

OSLSA "Tooling" Meeting Notes

Meeting Info

When: every Friday at 10-11am Eastern ([OSSF Public Calendar](#))
Video Call: [Zoom](#)
Mailing list: [slsa-tooling](#)
Discussion: [#slsa-tooling](#) (for invite, see [slack.openssf.org](#))
Leads: Michael Lieberman eric.tice@wipro.com

[Meeting Info](#)

[Legal / code of conduct](#)

[Meeting notes](#)

[Future agenda items](#)

[2023-03-24](#)

[2023-03-17](#)

[2023-03-10](#)

[2023-03-03](#)

[2023-02-24](#)

[2023-02-17](#)

[2023-01-27](#)

[2023-01-20](#)

[2023-01-13 - Canceled](#)

[2023-01-06](#)

[2022-12-02 - Canceled](#)

[2022-11-18](#)

[2022-11-04](#)

[2022-10-14](#)

[2022-10-07](#)

[2022-09-30](#)

[2022-09-23](#)

[2022-09-16](#)

[2022-09-09](#)

[2022-09-02](#)

[2022-08-26](#)

[2022-08-19](#)

[2022-08-12](#)

[2022-08-05](#)

[2022-07-29](#)

[Current Tooling and Gaps](#)

Legal / code of conduct

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 Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

All meeting attendees must follow [SLSA's code of conduct](#).

Meeting notes

Most recent on top.

Future agenda items

2023-06-23 - CANCELED

Attendees:

-

Regrets:

-

Agenda:

-

2023-06-16 - CANCELED BECAUSE NO AGENDA

Attendees:

-

Regrets:

- Trishank Kuppusamy (Datadog) [conflicting meeting]

Agenda:

2023-06-09 - CANCELED BECAUSE NO AGENDA

2023-06-02 - CANCELED BECAUSE NO AGENDA

Attendees:

Agenda:

-

2023-05-26

Attendees:

- Mike Lieberman (Kusari)
- Arnaud Le Hors (IBM)
- Wietse Z Venema
- Brandon Mitchell (IBM)
- Matthew Wood (Intel)
- Jonathan Leitschuh (Alpha-Omega)
- Mikey Strauss

Agenda:

- SLSA and Jenkins
 - Is it even possible? Is it worth it?

- The build part could be locked to support SLSA even though the rest of Jenkins is insecure
- The effort to secure Jenkins might be more than it takes to get off Jenkins
 - Securing the build part might work but may not satisfy the people who want to use Jenkins for all the things that make it insecure
 - There is a Jenkins plugin under slsa-framework but unfortunately it is not being maintained.
<https://github.com/slsa-framework/slsa-jenkins-generator>
 - One needs to lock down the orchestrator, which means not allowing the use of plugins - this corresponds to what SLSA refers to as the control plane
 - Mike suggests we develop a document/guide describing what one might do to achieve Level 2. If anyone is interested please join and let others know.

2023-05-19

Attendees:

- Mike Lieberman (Kusari)
- Brandon Mitchell (IBM)
- Mikey Strauss (scribe)
- Wietse Z Venema
- Arnaud Le Hors (IBM)

Agenda:

- OSS NA Feedback
 - Lots of great feedback on SLSA
 - Where is v1 provenance in the github actions/workflows?
 - Test examples of sorts of attacks SLSA could prevent to help with testing of SLSA conformant building tooling
-


2023-05-05

Attendees:

- Fredrik Skogman (GitHub)
- Mike Lieberman (Kusari)
- Wietse Z Venema
- Brandon Mitchell (IBM)
- Tom Hennen (Google)
- Amanda Martin (Linux Foundation)

Agenda:

- OSS NA

- Panel
- Talks
 - [Talk on npm provenance](#)
- Who else is going?
 - Brandon Mitchell
 - Mike Lieberman
 - Amanda Martin
- GitLab npm provenance in progress: <https://github.com/npm/cli/pull/6375>
- What is required to get be able to be an npm provenance provider
 - The #1 thing is getting into Fulcio
 - List OIDCs to provide: <https://github.com/sigstore/fulcio/blob/main/docs/oid-info.md>
 - OIDC integration guide: <https://github.com/sigstore/fulcio/blob/main/docs/oidc.md>
- What is required to generate provenance for closed ecosystems?
 - Push on companies who are using SLSA but not generating packages intended for public consumption to at least talk about how they're using it.
- Webinars with the tools
- In-toto vs. SLSA
 - <https://slsa.dev/blog/2023/05/in-toto-and-slsa>
-  Malicious Compliance: Reflections on Trusting Container... - Coldwater, Coole...

