

Sicherheit in Open Source



KARAKUN

François Martin



François Martin

Full Stack Software Engineer

 francois.martin@karakun.com

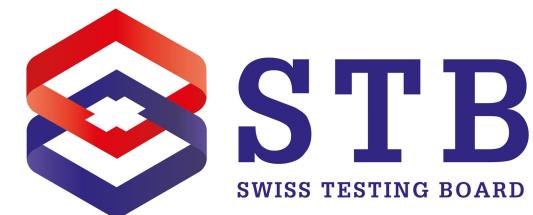
 [martinfrancois](https://github.com/martinfrancois)

 [/in/francoismartin](https://www.linkedin.com/in/francoismartin)

 [@fmartin_](https://twitter.com/fmartin_)



Fachhochschule Nordwestschweiz
Hochschule für Technik





karakun

<https://karakun.com/jobs/>

Wir entwickeln Software.

Agil. Nutzerzentriert. Erfolgreich.





Karakun im Detail

Nachhaltige Individuallösungen

Kunden aus unterschiedlichen Bereichen, u.a. Versicherung, Finanz, Life Science, Logistik

Kompetenzen

State-of-the-Art Tech-Stack (Java, Web)
Text Analytics / KI / Big Data
Fokus auf Open-Source-Software

Community Engagement

Autoren, Referenten, Java Champions,
Universitätsdozenten, Kontributoren in
Open-Source-Projekten

Dienstleistungen

Software Engineering, UX-Design,
Consulting, Training, Wartung &
Support

Plattformen & Produkte

Effizienzsteigernde Software-
Plattformen, fertige Produkte für
ausgewählte Bereiche

Erfahrenes, eingespieltes Team

60+ Mitarbeitende an 4 Standorten in
CH (Hauptsitz), D und IN





Open Source

vs.

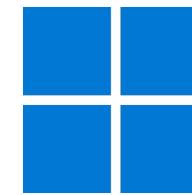
Closed Source

Beispiele

Google Android



Mozilla Firefox



Microsoft Windows



Microsoft Office

Source Code (Quelltext) der Software ist...

öffentlich und von jedem einsehbar

unter Verschluss, "proprietary", "Betriebsgeheimnis"



Wie gut funktioniert Closed Source?

Auszug von Closed Source Code veröffentlicht durch Hacker:

März 2022	<u>Microsoft</u> (Bing Suche, Bing Maps, Cortana)	9 GB
März 2022	<u>Samsung</u>	190 GB (inkl. Daten)
Februar 2022	<u>NVIDIA</u>	1 TB (inkl. Daten)
Oktober 2021	<u>Twitch</u>	125 GB (inkl. Daten)
Juli 2021	<u>EA Games</u>	751 GB (inkl. Daten)
Januar 2021	<u>Nissan</u>	18.4 GB
Mai 2020	<u>Mercedes Benz</u>	9 GB



Hacker veröffentlichen Source Code

Konsequenzen

Im schlimmsten Fall

- Schlechte Sicherheitspraxis wird publik
- Offensichtliche Sicherheitslücken
- Absichtlich eingegebauten Hintertüren
- Veraltete Libraries mit bekannten Sicherheitslücken ([Log4J2](#), [Spring](#))
- Zugangsdaten von Datenbanken / E-Mail-Server im Code
- Bei Kundensoftware Kollateralschäden

Im besten Fall

- **keine**
- Auszug aus [Microsoft's Reaktion](#) auf durch Hacker veröffentlichten Source Code (übersetzt):

“ Microsoft verlässt sich nicht auf die Geheimhaltung von Source Code als Sicherheitsmaßnahme und die Einsicht von Source Code führt nicht zu einer Risikoerhöhung.

Ist öffentlicher Code nicht weniger sicher?



Über Mythen und Legenden

- Häufige Fehlannahme, dass öffentlicher Code weniger sicher ist
 - Argument: "Hacker können Schwachstellen in öffentlichem Source Code finden und ausnutzen."
- Argument ist nicht falsch, **aber:**
 - Hacker haben bei Open Source eher etwas davon, Schwachstellen in viel genutzten Libraries (z. B. Log4j2) zu finden
 - Bei der Menge an Open Source Code ist Wahrscheinlichkeit dass Code gelesen wird gering
 - Auch wenn Code nicht öffentlich ist, sind Schwachstellen trotzdem da und werden gefunden



Ist öffentlicher Code nicht weniger sicher?

Im Gegenteil

- Bei Open Source ist es einfacher für ehrliche Sicherheitsforscher Schwachstellen zu finden und auf diese hinzuweisen (besonders bei Bug Bounty)
- Sicherheit sollte nicht in der Geheimhaltung von Source Code bestehen (Security through obscurity)
- Schweizer Regierung schreibt bei E-Voting vor, dass der Source Code Open Source sein muss:

“ Der **Quellcode** und die Dokumentation von vollständig verifizierbaren Systemen **müssen veröffentlicht werden**, so dass fachkundige Personen das System bei sich in Betrieb nehmen und analysieren können. Der Quellcode darf für **ideelle** und namentlich **wissenschaftliche Zwecke genutzt** werden. Dazu gehört der **Austausch** bei der **Fehlersuche** sowie das Recht zu **publizieren**. ”

“ Zum Einbezug unabhängiger Fachkreise im Sinne einer **öffentlichen Überprüfung** soll zum **offengelegten Quellcode** und der Dokumentation ein **Bug-Bounty-Programm** geführt werden. ”



Gute Sicherheitspraxis

Gemeinsamkeit von Open Source und Closed Source

- Secrets (Geheimnisse wie Zugangsdaten) gehören **nicht** in den Source Code
 - Tools wie [Gitleaks](#) zur Vorbeugung verwenden
- Entwicklern regelmässig genug Zeit für Wartung und Sicherheitsverbesserungen geben
 - Gute Sicherheit kostet, schlechte Sicherheit kostet mehr
- Vertrauen in Entwickler nötig
 - Gut versteckte Hintertüren sehen aus wie Bugs und sind fast unmöglich zu finden
 - Neu schreiben der Software häufig günstiger als ein Audit des Source Codes auf Hintertüren



Empfehlung

Closed Source Code so behandeln, als wäre er öffentlich

- **Aber:** warum dann nicht gerade Source Code öffentlich machen?
 - Passende Open Source Lizenz wählen
- Open Source schafft Vertrauen und Transparenz gegenüber Kunden
 - Kunden wissen auch ohne Programmiererfahrung: ist Source Code nicht sicher oder schlecht implementiert, würde man nicht veröffentlichen
- Bei Softwareprodukt: Kunden können Verbesserungen in Form von Code beitragen, von denen alle Kunden profitieren
- “Kostenlose Einarbeitung” von neuen Entwicklern
 - z. B. React von Facebook (Meta)



Open Source Verwendung

Wie sicher ist die Verwendung von Open Source Dependencies?

- Open Source Code bedeutet **nicht** keine Sicherheitslücken!
- Risiko von Supply Chain Attacks
 - Beispiel: node-ipc überschreibt bei Update bei russischen Entwicklern Dateien mit ