Verantwortung und Sicherheit

Verhaltenskodex, Überprüfungsmechanismen, Transparenz, Updates, gesellschaftlicher Mehrwert

- Linksammlung Verantwortung und Sicherheit
- gget - Ressourcen zu Code-Dokumentation, Contributing Guides und Verhaltenskodizes
- Manipulation durch Mitwirkung: Sicherheitsrisiko Social-Engineering im Open-Source-Kontext

□□ Linksammlung Verantwortung und Sicherheit

Sicherheit

- Das Red Team Lab des Open Technology Fund bietet Dienstleistungen von Drittanbietern und direkte Mittel zur Stärkung der Sicherheit von Open-Source-Software an. Das Red Team Lab kann euch zum Beispiel bei einem Audit eurer Software helfen oder euch dabei unterstützen, die Best Practices für Sicherheit in eurem Projekt voranzutreiben. (Kostenlos, auf Bewerbung)
  Für: Audits, Security Best Practices, Bug Bounties, Validierung von Security Modellen
  Wo: https://www.opentech.fund/labs/red-team-lab/
- Vom Prototyp bis zur Produktion befasst sich das Consulting-Unternehmen Least Authority mit der Identifizierung von Sicherheitsproblemen und der Förderung von Best Practices.
  Für: Sicherheitsaudits, Design-Reviews, Begleitung von Projekten im Sicherheitsbereich
  Wo: https://leastauthority.com/
- Best Practices Program der Core Infrastructure Initiative.
  Für: Sicherheitslabel basierend auf Kriterien, die mit der FOSS-Community erarbeitet wurden
  Wo: https://www.coreinfrastructure.org/programs/best-practices-program/
- Der Open Source Technology Improvement Fund (OSTIF) bietet finanzielle Unterstützung für Security Audits und Reviews an: https://ostif.org/

Verantwortung

- GitHub, *Dos and don’ts when sunsetting open source projects*: https://github.blog/open-source/maintainers/dos-and-donts-when-sunsetting-open-source-projects/
- CHAOSS, *Practitioner Guide: Getting Started with Sunsetting an Open Source Project*: https://chaoss.community/practitioner-guide-sunset/

□□ gget - Ressourcen zu Code Dokumentation, Contributing Guides und Verhaltenskodizes

Wer möchte, dass andere sinnvoll zum eigenen Code beitragen und die Software verbessern, tut gut daran, potenzielle Contributors an die Hand zu nehmen.

Mit ihrem Projekt gget, einer Software zum Handling großer genomischer Datenbanken, tut Laura Luebbert genau das. In diesem Beitrag zeigen wir euch, welche Maßnahmen im Fall von gget ergriffen wurden – und auch für viele andere FOSS-Projekte von Interesse sind!

Contributing Guide

In einem Contributing Guide wird zusammengefasst, was Beitragende beachten sollen: Von der Erstellung eines Issues oder Pull Requests bis zum Merge sind bei gget alle Schritte abgedeckt.

Der Guide beinhaltet auch Hinweise zur Überprüfung des eigenen Codes vor dem Pull Request, sogar inklusive einer Vorgabe für die Formatierung.

Übrigens: Auch bei GitHub finden sich Tipps zum Erstellen von Guidelines für Contributors.

Code of Conduct

Damit Beitragende nicht nur wissen, was die technischen Ansprüche sind, sondern auch welche Werte in der Community von Bedeutung sind, gibt es bei gget einen Code of Conduct, den alle Personen befolgen sollen, die sich dem Projekt anschließen, mit dessen Mitgliedern interagieren oder in seinem Namen kommunizieren.

Er beinhaltet neben den Werten der Community auch konkrete Beispiele für unangebrachtes Verhalten, verdeutlicht, welche Folgen dieses mit sich bringt und wer in der Verantwortung steht, die Regeln des Codes of Conduct durchzusetzen.

Code-Dokumentation

Ein weiterer Punkt, auf den viel Wert gelegt wurde, ist die Code-Dokumentation. In ihrer Doktorarbeit hat Laura unter anderem ein Kapitel für das Schreiben leicht nutzbarer Software veröffentlicht. Zum Thema Dokumentation finden sich darin diese Hinweise (mit Bezug auf gget):

“Erstellung einer verständlichen und schlüssigen Dokumentation auf verschiedenen Wegen:
- Bereitstellung von Funktionsbeschreibungen in Python/R/etc.
- Hinzufügen von Hilfemöglichkeiten (-h / --help) zu Shell-Skripten
- Schreiben eines GitHub README, das die Dokumentation enthält oder auf sie verlinkt
- Schreiben einer ausführlichen Dokumentation, die eine Installationsanleitung und idealerweise eine "Schnellstart"-Anleitung [...] enthält
- Verfassen von Tutorials für verschiedene Anwendungsfälle der Software, idealerweise in Form von sofort ausführbaren Google Colab-Notebooks, und Verlinkung in der Dokumentation
- Um die Zugänglichkeit zu verbessern, Alt-Text für Bilder verwenden und Dokumentation in verschiedenen Sprachen, z. B. Englisch und Spanisch, bereitstellen”

Die kompletten Guidelines finden sich hier (S. 7-8).

Werft gern einen Blick in die vielfältigen Ressourcen von gget und lasst euch für eure eigenen Projekte inspirieren!

□□ Manipulation durch Mitwirkung: Sicherheitsrisiko Social-Engineering im Open-Source-Kontext

Open-Source-Bausteine sind zentrale Pfeiler digitaler Infrastruktur und rücken deshalb auch im Zuge der Souveränitätsdebatte stärker in den Fokus. Als grundlegende Infrastrukturkomponente ist ihre Sicherheit zentral. Open-Source-Projekte haben in Bezug auf ihre Sicherheitslage mindestens eine Besonderheit: Die kollaborative Entwicklung ist einerseits eine Stärke, kann andererseits durch Social-Engineering-Angriffe aber auch als Angriffsfläche ausgenutzt werden. Unser Fokusbericht 01 befasst sich deshalb mit Maßnahmen zur Risokoreduktion von Social-Engineering-Angriffen und den damit verbundenen Herausforderungen.

Social-Engineering als Angriffsmuster

Social-Engineering wird zunehmend als Angriffsmuster gegen digitale Infrastruktur eingesetzt. Dabei werden statt technischer Schwachstellen, menschliche Eigenschaften ausgenutzt, also Menschen getäuscht, um Zugang zu Informationen oder Systemen zu erlangen. Phishing-Mails oder Anrufe von vermeintlichen Enkeln sind Beispiele für Social-Engineering aus dem privaten Raum. Zunehmend treten ähnliche Manipulationspraxen auch in der kollaborativen Softwareentwicklung auf Plattformen wie GitHub, GitLab oder Codeberg auf. Angreifer*innen versuchen dort durch durchaus langwierige und komplexe Täuschungsmanöver Projektzugänge zu erschleichen, um Schadcode einzuschleusen. Spätestens seit dem Angriff auf xz-utils, eine zentrale FOSS-Bibliothek, im Jahr 2024 steht dieses Thema auf der Sicherheitsagenda für Open-Source-Infrastruktur. Unser neuester Fokusbericht befasst sich deshalb mit dem Umgang mit Social-Engineering-Angriffen und den damit verbundenen Herausforderungen. Den vollständigen Bericht findet ihr hier.

Maßnahmen zum Schutz vor Social-Engineering-Angriffen

Für den Bericht haben wir Gegenmaßnahmen auf unterschiedlichen Ebenen zusammengetragen. Auf Projektebene stehen klare Strukturen, Transparenz und technische Sicherheitsmaßnahmen wie Zugriffskontrollen und Code-Signaturen im Fokus, die durch die sicherheitsorientierte Gestaltung von Entwicklungsplattformen ergänzt werden. Auf der übergeordneten Policy-Ebene spielen Gesetzgebung, Kapazitätsaufbau und Förderprogramme eine zentrale Rolle in der Absicherung von FOSS-Infrastruktur.

Stimmen aus der Community

Um diese Maßnahmen im Kontext des Prototype Fund zu verorten und Stimmen aus der FOSS-Community einzubinden, haben wir Interviews mit durch den Prototype Fund geförderten Projektverantwortlichen geführt und einen Workshop auf einem Chaos-Computer-Club-Event durchgeführt. Neben technischen Maßnahmen wurden soziale Faktoren wie Vertrauensaufbau durch persönliche Treffen, menschliche Fehlbarkeit und unterschiedliche Betroffenengruppen entlang der Softwarelieferkette hervorgehoben. Zugleich wurden Ressourcenknappheit und der Bedarf an stärkerem politischen und wirtschaftlichen Rückhalt als zentrale Herausforderungen benannt.

Abwägungen für FOSS-Ökosysteme

Aus unterschiedlichen Perspektiven zeigt sich, dass Social-Engineering-Angriffe in einem komplexen, heterogenen und vor allem sozialen FOSS-Ökosystem stattfinden. Während viele Maßnahmen einen formalisierten und prozeduralen Ansatz verfolgen, wird insbesondere in Interviews und im Workshop deutlich, dass Sicherheitsmaßnahmen vielversprechend sind, wenn sie die soziale Beschaffenheit des FOSS-Ökosystems und die Bedürfnisse der Beitragenden berücksichtigen. Darüber hinaus erhöht ein gesundes Ökosystem auch die Resilienz von Projekten, die in ebendiesen entstehen. Veranstaltungen in Präsenz können beispielsweise den persönlichen Vertrauensaufbau zwischen Beitragenden und Maintainer*innen stärken. Die ohnehin zentrale Finanzierungsfrage im FOSS-Bereich verschärft sich dabei zusätzlich angesichts strukturell begünstigter und häufig besser finanzierten Angreifer*innen.

Nach prominenten Angriffen und damit steigender Sensibilität für Social-Engineering-Angriffe haben zwar einige Akteur*innen begonnen Gegenmaßnahmen zu ergreifen, da bei Social-Engineering der kollaborative Kern von FOSS-Projekten angegriffen wird, ist der Umgang mit dieser Bedrohung jedoch ein Balanceakt zwischen Offenheit und Sicherheit. Dieser erfordert eine hohe Sensibilität für die Besonderheiten von FOSS-Projekten, ausreichende Ressourcen und vor allem ausreichendes Bewusstsein und politischen Willen, digitale Open-Source-Infrastruktur zu sichern.

Zum Fokusbericht

Zum Fokusbericht