2023-04-28

Attendees:

- Mike Lieberman (Kusari)
- Parth Patel (Kusari)
- Tom Hennen (Google)
- Brandon Mitchell (IBM)
- Shripad Nadgowda (Intel)

Agenda:

- 1.0 updates
 - Not a lot of tooling supports 1.0 yet.
 - Libraries
 - [In-toto attestations](#) has added language bindings for predicates with protos. SLSA provenance hasn't been added there yet since the proto is defined elsewhere.
 - See: <https://github.com/slsa-framework/slsa/issues/574>
 - We should double check in meeting what now supports 1.0
 - There is an npm workflow that Google is working on and plans to donate to Github

- Conformance program - <https://docs.google.com/document/d/1r6jM84mTa1dBJ6-KTPJKzCPUQ3GA8BuDlzFjbVfH7P8/edit>
 - Macaron - <https://github.com/oracle-samples/Macaron>
- Kubecon
 - <https://github.com/slsa-framework/slsa/issues/850>
- Spector
 - Tool intended to generate, validate, and verify SLSA very strictly.
- Folks still asking for some consistency in how to distribute SLSA
 - CDN friendly
 - Something like guidelines.
 - E.g. the URL should have some consistency
 - Is it important that this be standardized across ecosystems or just within them?
 - Keep verification & failure domains in mind??
 - Can GUAC or similar be used for querying the canonical location.
- SLSA 1.0
 - Guidelines on how to develop tools, libraries, etc. for builds as well as generating, validating, verifying SLSA provenance
 - Example libraries for SLSA. e.g. generating, validating provenance, build isolation libraries, etc. (VSA generation?)
 - Examples tools for SLSA, e.g. generating, validating provenance, build isolation libraries, etc. (VSA generation?)
 - Not just SaaS supporting.
 - Policy management?
 - Tooling to create & manage policies at scale (e.g. an entire org, entire packaging ecosystem, etc...)
 -

2023-04-14

Attendees:

- Mike Lieberman (Kusari)
- Fredrik Skogman (GitHub)
- Brandon Mitchell (IBM)

Agenda

- Canceled due to no agenda

2023-04-07

Attendees:

- Mike Lieberman (Kusari)
- Brandon Mitchell (IBM)
- Arnaud Le Hors (IBM)
- Wietse Z Venema (self)

Agenda:

- 1.0
 - Added ticket for cosign support
 -

2023-03-24

Attendees:

- Mike Lieberman (Kusari)
- Fredrik Skogman (GitHub)
- Brandon Mitchell (IBM)
- Arnaud Le Hors (IBM)
- Sanket Naik (Palosade)
- Shripad Nadgowda (Intel)

Agenda:

- Any Updates on 1.0
 - Some changing around isolated and ephemeral language
 - Some changing language around responsibilities of project builders, package ecosystems, consumers, etc.
-

2023-03-17

Attendees:

- Mike Lieberman (Kusari)
- Arnaud Le Hors (IBM)
- Brandon Mitchell (IBM)
- Manoharan Ramasamy (IBM)
- Shripad Nadgowda (Intel)
- Jonathan Leitschuh

Agenda:

- Any Updates on 1.0
 - <https://github.com/slsa-framework/slsa/issues/574>
 - Frederik created npm tracking issue for 1.0

- Mike is looking at maybe making a quick and dirty SLSA provenance format validator

2023-03-10

Attendees:

- Mike Lieberman (Kusari)
- Fredrik Skogman (GitHub)
- Arnaud Le Hors (IBM)
- Wietse Z Venema (self)
- Brandon Mitchell (IBM)
- Shripad Nadgowda (intel)

Agenda:

- Any updates on 1.0
 - <https://github.com/slsa-framework/slsa/issues/574>
- Npm policy
 - <https://github.com/slsa-framework/slsa-proposals/pull/8>
- Still waiting to hear more on conformance program
-

2023-03-03

Attendees:

