Interview Guide v0.5

Metadata	**Interview**
Created: 2022-03-20	Date:
Updated: 2022-07-29	Participant:
Language: **English**	Interviewer:
	Duration (est.):

Changelog

–	v0.5	| Updated some question wordings
–	v0.4	| Moved "Policy" section between "Usage" and "Experiences" for better flow
–	v0.3	| Added "selection criteria for osc" subsection
		| Renamed "incident" section to "challenges" & moved to later
–	v0.2	| Split "S3 Open Source Components" into "S3 Usage" and "S4 Experiences"
–	v0.1	| Created initial interview guide draft

Intro

– **Thanks**: Thank you very much for offering your valuable time for this interview. We are very grateful for your contribution.
– **Ready**: Are you ready to start the interview?
– **Structure**: First off, I am going to talk about the context and data handling, and if you are okay with everything and have no further questions, only then would we start with the actual interview and the recording.

Context

– **We**: We are researchers at the George Washington University, the German science institution CISPA Helmholtz-Center for Information Security, and the German University of Hannover.
– **Our research**: focuses on the area "Usable Security for Developers". This boils down to "How can we enable and empower developers to write more secure code".
– Open Source components are part of the software supply chain and impact many different software projects. External components include unique challenges, such as code from unknown sources and challenges in trusting external projects.
– **Now**: "How can we support software projects with selecting & including open source components in secure ways?"
– **This interview**: is intended as a start/exploration of internal processes and decisions around open source components and the supply chain.
– For this interview:
    o We are **not judging** the security or privacy of a project, we are just interested in the underlying structures and processes.
    o Projects are often very complex, if you **don't know** the answer, **don't want to answer**, **or are not allowed to answer** a question, feel free to just say "next".
    o We are not just interested in structures, but also your **personal opinions and experiences**.
– **Questions?** Any questions about this interview context so far?

Consent

– **Voluntary**: Your responses in this interview are of course entirely voluntary. You may skip any or all of the questions and can of course leave the interview at any time.
– **Duration**: Duration of the interview depends a bit on the duration of your answers, in our experience 30-45 min, median interview duration so far was about 32 minutes.
– We will **fully de-identify** you and your projects in any publication and only include short quotes.
– We **will send you a preprint** before a potential publication, so you can veto quote usage etc.
– **Recording**: We would like to record this interview so that we can transcribe the answers later
    o The recording will be destroyed when we transcribed the answers (a few days)
    o The transcripts will be destroyed after analysis (a few months)
– **Questions?** Any more questions about data handling or recording?
– "The recording is now on"
– **Restate**: For the recording: "Are you okay with this interview being recorded?"

❐ ● S1 Project & Demographics

[Check project(s) beforehand]

Notes: ______________________________

____

____

____

____

- ☐ S1Q1 **Project:** Can you tell us a bit about the `[project(s)]` you are involved in?
  - ☐ S1Q1.1 **About:** What is the project about? What is its purpose?
  - ☐ S1Q1.2 **Age:** When was the project created?
  - ☐ S1Q1.3 **Developers:** How many regular developers does the project have?
  - ☐ S1Q1.4 **Structure:** How is the team structure for this project?
- ☐ S1Q2 **Project Relation:** How are you related to `[project(s)]`?
  - ☐ S1Q2.1 **Join:** When did you join the project?
  - ☐ S1Q2.2 **Role:** What is your role in the project?
- ☐ S1Q3 **Setup:** Can you tell us a bit more about the general development setup of the project?
  - ☐ S1Q3.1 **Tools**: Is the project using specific (software) tools? (SAST, DAST, Manual, Pentests)
  - ☐ S1Q3.2 **Project Stages:** At what stages of the project? (Only initially, on changes, etc.)
  - ☐ S1Q3.3 **Frequency**: How often are manual security reviews done? Pentesting etc.? Who carries out these reviews? (skills, external/internal person)
  - ☐ S1Q3.4 **Security Roles**: Are there security specific roles in your projects?

❒ ● S2 Usage of Open Source Components (OSCs)

- ☐ S2Q1 **Components**: Are you aware of any OSCs included in your project?
  - ☐ S2Q1.2 **Components**: How/If do the different roles interact with components?
- ☐ S2Q2 **Metrics**: Are you aware of any metrics for selecting those components?
  - ☐ S2Q2.1 **Exclusion**: What are exclusion criteria?
  - ☐ S2Q2.2 **Opinion**: What metrics would you personally like to use?
  - ☐ S2Q2.3 **Awareness**: Are you aware of already prepared metrics such as the *OSSF Scorecards* for repos or *socket.dev* for JS? [see Appendix for explanations what those are]
- ☐ S2Q3 **Supply Chain**: How are those external components pulled/included into the build process?
  - ☐ S2Q3.1 **Inclusion**: What is the process for including components in your project?
  - ☐ S2Q3.2 **Aspects**: What parts of this process include some form of security checks?
  - ☐ S2Q3.3 **Mirrors**: Do you have internal sources/mirrors for the components, instead of public ones? (e.g., internal versions of packages on a local package server, instead of a public one like PyPI)
- ☐ S2Q4 **Infrastructure**: Does additional infrastructure use open source components?
  - ☐ S2Q4.1 **CI/CD**: In the build system?
  - ☐ S2Q4.2 **Other**: Website / Documentation / etc ?
- ☐ S2Q5 **Support**: Does your company contribute back to Open Source Projects (independent of use)?
  - ☐ S2Q5.1 **Selection**: How do you choose which projects you support?
  - ☐ S2Q5.2 **Type**: How does your company contribute back? (pull requests, issues, infrastructure, sponsorship, monetary, …)
  - ☐ S2Q5.3 **Not:** If not, would you like your company to contribute back?

❐ ● S3 Security Policies & Guidance

- ☐ S3Q1 **Security Policies**: If/what are your project's security policies for including external code?
  [IF NOT Policies]: If/what would you like your security policies to contain?
  - ☐ S3Q1.1 **Content**: What specific parts do they cover?
  - ☐ S3Q1.2 **Applicability**: Are the teams aware of these policies? Do these have to be read and acknowledged by developers/admins/managers?
  - ☐ S3Q1.3 **Guidance**: Does this include guidance (e.g., guides or resources for new team members)
- ☐ S3Q2 **Security Incidents:** How are security incidents/issues in components generally handled?
  - ☐ S3Q2.1 **Policy:** By what policy?
  - ☐ S3Q2.2 **Who:** By whom? (specific security team?)
  - ☐ S3Q2.3 **Access:** Discussed in private groups or open within the company?
  - ☐ S3Q2.4 **Process:** Are there “Playbooks” or a specific process for (component) Incident Response and Vulnerability Management? (What do these include?)
  - ☐ S3Q2.5 **History:** How has this process developed?
  - ☐ S3Q2.6 **Disclosure Policies:** What is the (coordinated) disclosure policy of the project?
- ☐ S3Q3 **Documentation:** Does your project provide guides/best practices/hints for including external code (e.g., open source components)?
  - ☐ S3Q3.1 **Opinion:** What are your personal thoughts about including such guides/best practices/hints (e.g, in a Wiki)?

[We are now about halfway done]

❒ ● S4 Experiences with Open Source Components

- ☐ S4Q1 **Components**: Can you tell us a bit about the developer experience of using OSCs in the project?
  - ☐ S4Q1.1 **Setup**: Was it difficult to set up these components?
  - ☐ S4Q1.2 **Documentation**: Are these components documented somewhere? (Could a new hire manage them?)
  - ☐ S4Q1.3 **Customization**: Did you need to customize components for your environment? Did you contribute changes back?
- ☐ S4Q2 **Updating**: How are you keeping open source components up to date?
  - ☐ S4Q2.1 **Responsible**: Who is responsible?
  - ☐ S4Q2.2 **Version**: In what version / release of your `[project(s)]` are OSCs updated?
  - ☐ S4Q2.3 **Checks**: Do you check code changes or changelogs before updating?
  - ☐ S4Q2.4 **Metrics**: How is the initial selection decision for an OSC different to an update decision of the same OSC?
- ☐ S4Q3 **Same**: Would you select the same components again and why?
- ☐ S4Q4 **Releases**: How are releases and updates of your project handled (and how are OSC considered in the process)?
  - ☐ S4Q4.1 **Decision**: How is decided if/when a (security) update is released?
  - ☐ S4Q4.2 **Secured**: What security concepts are considered for releases? Are releases “secured” in any way?
  - ☐ S4Q4.3 **Update System**: Are security updates made automatically? How does the update system work?
  - ☐ S4Q4.4 **Deprecation**: Do you publish information about deprecated / insecure versions?
  - ☐ S4Q4.5 **Dependencies**: How do you handle dependencies to external components when you publish a release/update?

