07.05.2021

Wie Apps und Programmcode überprüft werden

Anlass

Diskussionen über Struktur, Sicherheit und Datenschutz von Apps und Programmen sind in der IT seit Jahrzehnten an der Tagesordnung. Insbesondere durch die Corona-Pandemie ist dieses Thema in letzter Zeit aber auch gesamtgesellschaftlich breiter diskutiert worden. Anwendungen wie die Corona-Warn-App, die Luca-App und auch der geplante digitale Impfnachweis wurden und werden in Fachkreisen, aber auch in der breiten Öffentlichkeit, viel thematisiert.

Die Corona-Warn-App war erst als Ansatz mit einer zentralen Dateninfrastruktur geplant, bevor auf einen dezentralen Ansatz gewechselt wurde, nachdem klar war, dass Google und Apple nur diesen Weg mit ihrer Schnittstelle unterstützen. Luca sieht sich immer noch starker Kritik ausgesetzt. Und auch beim digitalen Impfnachweis gibt es noch Fragen zu Sicherheit und Datenschutz.

Dabei spielt immer die Frage eine große Rolle, inwiefern Systembeschreibung und Quellcode der jeweiligen Anwendungen öffentlich einsehbar sind – also Open Source sind. Die Corona-Warn-App wurde für die transparente Veröffentlichung gelobt und die nationalen Anwendungen für den digitalen Impfnachweis sind von Vorneherein als Open Source Projekte geplant – dazu sollen auch bald weitere Informationen erscheinen. Im Gegensatz dazu gab es bei Luca von der IT-Sicherheits-Community Kritik für die späte, unvollständige und zaghafte Veröffentlichung des Quellcodes.

Weshalb es so wichtig ist, dass solche digitalen Anwendungen von unabhängigen Stellen geprüft werden können; was Expertinnen und Experten tun, wenn sie den Quellcode und eine App generell begutachten; wie lange es dauert, bis es nach Veröffentlichung des Codes verlässliche Aussagen dazu gibt und was Journalistinnen und Journalisten bei der Berichterstattung über solche Themen beachten sollen, darüber liefert dieses Fact Sheet einen kurzen Überblick.

Übersicht

Warum ist es wichtig, Apps zu überprüfen? Was versprechen sich Forschende davon?

- Hauptziel: mögliche Fehler finden – möglichst bevor ein System eingeführt bzw. gekauft wurde
- So kann eine unabhängige Instanz überprüfen, ob die App wirklich das tut, was versprochen wurde und kontrollieren, was wirklich implementiert wurde.
- Insbesondere wichtig, wenn es um sensible Daten geht (private Informationen, Standortdaten, Gesundheitsdaten) und wenn es sich um Anwendungen handelt, deren Nutzung verpflichtend ist oder quasi verpflichtend sein könnte
- Das gibt den Entwicklerinnen und Entwicklern die Gelegenheit, die Fehler rechtzeitig zu beheben.
- Da kein System zu einhundert Prozent sicher ist, kann man oft nur mögliche Schwachstellen und Angriffswege frühzeitig identifizieren, die Fehler beheben und so die Angriffsmöglichkeiten minimieren.
- Der Ansatz, Quellcode und genauere Informationen zur App zurückzuhalten, um Angreifern so wenig Informationen wie möglich zu geben (Stichwort „Security by Obscurity") ist in der IT-Szene eher verpönt, da so auch mögliche Fehler von der Community schwerer zu finden und somit schwerer zu beheben sind.
- Daher haben große Firmen auch oft sogenannte „Bug Bounties", im Zuge derer sie „Kopfgelder" für gefundene und an sie gemeldete Fehler in ihrer Software bieten.

Was wird bei einer Begutachtung einer App genau betrachtet?