- Mike Lieberman (Kusari)
- Brandon Mitchell (IBM)
- Fredrik Skogman (GitHub)
- Wietse Z Venema (self)
- Trishank Kuppusamy (Datadog)
- Arnaud Le Hors (IBM)
- Eric Tice (Wipro)

Agenda:

- Looking for additional co-leads of this meeting.
- 1.0 draft is out. What tools are planning to, or should be implementing 1.0 (attestation spec, requirements)
 - Mike - Reached out to Tekton Chains. Some but not much support for implementing 1.0
 - Tools
 - Tekton Chains
 - <https://github.com/tektoncd/chains/issues/728>

- Docker buildkit: [Provenance attestations | Docker Docs](#)
 - <https://github.com/moby/buildkit/issues/3684>
- npm CLI <https://github.com/npm/cli/> (no issue exists yet, Fredrik Skogman to create one)
- SLSA verifier [GitHub - slsa-framework/slsa-verifier: Verify provenance from SLSA compliant builders](#)
 - <https://github.com/slsa-framework/slsa-verifier/issues/515>
- SLSA Github Generator: <https://github.com/slsa-framework/slsa-github-generator>
 - <https://github.com/slsa-framework/slsa-github-generator/issues/1734>
- GitLab attestations (2022): <https://about.gitlab.com/blog/2022/08/10/securing-the-software-supply-chain-through-automated-attestation/>
- TestifySec Witness(?): <https://github.com/testifysec/witness>
- SLSA 1.0 Validator - Tool to help verify that a SLSA document complies with the spec.
- Arno - We need to get to the point where people can mix and match different tools from different independent implementations
- <https://github.com/oracle-samples/Macaron>

2023-02-24

Mike Lieberman will miss the meeting, so if anyone wants to facilitate the meeting, they're more than welcome to!

Agenda:

- 1.0 draft should be going live on 2/24 or in the next few days
-

2023-02-17

Attendees:

- Mike Lieberman (Kusari)
- Fredrik Skogman (GitHub)
- Wietse Z Venema (emeritus)
- Eric Tice (Wipro)
- Andrew McNamara (Red Hat)

Agenda:

- <https://github.com/sigstore/fulcio/pull/945>

- Wietse mentioned Tom Hennen's Verification Summary Attestation (VSA) attestation that may be useful for closed-source builds, <https://security.googleblog.com/2022/04/how-to-slsa-part-1-basics.html>

2023-01-27

Attendees:

- Fredrik Skogman (GitHub)
- Brandon Mitchell (IBM)
- Mike Lieberman (Kusari)
- Parth Patel (Kusari)
- Aaron Bacchi (Verizon)
-

Agenda:

- [Mike Lieberman] SLSA (and other Metadata) API Spec
 - Bundle - accepted and released as a 0.1.0
 - Different interest at different levels between ecosystems
 - Some want to be involved in developing a spec/standard
 - Some want to be involved in implementing what others decide
 - Some largely want to be left out
 - Principles
 - Make it CDN friendly
 - Package ecosystems get on the order of millions to billions requests an hour. Application server heavy requests could be rough.
 - Should separate out the package from the metadata documents
 - I.e. not included in the actual tarball or whatever
 - For a single package there should be a consistent path for finding the various provenance documents
 - Keep it simple
 - Not many API nouns and verbs. We don't want this to be some giant API supporting every use case imaginable.
 - Concerns
 - Notary v1 tried a separate API and it failed because of things moving (name hard coded in the reference) and the need to run a separate server.

2023-01-20

Attendees:

- Mike Lieberman (Kusari)
- Fredrik Skogman (GitHub)
- Trishank Kuppusamy (Datadog)
- Jay White (Microsoft)

Agenda:

- 1.0 near end of Feb
 - Looking for contributors for getting tools ready for 1.0 spec and requirements
 - [GitHub Generator](#)
 - [Verifier](#)
 - FRSCA
 - Tekton
 - Npm cli
 - Tracking issue - <https://github.com/slsa-framework/slsa/issues/574>
- Attestation discovery and distribution
 - Need for a best practices document
 - <https://github.com/slsa-framework/slsa/issues/269>

2023-01-13 - Canceled

2023-01-06

Attendees:

- Mike Lieberman (Kusari)
- Brad Beck (Citi)
- Parth Patel (Kusari)
- Trishank Kuppusamy (Datadog)

