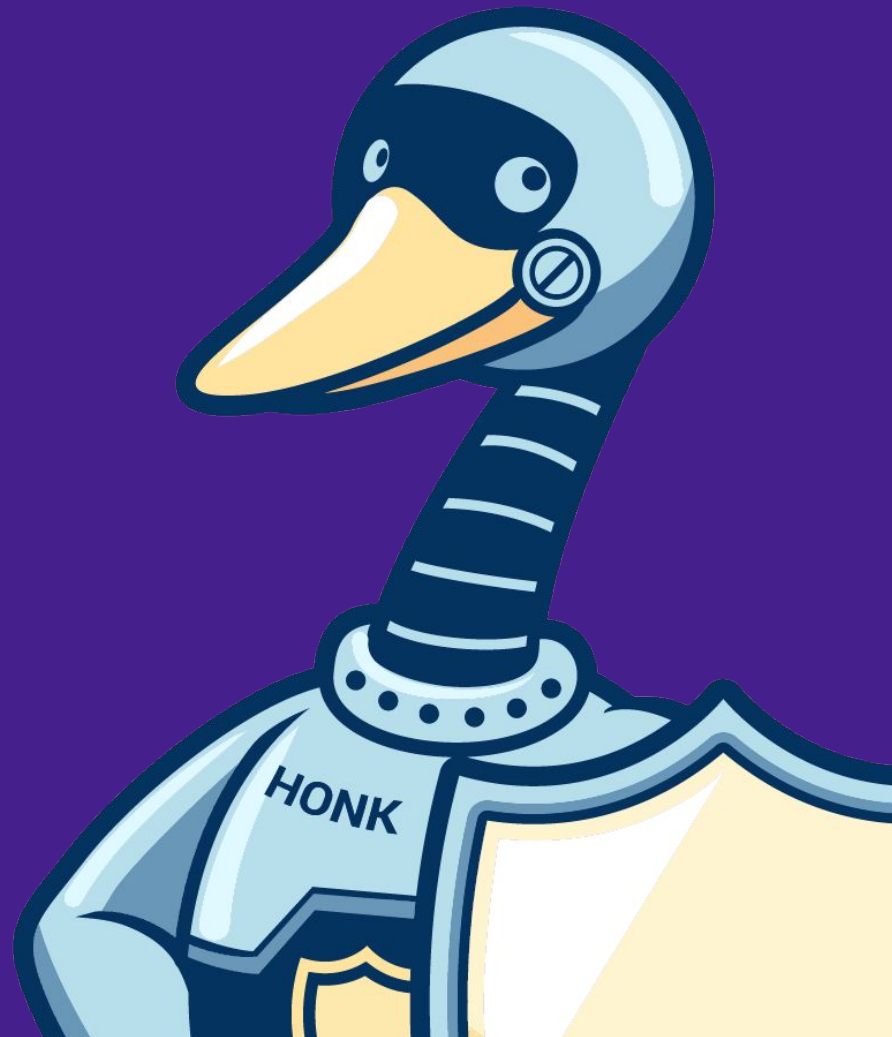
How Does This Protect Us?

- SLSA L1 – Something is better than nothing
- SLSA L2 – Associates identities and systems with the software
- SLSA L3 – Enforces security at the individual builds



Implementing SLSA

Joshua Lock



Who, what, where, when?



Who implements SLSA?

Platforms and ecosystems



How do their users gain confidence in that implementation?

Review builder evaluation (self-attestation or, in future, third-party certification)

<https://slsa.dev/spec/v1.0/verifying-systems>



How do users verify artefacts produced by a trusted SLSA implementation?

Verify the package and its associated provenance



SLSA verification



How do users* verify artefacts produced by a trusted SLSA implementation?

Add builder to verification tool's trusted builder configuration



Verify the signature on the provenance envelope



Ensure provenance values match expectations



*it is expected that verification is performed automatically by tools and that expectations are formed by trusted authorities (i.e., package registry) or automatically (to detect unexpected changes)

Let's see it!

slsa-github-generator: SLSA provenance generation for Github Actions.

Uses GitHub features + Sigstore to meet SLSA requirements:

Reusable workflow → Isolation strength: isolated

Workflow identity → Provenance generation: unforgeable

A detailed technical implementation specification is provided:

<https://github.com/slsa-framework/slsa-github-generator/blob/main/SPECIFICATIONS.md>

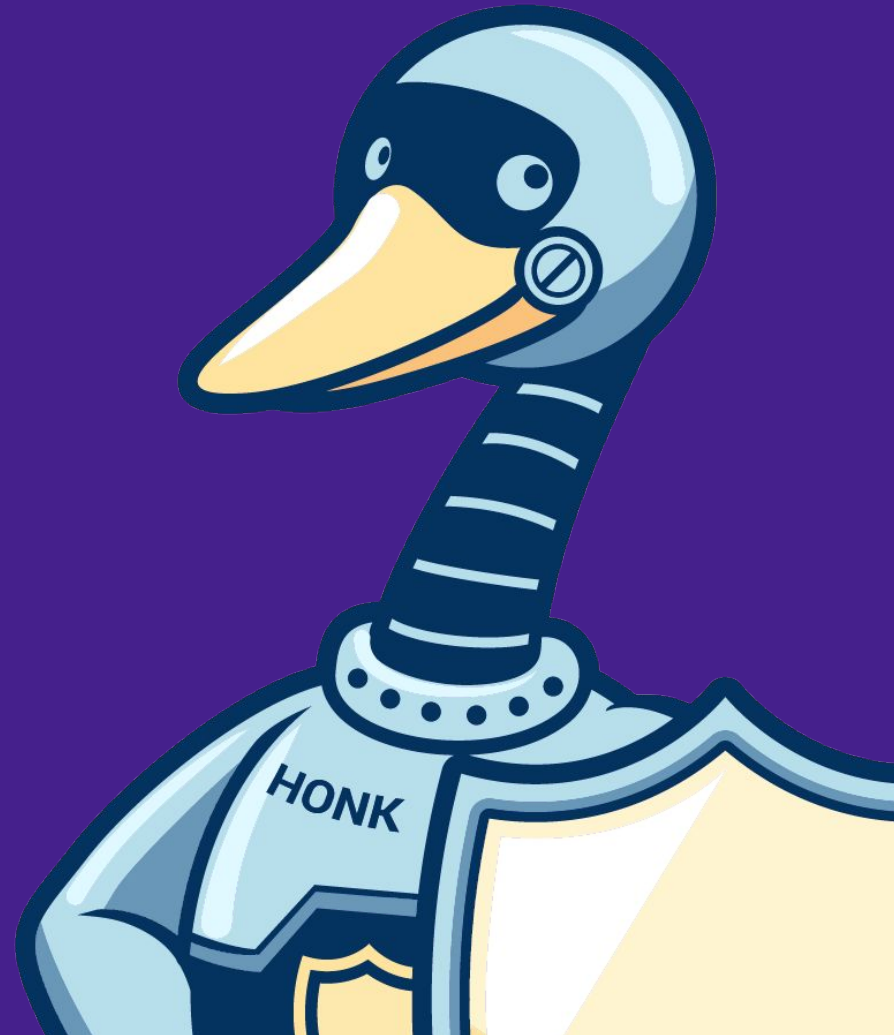


```

"ip": "192.168.1.10",
"hostname": "ip-192-168-1-10",
"mac": "08:00:27:00:00:00",
"vendor": "Arista Networks",
"model": "7700",
"series": "7700",
"generation": "1",
"part": "7700",
"revision": "1",
"serial": "1234567890",
"asset": "1234567890",
"parent": null,
"children": []
}

```

Industry Impact & Future Trends



The Future of SLSA



New Tracks!

Source (early discussion)

Dependencies (desired, not started)

Build System (early discussion)

Where's SLSA 4???

Come help us define it!

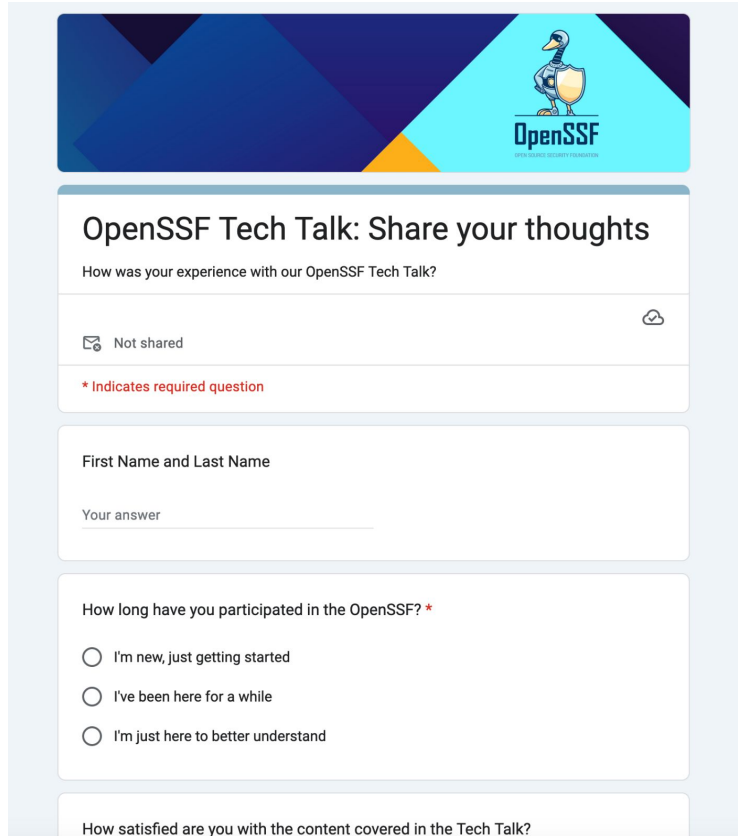
Conformance program

Get a fancy SLSA badge

Tools

Come integrate your tool with SLSA and join the SLSA tooling SIG!

Q&A



OpenSSF Tech Talk: Share your thoughts

How was your experience with our OpenSSF Tech Talk?

Not shared

* Indicates required question

First Name and Last Name

Your answer

How long have you participated in the OpenSSF? *

- I'm new, just getting started
- I've been here for a while
- I'm just here to better understand

How satisfied are you with the content covered in the Tech Talk?

Take our quick survey

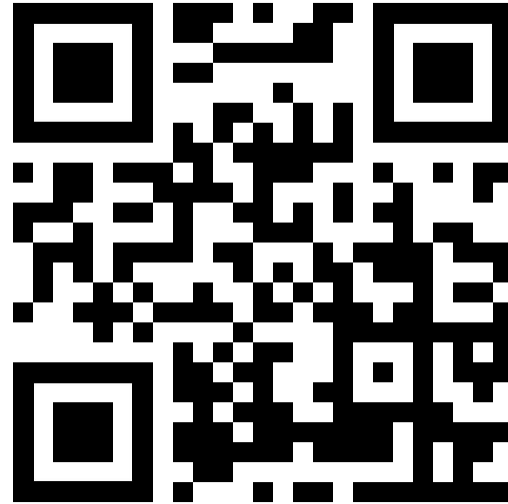
<https://forms.gle/QBQYLMezwCP3ow1T7>



Join Us!



OpenSSF Get Involved

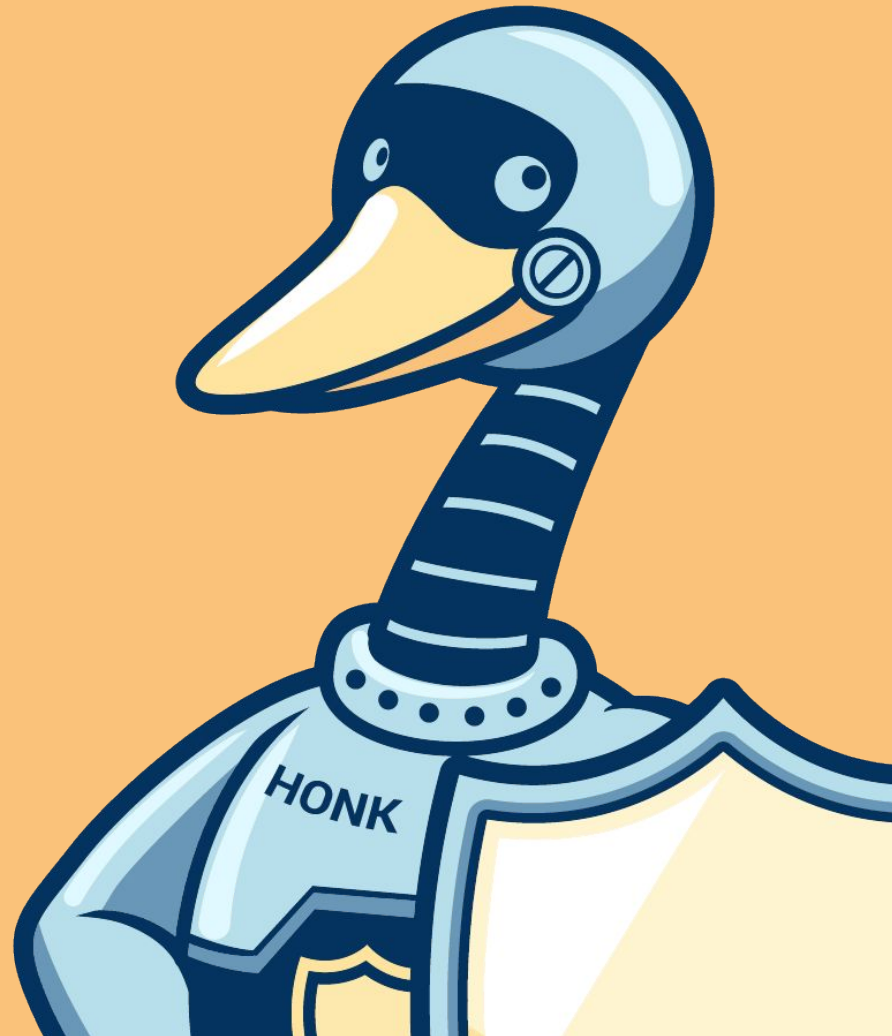


slsa.dev

Thank You!



Appendix



Is your organization a member?

<https://openssf.org/join>

Questions? Contact membership@openssf.org



Ways to Participate



[Join the OpenSSF Mailing List](#)



[Follow us on Twitter](#)



[Follow us on LinkedIn](#)



[Follow us on Mastodon](#)



[Follow us on Facebook](#)



[Subscribe to our YouTube Channel](#)



[Join a Working Group/Project](#)



[Access the Public Meetings Calendar](#)



[Participate on Slack](#)



[Follow OpenSSF on GitHub](#)



[Become an Organizational Member](#)



Open Source Security Foundation (OpenSSF)

560 followers San Francisco, CA <https://openssf.org>

Overview Repositories 54 Projects 12 Packages Teams 18 People 92 Insights Security

wg-best-practices-os-developers Public

The Best Practices for OSS Developers working group is dedicated to raising awareness and education of secure code best practices for open source developers.

JavaScript 416 56

wg-identifying-security-threats Public

The purpose of the Identifying Security Threats working group is to enable stakeholders to have informed confidence in the security of open source projects. We do this by collecting, curating, and ...

209 36

wg-security-tooling Public

OpenSSF Security Tooling Working Group

254 44

wg-securing-critical-projects Public

Helping allocate resources to secure the critical open source projects we all depend on.

262 30

wg-securing-software-repos Public

OpenSSF Working Group on Securing Software Repositories

39 6

wg-endusers Public

OpenSSF Endusers Working Group

8 5

Repositories

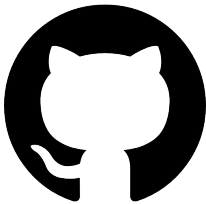
Find a repository...

Type

Language

Sort

New



People



View all

Top languages

Python Go JavaScript Perl

Most used topics

Manage

security fuzz-testing fuzzing github github-actions

Join a Technical Working Group

[@ossf](https://github.com/ossf)



Attend a Public Meeting

bit.ly/ossf-calendar



general

✓ Joined · 2,026 members · This channel is for workspace-wide communication and announcements. All memb...

wg_security_tooling

526 members · This WG is chaired by @Josh Bressers

wg_supply_chain_integrity

517 members · Our objective is to enable open source maintainers, contributors and end-users to understand an...

wg_securing_critical_projects

✓ Joined · 460 members · Helping allocate resources to secure the critical open source projects we all depend ...

slsa

✓ Joined · 451 members · discuss slsa framework

wg_best_practices_ossdev

428 members · The Best Practices for OSS Developers working group is dedicated to raising awareness and educ...

wg_vulnerability_disclosures

427 members · OpenSSF-Vulnerability Disclosures Working Group seeks to help improve the overall security of t...

security_scorecards

397 members · security scorecard project <https://github.com/ossf/scorecard> Bi-Weekly meetings on Thursday 1:...

Message on Slack

slack.openssf.org

Subscribe to the Mailing List

<https://openssf.org/sign-up>



Follow us on Social Media



[Twitter](#)

@openssf



[LinkedIn](#)

OpenSSF



[Mastodon](#)

social.lfx.dev/
@openssf



[YouTube](#)

OpenSSF



[Facebook](#)

OpenSSF