Wichtig: verschiedene Aspekte, die betrachtet und analysiert werden müssen:

- Systembeschreibung der App selbst
- Code und Funktionsweise der App selbst (Quellcode)
- Zusammenspiel der Software mit der dazugehörigen IT-Infrastruktur
- eventuell Schnittstellen zu und Funktionsweise von angebundenen Strukturen (beispielsweise die Google/Apple-Schnittstelle der Corona-Warn-App)

1. Schritt: Betrachtung der Systembeschreibung

- In der Systembeschreibung wird klargestellt, was die Software macht. Die Zielsetzung des Systems, erwartete Funktionen und erwartete Sicherheitseigenschaften werden präzise beschrieben.
- Mögliche Informationen, die Begutachtende aus der Systembeschreibung entnehmen können:
  - Welche Algorithmen werden benutzt,
  - wo liegen Daten und wie werden sie verarbeitet,
  - wie ist die Verschlüsselung angedacht,
  - was sind die Sicherheitsanforderungen und welche Teile des Systems sind sicherheitskritisch?
- Begutachtende können so das Ziel und die eingesetzten Methoden der Software verstehen und mögliche grundlegende Fehler erkennen.

- Fehler oder Probleme, die bei der Betrachtung der Systembeschreibung auffallen, sind meist systeminhärent und können nicht durch einen einfachen Patch der Software behoben werden (Stichwort Designfehler bzw. Fehler in der Spezifikation).
- Wie genau die Details im Code umgesetzt sind, ist durch die Systembeschreibung aber noch nicht klar.
- Wenn keine Systembeschreibung vorliegt oder diese lückenhaft ist, kann diese im Prinzip aus dem Quellcode rekonstruiert werden, das ist jedoch sehr aufwendig.

2. Schritt: Betrachtung des Quellcodes

- Der Quellcode ist der für Menschen lesbare und in einer Programmiersprache geschriebene Code, auf dem das entsprechende Programm basiert.
- Durch die Betrachtung des Quellcodes können Begutachtende verstehen, wie das System genau funktioniert und wie die in der Systembeschreibung festgehaltenen Funktionen umgesetzt werden.
- Fehler und Probleme, die bei der Betrachtung klar werden, können oft mit einem Patch behoben werden.
- Die häufigste Ursache für Schwachstellen sind Programmierfehler, diese können durch eine Analyse des Quellcodes entdeckt werden.
- Mögliche weitere Probleme sind z.B. die Verwendung von veralteten und unsicheren Standards und Implementierungen.

3. Schritt: Systeme und Infrastruktur in der Praxis systematisch auf Fehler und Schwachstellen prüfen

Die genauere Systemprüfung kann durch einen Penetrationstest (auch: Penetration Test oder Pentest) oder IT-Sicherheitsaudit erfolgen. Diese Prüfung wird meist von auf solche Tests spezialisierten Firmen durchgeführt, kann aber auch von unabhängigen Gruppen wie Universitäten oder Vereinen wie dem Chaos Computer Club durchgeführt werden.

- Bei Penetrationstests werden Systeme nach Schwachstellen durchsucht und zum Test auf verschiedene Weisen angegriffen – im Grunde ein professionell durchgeführter simulierter Hackerangriff. Diese Tests sind stark auf die jeweilige Situation zugeschnitten und folgen keinem Standardschema.
- Unterschiede bei der Prüfung, je nach Situation und Auftrag für die Prüfenden:
  - Was wird überprüft? Gesamtsystem oder einzelne Teile des Systems?
  - Wie genau ist die Prüfung? Eher oberflächlich oder tiefgehend?
  - Liegt den Prüfenden der Quellcode vor oder nicht?
  - Back-Box Test (Prüfende kennen vorher keine Systemdetails) vs. White-Box Test (Prüfende kennen die gesamte Systembeschreibung/-architektur). White-Box Tests sind meist aussagekräftiger.
  - Wird auf Sicherheit gegen Angriffe von Innen oder Außen geprüft?
- Die Qualität des Penetrationstests hängt von Expertise und Spezialisierung der Prüfenden sowie dem Prüfauftrag und der investierten Zeit ab.
- Schwachstellen werden protokolliert, gemeldet und können später behoben werden.