Agenda:

- 1.0 near end of Feb
 - 1.0 spec PR - <https://github.com/slsa-framework/slsa/pull/525/files>
 - We want tools to be ready to produce 1.0 metadata sooner rather than later

2022-12-02 - Canceled

2022-11-18

Attendees:

- Mike Lieberman (Kusari)
- Eric Tice (Wipro)
- Fredrik Skogman (GitHub)
- Matt Spiekerman (Dow)
- Sebastien Awwad (Anaconda)
- Trishank Kuppusamy (Datadog)
- Shripad Nadgowda (Intel)
- Justin Cappos (NYU)
- Parth Patel (Kusari)
- Sunny Yip (Kusari)
- Aaron Bacchi (Verizon)

Agenda

- Introductions
 - Justin Cappos - active in CNCF Tag security, TUF/in-toto
 - Matt Spiekerman - from Dow
 - Trishank Kuppusamy - from Datadog
 - Sunny Yip - from Kusari
- Updates
 - Npm RFC that pushes Sigstore/SLSA for npm packages is merged.
 - SigstoreJS being built out
 - Sigstore TUF being built out
 - Some challenges with cJSON
 - Potentially not valid json. Some uncertainty.
 - <https://github.com/theupdateframework/specification/issues/92>
 - Sigstore itself uses <https://datatracker.ietf.org/doc/html/rfc8785>
- Verifier/stages of verification
 - <https://github.com/slsa-framework/slsa-verifier>
 - As we are pushing for 1.0, we want to make sure we have some set of tools, libraries, services, etc. that can verify SLSA attestations.
 - Stages
 - Stage one is ensuring that the content of a SLSA attestation actually complies with the specification of the predicate, e.g. url fields are valid url.
 - Schema validation
 - Stage two is verifying signatures on the SLSA attestation envelope.

- Which envelope format?
 - [DSSE?](#)
 - Can help with parsing untrusted payloads
 - Stage three is inspecting the content inside of the SLSA attestation
 - e.g. does the URI of the builder match some policy?
 - Where does the policy come from?
 - There is no SLSA-described format yet
 - Can use third-party policy engines
 - Run as a public api/service? [Laurent Simon]
 - Builders
 - Certification
 - Eric - Working on an end to end CI solution that can help. How do you do this with multiple different tools? What is the responsibility of the tools vs. the project.
 - Verification of a Project's pipeline against. SLSA
 - Automated certification?
 - Something like scorecard?
 - Continuous certification?
 - Predicate dictionary
 -

2022-11-04

Attendees:

- Fredrik Skogman (GitHub)
- Anand Chugh(Microsoft)
- Parth Patel (Kusari)
- Shripad Nadgowda (Intel)

Agenda:

- Interesting talks at KubeCon?
 - SBOM XRay super power
 - FRSCA
 - GUAC
 - Tetragon in general
- SBOM-scorecard
 - <https://github.com/eBay/sbom-scorecard>

- Evaluate SBOMs to determine their usefulness and validity
- Use metrics to push for change and fixes

2022-10-14

Attendees

- Mike Lieberman (Kusari)
- Sandra Monroe (Anaconda)
- Shaun Lowry (ActiveState)
- Brandon Mitchell (IBM)
- Wietse Z Venema (Google)
- Fredrik Skogman (GitHub)
- Aaron Bacchi (Verizon)
- Shripad Nadgowda (Intel)
- Parth Patel (Kusari)
- Mark Lodato (Google)
- Isaac Hepworth (Google)
- Eric Tice (Wipro)
- Jay White (Microsoft)

Agenda:

- Who is going to be at Kubecon? Any good talks?
 - Eric Tice - eric.tice@wipro.com - Panel discussion on "What data tells us about Software Supply Chain Security and what to do about it"
 - Fredrik Skogman - and sigstorecon - kommendorkapten@github.com
 - Shripad Nadgowda: shripad.nadgowda@intel.com
 - Parth Patel - parth@kusari.dev
 - Mike Lieberman - mike@kusari.dev
 - Cheng Lee cleee@anaconda.com
 - Hassam Mian hmian@anaconda.com
 - Nicole Schwartz nicoles@activestate.com
 - Asra Ali [SigstoreCon too] asraa@google.com
 - Some folks from OCI will be there
 - Talks
 - FRSCA talk - generating SLSA through secure build
 - Runtime Attestation (SecurityCon)
 - GUAC talk - Ingesting SLSA attestations (along with SBOMs and other metadata) for analysis and policy
 - Road to SLSA 4 (SigstoreCon) - <https://sigstoreconna22.sched.com/event/1Aykv/the-road-to-slsa4-a>

