Interview Guide v2.5 GER

Metadata
Created: 2021-08-10
Updated: 2021-10-20
Language: **German**

Interview
Date:
Participant:
Interviewer:
Duration (est.):

Changelog

- v2.5 | Added some additional instructions for better replicability.
- v2.4 | Moved changelog to interview guide.
- v2.3 | Extended intro slightly with some additional motivation.
- v2.2 | Extended metadata entries with interview data.
- v2.1 | Extended intro with full list of institutions.
- v2.0 | Added German version.
- v1.2 | Added preamble and outro.
- v1.1 | Moved interview guide to Nextcloud.
- v1.0 | First interview-ready version.

[Allgemeine Begrüßung]

Intro

- Du vs. Sie [Dieser Guide nutzt von hier an “Du”]
- **Thanks**: Wir bedanken uns vielmals für deine bisher aufgewendete Zeit und dem Interesse an unserem Interview.
- **Ready**: Bist du bereit, mit dem Interview zu beginnen?
- **Structure**: Zunächst werde ich was zum Hintergrund des Interviews und der Verwendung von Daten sagen, und nur wenn du mit allem zustimmst, werden wir dann mit dem eigentlichen Interview beginnen.

Context

- **We**: Wir sind Wissenschaftler am CISPA Helmholtz Center (kurze Einführung)
- **Our research**: Unsere Forschung ist im Bereich Nutzbare IT Sicherheit für Entwickler (kurze Übersicht)
- **In the past**: Open Code Projekts als Datenquelle.
- Open Code Projekte wichtiger Baustein, aber auch besondere Herausforderungen.
- **Now**: “Wie können wir im speziellen Open Source Contributorn helfen, ihre Open Code Projekte sicherer zu machen.”
- Diese Interview als Basis: Explorative Projekte und Abläufe über Dateien/Repos hinaus anschauen (insbesondere weil viele Abläufe nicht direkt im Repo erkennbar sind).
- Im Speziellen für das Interview heißt das:
    - Dies ist keine Bewertung von Sicherheitsabläufen oder ähnlichem, wir sind lediglich an den Strukturen interessiert.
    - Projekte können sehr komplex sein, wenn keine Antwort bekannt: “weiter”.
    - Nicht nur stumpfe Abfrage von Daten, wir haben die Form eines Interviews gewählt, weil wir auch an persönlichen Meinungen und Einschätzungen interessiert sind.
- **Questions?** Fragen zum Interview-Hintergrund?

Consent

- **Voluntary**: Dieses Interview ist komplett freiwillig, zu jedem Zeitpunkt kann das Interview beendet oder Fragen übersprungen werden
- **Duration**: Länge des Interviews hängt von der Ausführlichkeit der Antworten ab, unserer bisherigen Erfahrung nach zwischen 30 und 45 Minuten.
- Wir anonymisieren dich und alle Projekte, sowie verwenden wenn überhaupt nur kurze Zitate
- Bevor wir irgendwas veröffentlichen, senden wir dir/Ihnen eine Kopie zum Vetoen.
- **Recording**: Interview Aufnahme
    - Diese Aufnahme wird nach dem transkribieren zerstört (ein paar Tage)
    - Die Transkripte werden nach der Auswertung zerstört (ein paar Monate)
- **Questions?** Fragen zur Datenverarbeitung oder Aufnahme?
- “Die Aufnahme läuft jetzt”
- **Restate** consent question

❒ ● S1 Project & Demographics

[Check project(s) beforehand]

Notes:_____

_____

_____

_____

- ▢ S1Q1 **Project:** Kannst du uns ein wenig zu [project(s)] erzählen?
  Follow-Ups:
  - ▢ S1Q1.1 **About:** Worum geht es bei dem Projekt? Was ist der Zweck des Projekts?
  - ▢ S1Q1.2 **Age:** Wann ist das Projekt entstanden?
  - ▢ S1Q1.3 **Contributors:** Wie viele reguläre Contributors hat das Projekt?
  - ▢ S1Q1.4 **Connection:** Wie kennen sich die Contributors? (Virtuell, Persönlich)
  - ▢ S1Q1.5 **Distribution:** Wie verteilen sich die Contributors geographisch?
- ▢ S1Q2 **Project Relation:** Was ist oder war dein Zusammenhang zu [project]?
  Follow-Ups:
  - ▢ S1Q2.1 **Join:** Wann bist du zum Projekt gestoßen?
  - ▢ S1Q2.2 **Role:** Was war deine Aufgabe in dem Projekt?