Weitere Aspekte

Je nachdem, was veröffentlicht wurde (App im AppStore, Systembeschreibung, ganzer Quellcode), können also verschiedene Analysen durchgeführt werden. Dabei muss man beachten:

- Diese Analysen sind abhängig vom Kontext und der Expertise der analysierenden Person. Nicht jede Person mit IT-Expertise kann sich zu allen Aspekten des Codes und der Infrastruktur kompetent äußern.
- Einzelthemen, zu denen eine Person oder Firma Expertise haben kann, sind z.B. Kryptographie, Server-Infrastruktur, Datenbanken, Web Security, verschiedene Programmiersprachen.
- Schnittstellen und Anbindungen an andere Apps oder Betriebssysteme können ähnlich analysiert werden. Da diese aber oft von anderen Anbietern stammen, ist der Zugriff darauf oft schwieriger – weshalb manche Anwendungen in ihrer Gesamtheit nur mit hohem Aufwand zu überprüfen sind.

Wie lange dauert eine Code-Analyse?

- Sehr von verschiedenen Faktoren abhängig:
  - Komplexität, Qualität und Dokumentation des Codes,
  - gibt es eine Ansprechperson bei der Entwicklerfirma selbst,
  - wie viele Leute reviewen,
  - Expertise der Reviewer,
  - Art der Analyse (Systembeschreibung kann schneller analysiert werden als Quellcode)
- Immer ein stetiger Prozess: Man kann nie sagen, dass eine Analyse zu 100 Prozent abgeschlossen und z.B. eine App komplett sicher ist.
- Je nach Expertise und Offensichtlichkeit der Mängel kann man eventuell nach Minuten oder Stunden schon sehr elementare Fehler finden.
- Generell dauert eine solche Analyse aber mindestens einige Tage. Bei komplexen Programmen und ohne Zugriff auf den Quellcode kann eine Analyse aber auch mehrere Wochen oder Monate dauern.

Worauf sollte bei den Ergebnissen und der Berichterstattung besonders geachtet werden?

Analysen von Programmen und Code sind oft ein langwieriger Prozess, der nicht immer leicht zu durchschauen ist. Daher sollten bei der Berichterstattung darüber einige Punkte beachtet werden. Die wichtigsten zählen wir im Folgenden auf.

Untersuchung des Codes

- Unterschied zwischen verschiedenen Arten der Begutachtung und Prüfung (erster Blick, genaue Prüfung, Penetrationstest des gesamten Systems)
- Unterschiede zwischen verschiedenen Dokumentationen der Funktion einer App (Systembeschreibung, Quellcode) sowie der Selbstdarstellung oder der Werbebeschreibung
- Grundlegende Probleme im Systementwurf einer App oder eines IT-Systems können oft schon an der Systembeschreibung erkannt werden, wenn diese detailliert genug ist. Eine

Quellcodeanalyse kann dann Probleme bei der konkreten technischen Umsetzung des Systementwurfs finden.

- Wie ist die Expertise der Reviewer zu beurteilen? Wichtig: einholen einer zweiten, unabhängigen Meinung

Verhalten der App-Entwickelnden

- Wie ist die Transparenz zu beurteilen? Wurden Dokumentation des zentralen Designs und Quellcode sowie eventuelle weitere Angaben – z.B. zu Lizensierung oder Datenschutz – rechtzeitig und gut überprüfbar veröffentlicht?
- Daraus lassen sich unter Umständen auch Schlüsse ziehen, ob die Entwickelnden überhaupt wollen, dass die Software überprüft wird, oder sie lieber keine Aufmerksamkeit darauf richten wollen.
- Gehen die Entwicklerinnen und Entwickler auf Feedback, Verbesserungsvorschläge und gemeldete Fehler ein?
- Wie schnell können Fehler behoben werden (abhängig von der Schwere der Fehler) und wie schnell werden sie behoben?

Anonymisierung vs. Pseudonymisierung

Diese Begriffe kommen immer wieder in Diskussionen um den Schutz sensibler Daten vor und müssen klar getrennt werden. Beide Verfahren sollen den Datenschutz der durch die Daten beschriebenen Personen gewährleisten.