[applying-the-sigstore-ecosystem-in-a-corporate-environment-alex-il-gayev-cycode](#)

- [Why You Can Trust Sigstore Signatures](#) (sigstore trust root; SigstoreCon)
- [OIDC Security with Fulcio](#) (SigstoreCon)
- What tools are generating 100% valid SLSA vs. mostly valid SLSA
 - Task- index / list tools out there
 - Tekton (<https://github.com/tektoncd/chains>)
 - Github action
<https://github.com/slsa-framework/slsa-github-generator>
 - Gitlab
https://docs.gitlab.com/ee/ci/runners/configure_runners.html#artifact-attestation
 - The Jenkins demo?
 - npm cli (in-progress)
 - Generate JSON schema?
 - Mark: Considered it originally but never prioritized. JSON Schema is not pleasant to write by hand. It's machine readable but not human readable. I considered writing the schema in protobuf descriptor, but that is too opinionated and wouldn't support our actual format due to the `buildType` and opaque objects. Proto does have an Any format, but that is proto-specific and wouldn't fit our use case well. I wish there were a better generic, human-readable schema language.
 - Mike: will create github issue for this

2022-10-07

Attendees

- Mike Lieberman (Kusari)
- Isaac Hepworth (Google)
- Preston Moore (Anaconda)
- Mark Lodato (Google)
-

Regrets

- Shaun Lowry (ActiveState) - Canadian Thanksgiving
- Brandon Mitchell (IBM)

Agenda:

- No agenda - cancelled

2022-09-30

Attendees

- Mike Lieberman (Kusari)
- Mark Lodato (Google)
- Shaun Lowry (ActiveState)
- Fredrik Skogman (GitHub)
- Aaron Bacchi (Verizon)
- Brandon Mitchell (IBM)
- Preston Moore (Anaconda)
- Wietse Z Venema (Google)
- Shripad Nadgowda (Intel)
-

Regrets

- Jay White (Microsoft) - Dentist

Agenda

- Wietse: Starting work on SLSA policy prototype for NPM
 - Follow up to NPM team's current work on provenance distribution
 - Prototype for future round of changes to npm client to warn or reject if the provenance does not meet some expectations (e.g. wrong source repo)
 - Email to reach out- wietse@google.com
- GUIX paper
 - <https://arxiv.org/ftp/arxiv/papers/2206/2206.14606.pdf> - references some stuff with SLSA
- Attestation distribution and discovery doc:
<https://docs.google.com/document/d/163H4cCeT7JaeZGYzkM2yxhavGigIirQjNxH3daVwo/edit#heading=h.jg3vckahrjgm>
-

2022-09-23

Attendees


- Mike Lieberman (Kusari, CNCF, Tools lead)
- Shaun Lowry (ActiveState)
- Fredrik Skogman (GitHub)
- Wietse Z Venema (Google)
- Brandon Mitchell (IBM)
- Jeremy Rickard (MSFT)
- Preston Moore (Anaconda)
- Nisha Kumar (Oracle)
- Kris K (Google)

- Shripad Nadgowda (Intel)

Regrets

- Mark Lodato (Google) - meeting conflict

Agenda


- Updates
 - Puerco's tejolet tool - <https://github.com/puerco/tejolote>
 - Kubernetes SIG Release meeting:  Kubernetes SIG Release
 - Starting work on SLSA policy(non-enforcing) for NPM (Wietse)
 - Alas my microphone wasn't cooperating.
- [Jeremy Rickard] Show slsa publishing w/ ORAS / Artifacts @ Microsoft (follow up to slack: <https://openssf.slack.com/archives/C03PDLFET5W/p1663341305537709>)

2022-09-16

Attendees

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Fredrik Skogman (GitHub)
- Shaun Lowry (ActiveState)
- Aaron Bacchi (Verizon)
- Asra Ali (Google)
- Brandon Mitchell (IBM)
- Jay White (Microsoft)
- Shripad Nadgowda (Intel)
- Sebastien Awwad (Anaconda)
- Parth Patel (Kusari)
- Wietse Z Venema (Google)
- Isaac Hepworth (Google)