❒ ● S2 Incidents

[Mention “hypocrite commits” incident: If participant is aware continue, else introduce (see Appendix)]

- ▢ S2Q1 **Opinion:** Was hältst du von diesem Vorfall?
- ▢ S2Q2 **Challenges:** Kannst du dich an irgendwelche Security-Herausforderungen bei deinem Projekt erinnern?
  Nudges:
  - boshafte Committers/commits,
  - Sicherheits-Probleme im Repository (software/provider)
  - Softwarewerkezeug-Kette, etc.?

❒ ● S3 Guidance

- ▢ S3Q1 **Guidance**: Gibt es Leitfäden/Best Practices/Hinweise für Contributors, Maintainers, usw.?

- [IF NOT Guidance]:
  - Was ist deine Meinung dazu, Leitfäden/Best Practices/Hinweise für Contributors, Maintainers, usw. bereitzustellen?

Follow-Ups:

- ▢ S3Q1.1 **Infrastructure:** Hat das Projekt Richtlinien für die Configuration und Ausführung der Infrastruktur? (Cloud, VCS, usw.)
- ▢ S3Q1.2 **Languages:** Hat das Projekt Sicherheits- oder Stil-Richtlinien für die verwendeten Programmiersprachen? Was decken die Richtlinien ab?
- ▢ S3Q1.3 **Cryptography**: Falls Kryptographie verwendet wird: Hat das Projekt Richtlinien bezüglich der Nutzung von Kryptographie? (verbotene Funktionen)

❒ ● S4 Security Policies

- ▢ S4Q1 **Security Policies** Was enthalten die Sicherheits-Richtlinien?

- [IF NOT Policies]:
  - Was sollten deiner Meinung nach die Richtlinien enthalten?

Follow-Ups:

- ▢ S4Q1.1 **Content**: Welche Bereiche decken sie ab?
- ▢ S4Q1.2 **Applicability**: Müssen sie von Commiters/Contributers gelesen und anerkannt werden?

- ▢ S4Q2 **Disclosure Policies** Was ist die (coordinated) disclosure policy des Projekts?
- ▢ S4Q3 **Security Incidents:** Wie wird mit Sicherheits-Vorfällen verfahren?

Follow-Ups:

- ▢ S4Q3.1 **Policy:** Nach welcher Richtlinien?
- ▢ S4Q3.2 **Who:** Von wem? (Sicherheits-Team?)
- ▢ S4Q3.3 **Access:** Privat/Öffentlich?
- ▢ S4Q3.4 **Access:** Gibt es “Playbooks” für die Reaktion auf Vorfälle oder Schwachstellen? (Was steht drin?)
- ▢ S4Q3.5 **History:** Wie ist dieser Prozess entstanden?

- ▢ S4Q4 **Security Testing / Reviews**: Welche Verfahren für Sicherheits-Tests gibt es?

Follow-Ups:

- ▢ S4Q4.1 **Aspects**: Welche Aspekte decken diese Tests ab, und wie?
- ▢ S4Q4.2 **Tools**: Nutzt das Projekt spezielle (Software-)Tools? (SAST, DAST, Pentests)
- ▢ S4Q4.3 **Project Stages:** Bei welchen Phasen des Projekts? (Am Anfang, für Änderungen, usw.)
- ▢ S4Q4.4 **Frequency**: Wie häufig werden manuelle Sicherheits-Bewertungen durchgeführt? Pentesting etc.? Wer macht die Reviews? (Fähigkeiten, Extern/Interne Person)
- ▢ S4Q4.5 **Threat Modeling**: Wird eine Art von Threat Modelling verwendet?

❒ ● S5 Project Structure

- ▢ S5Q1 **Repository**: Wie sieht die Repository-Struktur aus? (Filesystem, Stufen, CI, usw.)
  Follow-Ups:
  - ▢ S5Q1.1 **Stages**: Welche Projekt-Stages gibt es? (Code → Commit → PRs → Review → CI for tests and Build → Deployment, …)
  - ▢ S5Q1.2 **Control**: Wer kontrolliert welche Stufe?
  - ▢ S5Q1.3 **Main**: Wie sind die Branches aufgesetzt? Ist es möglich, direkt auf den main Branch zu pushen?
  - ▢ S5Q1.4 **Systems**:Wie sind Build- und Deployment-System gesichert? Wer hat Zugriff/Kontrolle?
  - ▢ S5Q1.5 **PRs**: Wie wird mit neuen Pull Requests verfahren?
  - ▢ S5Q1.6 **Signed Commits**: Sind Commits signiert? (Braucht PGP-Schlüssel, wie wird denen vertraut?)
  - ▢ S5Q1.7 **Secret Management**: Nutzt das Projekt eine Art von Secret Management System?
  - ▢ S5Q1.8 **Access**: Wer hat Zugriff auf diese Systeme?
- ▢ S5Q2 **Supply Chain**: Wie sieht das Setup for die Supply Chain (bspw. Libraries und andere Abhängigkeiten) aus?
  Follow-Ups:
  - ▢ S5Q2.1 **Criteria**: Welchen Kriterien werden bei der Auswahl von externen Libraries und Dependencies berücksichtigt?
  - ▢ S5Q2.2 **Checks**: Nach welchen Prozessen wird deren Sicherheit und Vertraubarkeit etabliert?
  - ▢ S5Q2.3 **Vulnerabilities**:Wie wird überprüft, ob eine Dependency Sicherheitslücken hat? (SAST, code reviews, prüfen von Open Source Projekten und deren Contributors, usw.)
  - ▢ S5Q2.4 **Signed**: Müssen Libraries kryptographisch signiert sein?
  - ▢ S5Q2.5 **Private Packages**: Nutzt das Projekt ein privates Package Repository mit geprüften und sicheren Dependencies?
- ▢ S5Q3 **Infrastructure**: Hat das Projekt zusätzliche Infrastruktur wie eine Webseite oder Tools zur Kommunikation?
  Follow-Ups:
  - ▢ S5Q3.1 **Access**: Wer hat Zugriff auf die zusätzliche Infrastruktur? Selben Maintainer wie die gesamte Infrastruktur?

[We are now about halfway done]

❒ 🟠 S6 Release and Updates

- ▢ S6Q1 **Releases**: Wie werden Releases und Updates veröffentlicht?

  Follow-Ups:
  - ▢ S6Q1.1 **Decision**: Wie wird entschieden, ob/wann ein Sicherheits-Update veröffentlicht wird?
  - ▢ S6Q1.2 **Secured**: Welche Sicherheitskonzepte werden für Releases berücksichtigt? Sind Releases in irgendeiner Weise “gesichert”?
  - ▢ S6Q1.3 **Update System**: Werden Security-Updates automatisch aufgespielt? Wie funktioniert das Update-System?
  - ▢ S6Q1.4 **Deprecation**: Werden Information über deprecated / unsichere Versionen veröffentlicht?

❒ 🟡 S7 Roles and Responsibilities

- ▢ S7Q1 **Contributors**: Kannst du uns was zur Maintainer / Contributor Hierarchy im Projekt erzählen?

  Follow-Ups:
  - ▢ S7Q1.1 **Security Roles**: Gibt es sicherheits-spezifische Rollen im Projekt?
  - ▢ S7Q1.2 **Roles known**: Sind diese Rollen / Gruppen allgemein bekannt?

❒ 🟢 S8 Trusting Contributors

S8Q1 **Trust**: Kannst du uns was zum Vertrauensmodell des Projektes erzählen?

Follow-Ups:
- ▢ S8Q1.1 **Establish**:Wie wird das Vertrauen in neue Committer etabliert?
- ▢ S8Q1.2 **Identity**: Gibt es eine Art von Identitätsüberprüfung? (bspw. Für Kollegen, auf Konferenzen, usw.)
- ▢ S8Q1.3 **Authentication**: Werden Committer authenticated? (How?)
- ▢ S8Q1.4 **Access Control**: Wird eine Art von Access Control verwendet?
- ▢ S8Q1.5 **Trusted**: Wie können neue Committer vertraute Mitglieder des Projektes werden?