- **Anonymisierung**: Löschen, aggregieren, trennen oder verfälschen aller Merkmale und Daten, die auf eine Person bezogen oder beziehbar sind
- So sind die einzelnen Datenpunkte untereinander nicht mehr verknüpfbar.
- Das Ziel ist, die Zuordnung der Daten zu einer Person auf diese Weise unmöglich zu machen.
- Grundsätze des Datenschutzes gelten nicht mehr für anonymisierte Daten [1].
- Komplex, da nicht nur personenbezogene Daten wie Name und Anschrift, sondern auch personenbeziehbare Daten wie Alter oder Beruf in Kombination oft zur Identifikation führen können
- In der Praxis können auch vermeintlich anonymisierte Datensätze unter Umständen deanonymisiert werden:
  - Entweder, wenn sie mit anderen Datensätzen kombiniert werden, die Informationen zur betroffenen Person enthalten,
  - oder wenn sie unzureichend anonymisiert wurden – etwa, indem nur der Name der Person entfernt wurde, die Kombination der restlichen Daten aber Rückschlüsse auf die Identität zulässt.
- **Pseudonymisierung**: Ersetzen der identifizierenden Daten wie Name oder Anschrift durch ein Pseudonym
- Die Person kann durch die Besitzer oder Verwalter der Daten oft wieder identifiziert werden, da diese meist einen getrennten Datensatz besitzen, in dem festgehalten wird, welches Pseudonym welcher Person entspricht.
- Hier ist eine Kombination der Datenpunkte zur Identifikation leichter als bei der Anonymisierung.

Literaturstellen, die zitiert wurden

[1] Datenschutz-Grundverordnung (DSGVO): Erwägungsgrund 26. Satz 4 und 5.

Weitere Recherchequellen

Gemeinsame Stellungnahme zur digitalen Kontaktnachverfolgung (2021). Unterzeichnet von 77 Wissenschaftlerinnen und Wissenschaftlern aus dem Bereich der IT-Sicherheit in Deutschland.

Neumann L (06.04.2021): 10 Prüfsteine für die Beurteilung von „Contact Tracing"-Apps. Chaos Computer Club.

Der Bundesbeauftragte für den Datenschutz und die Informationsfreiheit (2020): Anonymisierung - Eine Standortbestimmung zwischen der DSGVO und dem TKG.

Ansprechpartner in der Redaktion

Bastian Zimmermann
Redakteur für Technik und Digitalisierung

Telefon +49 221 8888 25-0
E-Mail redaktion@sciencemediacenter.de

Disclaimer

Dieses Fact Sheet wird herausgegeben vom Science Media Center Germany. Es bietet Hintergrundinformationen zu wissenschaftlichen Themen, die in den Schlagzeilen deutschsprachiger Medien sind, und soll Journalisten als Recherchehilfe dienen.

SMC-Fact Sheets verstehen sich nicht als letztes Wort zu einem Thema, sondern als eine Zusammenfassung des aktuell verfügbaren Wissens und als ein Hinweis auf Quellen und weiterführende Informationen.

Dieses Fact Sheet wurde von Experten aus der Wissenschaft auf Korrektheit geprüft.

Sie haben Fragen zu diesem Fact Sheet (z. B. nach Primärquellen für einzelne Informationen) oder wünschen Informationen zu anderen Angeboten des Science Media Center Germany? Dann schicken Sie uns gerne eine E-Mail an redaktion@sciencemediacenter.de oder rufen Sie uns an unter +49 221 8888 25-0.

Impressum

Die Science Media Center Germany gGmbH (SMC) liefert Journalisten schnellen Zugang zu Stellungnahmen und Bewertungen von Experten aus der Wissenschaft – vor allem dann, wenn neuartige, ambivalente oder umstrittene Erkenntnisse aus der Wissenschaft Schlagzeilen machen oder wissenschaftliches Wissen helfen kann, aktuelle Ereignisse einzuordnen. Die Gründung geht auf eine Initiative der Wissenschafts-Pressekonferenz e.V. zurück und wurde möglich durch eine Förderzusage der Klaus Tschira Stiftung gGmbH.

Nähere Informationen: **www.sciencemediacenter.de**

Diensteanbieter im Sinne MStV/TMG
Science Media Center Germany gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg
Amtsgericht Mannheim
HRB 335493

Redaktionssitz
Science Media Center Germany gGmbH
Rosenstr. 42–44
50678 Köln

Vertretungsberechtigter Geschäftsführer
Volker Stollorz

Verantwortlich für das redaktionelle Angebot (Webmaster) im Sinne des §18 Abs.2 MStV
Volker Stollorz

science
media center
germany