Agenda

- Updates
 - Samsung Jenkins plugin - <https://github.com/Samsung/slsa-jenkins-generator>.
-  Attestation Distribution and Discovery Design Doc
 - OCI - Release candidate waiting to be tagged
 - The [proposal](#) supports an optional `?artifactType=<artifactType>` parameter to filter by artifact type. For attestations, this will always be the DSSE media type (not yet registered), with no way to filter by the payload type (one layer

down) or even predicate type (two layers down). Should we add parameters to the media type to allow this?

- No. Instead, add these other properties as attributes to the artifact in OCI. Clients are expected to fetch them all and filter client-side. This would also allow filtering by things like the signer.
- Presentation of OCI Referrers: <https://youtu.be/-QTz81JvbCo>
- NPM - RFC still open. One more NPM cli meeting. RFC should be merged in next few weeks.
- Convention for naming bundle files
 - In-toto recommendation here: <https://github.com/in-toto/attestation/blob/main/spec/bundle.md#file-naming-convention>
 - Shaun: it's not so much about multiple subjects, but about what the consumer is trying to verify. So if you're verifying file X,
 - Mark: agreed. Will file issue.
 - Fredrik: NPM is planning to use the Sigstore bundle format rather than JSONL so that it can be uploaded to Fulcio naturally.
 - Includes not just the attestation but also the Rekor signed timestamp attestation so that it can be verified offline.
 - Link to sigstore bundle: <https://github.com/sigstore/cosign/pull/2204>
 - Term "bundle" is used differently: Sigstore means a single attestation (envelope + extra stuff), In-toto means multiple attestations.
 - Mark: Might be good to move some of this information to DSSE.

2022-09-09

Attendees

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Shaun Lowry (ActiveState)
- Shripad Nadgowda (Intel)
- Jay White (Microsoft)
- Sebastien Awwad (Anaconda)
- Preston Moore (Anaconda)
- Aaron Bacchi (Verizon)
- Isaac Hepworth (Google)
-

Agenda

- Json Lines discussion - Json lines is the recommended file based pattern for distributing in-toto attestations

- Updates on other tools in attestation discovery and distribution
 - [Attestation Distribution and Discovery Design Doc](#)
- [Trust in attestations](#) examples (see slides 17, 18, 19)

Notes

2022-09-02

Meeting canceled due to holiday weekend in US

2022-08-26

Attendees

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Isaac Hepworth (Google)
- Daniel Appelquist (Snyk)
- Shaun Lowry (ActiveState)
- Fredrik Skogman (GitHub)
- Aaron Bacchi (Verizon)
- Brandon Lum (Google)
- Laurent Simon (Google)
- Eric Tice (Wipro)
- Shripad Nadgowda (Intel)
- Parth Patel (Kusari)
- David Dillard (Veritas)
- Mike Brown (IBM)

Agenda

- Finalize initial set of reqs for tooling
 - Not everything, but enough for folks to start hands on keyboard working on it.

Notes

- OCI: <https://github.com/opencontainers/image-spec/pull/934>
<https://github.com/opencontainers/distribution-spec/pull/335>

2022-08-19

Attendees (please add yourself)

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Eric Tice (Wipro)
- Fredrik Skogman (GitHub)

- Shaun Lowry (ActiveState)
- Eric Herget (Red Hat)
- Mark Lodato (Google)
- Isaac Hepworth (Google)
- Shripad Nadgowda (Intel)
- Sebastian Crane
- Aaron Bacchi
- Parth Patel (Kusari)
- David Dillard (Veritas Technologies)
- Roy Williams (Microsoft)
- Aranyajit Sarkar (Wipro)

Agenda:

- [Mike Lieberman] Prioritize any big ticket items for specific tooling

Notes:

- Questions about who to trust and when, and for what
- There are multiple package managers and it is not practical to implement SLSA verification in every package manager
- We can potentially leverage existing working happening on the SBOM side for distribution.
- Distribution and discovery - [Distribution and Discovery Design Doc](#)

2022-08-12

Attendees (please add yourself)