- ▢ S8Q2 **License** Gibt es eine Contributor License Agreement?
  - ▢ **Yes**: Was enthält sie?
  - ▢ **No**: Weißt du, warum das Projekt keine hat?
- ▢ S8Q3 **Public:** Verwaltet das Projekt eine öffentliche Liste von Contributors und deren Beiträgen?

❒ ● S9 Untrustworthy Contributors

- ▢ S9Q1 **Trust Incidents**: Weißt du von Contributors, die sich als nicht vertrauenswürdig herausgestellt haben?
- [IF NO]:
  - Angenommen es gäbe einen nicht vertrauenswürdigen Contributor ...

  Follow-Ups:
  - ▢ S9Q1.1 **Approach**: Wie ist das Projekt mit dieser Situation umgegangen?
  - ▢ S9Q1.2 **Excluding**: Falls zutreffend, wie läuft der Prozess für einen Projekt-Ausschluss von Contributors ab?
  - ▢ S9Q1.3 **Identifying**: Wie sieht der Prozess zum Identifizieren von nicht vertrauenswürdigen Contributors aus?
- ▢ S9Q2 **Trust Strategy**: Wie sieht die Projekt-Strategie für nicht vertrauenswürdige Contributors aus?

  Follow-Ups:
  - ▢ S9Q2.1 **Who**: Wer trifft die Entscheidung? (BDFL, committee, maintainers)
  - ▢ S9Q2.2 **Playbook**: Gibt es ein Playbook / definierten Prozess?
  - ▢ S9Q2.3 **Circumstances**: Unter welchen Umständen werden nicht vertrauenswürdige Contributors von zukunftigen Beiträgen ausgeschlossen?
  - ▢ S9Q2.4 **Process**: Wie sieht der Ausschlussprozess aus?
  - ▢ S9Q2.5 **Investigation**: Hat das Projekt einen definierten Ablauf, um möglicherweise unsichere Beiträge zu untersuchen?
- ▢ S9Q3 **Removal**: Hat sich das Projekt entschieden, deren Beiträge zu entfernen?
  - ▢ S9Q3.1 **Decision**: Wie wurde die Entscheidung getroffen?
  - ▢ S9Q3.2 **Process**: Wie sah der Ablauf der Entfernung aus?

[We are now at the last question block]

❒ ● S10 Problems and Improvements

- ▢ S10Q1 **Reputation**: Was ist, deiner persönlichen Meinung nach, der Ruf des Projekts im Kontext von Sicherheit und Vertrauen?

  Follow-Ups:
  - ▢ S10Q1.1 **Internal**: Interner Ruf.
  - ▢ S10Q1.2 **External**: Externer Ruf.
- ▢ S10Q2 **Improvements Project**: Angenommen es gäbe keine Einschränkungen (Geld, Arbeitszeit), wie würdest du persönlich am liebsten die Sicherheit / das Vertrauen des Projektes verbessern?

  Follow-Ups:
  - ▢ S10Q2.1 **Problems**: Wo siehst du die Probleme im aktuellen System?
  - ▢ S10Q2.2 **Why Exist:** Was denkst du sind die Gründe für das aktuelle System
  - ▢ S10Q2.3 **Improvements:** Was würdest du verbessern?

Outro

- “The recording is now off”
- Thank the participant again for their valuable time

Debrief

- Gibt es etwas, das wir während des Interviews nicht angesprochen haben, über das Du aber gerne gesprochen hättest?
- [Falls angemessen:] Kennst du noch weitere Personen, die für ein solches Interview potentielle Teilnehmende sein könnten?

Appendix

Incident Hypocrite Commits

In April 2021, after receiving “poor quality patches” Linux kernel developer Greg Kroah-Hartman reverted 68 patches submitted by University of Minnesota email addresses and announced that all further patches coming from University of Minnesota addresses should be summarily rejected by default.

This incident actually started in August 2020, when a number of “bad faith” patches were sent to Linux kernel developers by University of Minnesota researchers under false identities. These patches were part of an ongoing work studying the feasibility of introducing vulnerabilities into open source software projects through minor patches (“hypocrite commits”).