❐ ● S5 Challenges & Incidents

[Mention "protestware node-ipc" incident: If participant is aware continue, else introduce (see Appendix)]

- ☐ S5Q1 **Opinion:** What is your personal opinion of this incident (in terms of supply chain trust)?
    - ☐ S5Q1.2 **Handling**: How would you react, if your project depends on this (node-ipc) dependency?
- ☐ S5Q2 **Trust Strategy**: In general, what is/would be your `[project(s)]` strategy to deal with components that become untrustworthy?
    - ☐ S5Q2.1 **Identifying**: What does the process for identifying untrustworthy components look like?
    - ☐ S5Q2.2 **Excluding**: If applicable, can you explain the process for excluding components from the project more in-depth?
- ☐ S5Q3 **Challenges:** Can you remember any supply-chain related security challenges that `[project]` faced in the past?
  Nudges:
    - Malicious / compromised dependency
    - Security vulnerability included from a dependency
- ☐ S5Q4 **Inconveniences:** Aside from major challenges, were you inconvenienced by components in the past?
  Nudges:
    - Dependency no longer maintained
    - Sudden changes to components' API
    - Changes how a component was distributed

[We are now at the last question block]

❐ ● S6 Problems and Improvements

- ☐ S6Q1 **Opinions**: In your experience, what is the perceived security of your project? Both by internal actors (like the team) and external actors (like the client or the public).
- ☐ S6Q2 **Improvements**: Assuming no limitations whatsoever (e.g., monetary or in terms of developer-hours), how would you personally like to improve supply-chain security of your project?
    - ☐ S6Q2.1 **Problems**: Where do you see problems in the current system?
    - ☐ S6Q2.2 **Why Exist:** What do you think are the reasons for the current (trust) system to be the way it is?
    - ☐ S6Q2.3 **Improvements:** What would you like to improve?

Outro

- **Anything else:** Is there something that we did not cover during the interview but you would like to talk about on the recording?
- **[IF second interviewer]**: Wait if the second interviewer has any further questions.
- “The recording is now off”
- Thank the participant again for offering their valuable time
- "We will later be in contact for a preprint for changes and veto"
- **[IF Voucher]:** Ask if the want/can be compensated for the interview in the form of an Amazon voucher
    - Ask for the email they want their voucher assigned to (likely their Amazon email). Write the email down!
    - Stress that it might take some time (weeks) for the University to process the charges
    - Stress that non-american or german accounts might not work, we will then email them to figure out another way to send them the money

Debrief

- Easy debrief starter is to ask the participant how they felt about the interview
- Do you have any recommendations for other interesting companies or stakeholders we should invite for an interview?

Appendix

Protestware-Incident (node-ipc)

In early March 2022, the JavaScript node-ipc library (dependency of, e.g., @vue/cli) was updated by the maintainer to include potentially malicious code: depending on the version when the node-ipc dependency was pulled in, the code scans the host's IP address.
If the IP address matched lists of Russian or Belarussian IPs, the package would go on to replace all of the user's system files with a heart emoji.
In other versions, the package always displayed a message of support for Ukraine.

OSSF Scorecard

Scorecard (https://github.com/ossf/scorecard) is a tool by the Open Source Security Foundation that allows users to judge the safety of dependencies automatically based on several metrics. The tool allows assessing several criteria, like project activity, contributors, known vulnerabilities, static analysis, etc., on a scale from 0 to 10. Overall, the project aims to help automate trust decisions, and can be used as a CLI tool or GitHub Action.

Socket.dev

Socket.dev is a tool to detect and block (ongoing) software supply chain attacks in JavaScript software and the npm ecosystem, e.g., by monitoring dependency changes and analyzing their behavior. It is available as a GitHub application.

Open Source

We don't care too much about the ideological idea of open source here, so a fitting example definition would be:

"Software components publicly available, e.g., on GitHub or language respective package platforms like pypi or npm. Using these components generally requires the user (our participant's company) to rely on code contributions from a number of unknown and untrusted maintainers or developers"