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Mark Lodato (Google)
- Fredrik Skogman (GitHub)
- Shaun Lowry (ActiveState)
- Sebastien Awwad (Anaconda)
- Matt Rutkowski (IBM)
- Parth Patel (Kusari)
- Shripad Nadgowda (Intel)

Agenda:

- NPM discussion - <https://github.com/npm/rfcs/pull/626>
- Identify and classify gaps/what level
- Figure out where we prioritize

Notes:

- NPM RFC
 - Want to make sure we have split out the value for both Maintainers and Consumers of SLSA for NPM
 - Conversations around Trusted Builders and Github Actions
 - Provenance stored in Rekor + in NPM registry, details still being worked out, ambiguous in the RFC because we wanted comments more on the higher level concept
 - Need to come up
 - Concerns around vend with an API that makes it efficient to query provenance in batchorization, perhaps reproducible builds could be a viable path in the future to address
 - What if the artifact itself does postinstall scripts?
 - Allow any type of provenance to be uploaded, but for the “verified” badge, likely NPM will maintain a list of trusted builders
- Added to [Current Tooling and Gaps](#) below
- Prioritization:
 - Top: Provenance Distribution and Discovery + Package Ecosystem Policy Integration (e.g. canonical source repo) - particularly around the open source ecosystem.

2022-08-05

Attendees (please add yourself)

- Mike Lieberman (Kusari, CNCF, Tools lead)
- Shaun Lowry (ActiveState)
- Aaron Bacchi (Verizon)
- Wietse Z Venema (Google)
- Brandon Lum (Google)
- Parth Patel (Kusari)
- Naveen Srinivasan
- Ian Lewis (Google)
- Laurent Simon (Google)
- Rob Winch (VMware)
- David A. Wheeler (Linux Foundation)
- Joshua Lock (VMware)
- Dmitry Raidman (Cybeats)

Agenda:

- [Joshua Lock] Please vote for which ecosystem(s) slsa-github-generator should support next: <https://github.com/slsa-framework/slsa-github-generator/issues/687>
- Go over the categories and list of projects

- We should also move this into a spreadsheet or better format.
- Identify gaps in the categories or list of projects
- Identify where this group should focus its development effort on.

Notes:

- David - We should have a maintained page on slsa.dev with this landscape, so people can easily find "tools that may help you"
 - Closest thing currently: <https://slsa.dev/get-started>, see <https://github.com/slsa-framework/slsa/issues/407>
 - David: Probably should be a separate page
 - David: It's okay to *also* make blog posts, etc., but those go out-of-date. Need a central maintained page for people so we can keep it up to date.
 - David: Make it easy to find!
 - We can use last week's list as a starting point
- Training/education
 - David: Make that a separate web page on slsa.dev, with a URL & title that hints at its contents, so it's easy to find
 - Linking to blogs from it is fine, as long as that's the current best info
 - Obviously update over time
- Naveen will create initial PRs for these two pages (Tooling, Education/Best Practices)
 - Get started
 - Tools
- On "Getting Started" change "Tools" to "Sample OSS Tools" - and be picky (we don't want to overwhelm people)
 - The "Tools" page can be much more complete
- Is there an issue listing closed source tools?
 - David: I don't think there's an issue in general. We need to ensure that we don't include closed source tool X & exclude closed source tool Y just because they're a competitor to someone in the group. It's okay to list only OSS tools, and I think it's okay to list closed source tools as well as long as the group agrees to list closed source tools. If there's a specific legal question, please let me know the specific question (via email dwheeler@linuxfoundation.org) so I can raise it up.
- Providing guidance/documentation for tools to implement SLSA, and recognise whether there are architectural barriers to implementing support for higher SLSA levels, falls within the scope of this team

2022-07-29

Attendees (please add yourself)

- Mike Lieberman (Kusari, CNCF, Tools co-lead)

- Rob Winch (VMware, Spring Security Project lead)
- Wietse Z Venema (Google)
- Isaac Hepworth (Google)
- Ian Lewis (Google)
- Shripad Nadgowda (Intel)
- Naveen Srinivasan
- Parth Patel (Kusari)

Agenda:

- [Isaac Hepworth] — alignment on precise scope and charter of this group
 - I've been thinking that we should think of "tooling" as naturally encompassing other types of SLSA enablement
 - Might we agree on this group's scope including patterns, examples, and documentation — as well as tools?
 - No need to change the group name; just want to gather thoughts and establish alignment on nominal scope
- [Mike Lieberman] - Discuss what types of SLSA tooling are required
 - Builders following the SLSA requirements
 - SLSA provenance generator libraries and tools
 - SLSA provenance verifier/validator libraries and tools
 - Validate adherence to provenance spec
 - Verify provenance was signed by trusted party
 - SLSA builder compliance checkers
 - Integrations with other tools like scorecard

Notes:

- Isaac suggested docs and walkthroughs
 - Makes sense to start with docs that relate to the tools
 - Might make sense to make some videos or walkthroughs once we have a few tools identified
- Should we include practice in this group?
- Let's include patterns and practices until the adoption group spins up?
 - Naveen - We should only do patterns and practices until adoption group gets spun up
 - Mike - +1
 - Isaac - +1
 - Wietse - +1
 - Shripad - +1
 - Parth - +1
- What are the categories of tools and documentation/training? - Mike L
 - One possible categorization
 - Build and Production
 - Distribution and Discovery

- How do you store and distribute the SLSA metadata
- Verification and Decisioning
- Training and best practices
- Other tooling involved (scorecard, source code control)
- Generic vs. Specific
- Label tool for SLSA level or requirement
- One tool can map to many categories

Current Tooling and Gaps

- Build and Production
 - Service
 - [slsa-github-generator](#) - various builders for GitHub Actions
 - <https://github.blog/2022-04-07-slsa-3-compliance-with-github-actions/>
 - [Gitlab builder](#)
 - [Google Cloud Build](#)
 - Software
 - Tekton Pipelines and Chains
 - [FRSCA](#)
- [Distribution and Discovery](#) - Includes RoT and Key discovery/distribution as related to SLSA
 - [Cosign](#) (OCI)
 - Distribution of provenance metadata and signature via OCI repositories is supported.
 - Relevant commands are cosign attest (upload), and cosign verify-attestation (download)
 - cosign has its own format for this so it's listed here specifically
 - Discovery is not implemented by cosign
 - General distribution of provenance metadata and signatures is not supported for binary artifacts.
 - General distribution and discovery may not be in scope of cosign or other sigstore projects.
 - TUF
 - Distribution of RoT
 - [GUAC](#)
 - This is an attestation discovery and query tool
 - Companion "file"
 - OCI Registry
 - Store signed attestation in the OCI image repo.
 - Package registry

- Npm, PyPI can store it
 - Release assets
 - GH, GitLab, etc
- Verification and Decisioning
 - Container/k8/infra-based:
 - [cosign](#) - cosign verify-attestation supports cue or rego policies
 - [Slsa-verifier](#)
 - [Kyverno](#)
 - Native support for SLSA provenance predicate
 - [Sigstore policy controller](#)
 - [Chainguard Enforce](#)
 - OPA Gatekeeper
 - Multiple gaps
 - No SLSA-specific support yet?
 - [Connaissanceur](#)
 - Verifies image signatures
 - No real SLSA-specific or policy support
 - Kubewarden
 - Supports web assembly so could just run
 - Others
 - [Slsa-verifier](#)
 - [Cosign](#) (signing for sigstore)
 - Package manages : TBD
- Package ecosystem integration with policy (separate from provenance distribution)
 - Npm
 - Pip
 - Gem/Bundler
- Training and best practices and blogs
 - Google Sec Blog:
 - <https://security.googleblog.com/2021/06/introducing-slsa-end-to-end-framework.html>
 - <https://security.googleblog.com/2022/04/how-to-slsa-part-1-basics.html>
 - <https://security.googleblog.com/2022/04/how-to-slsa-part-2-details.html>
 - <https://security.googleblog.com/2022/04/how-to-slsa-part-3-putting-it-all.html>
- Other tooling involved (scorecard, source code control)
 - Github PR
 - Satisfies 2 person PR requirement if setup correctly
 - SBOM
 - Related. We don't have a requirement but there's a lot of

- Witness
- [Project Oak / Transparent Release](#) (unsure if this is Oak specific, but generates SLSA provenance)
- POC code for discovery: <https://github.com/mlieberman85/scq>
- Endemic tools that have gaps
 - Jenkins - Plugins?
 - <https://github.com/slsa-framework/slsa/issues/461>
 - Buildbot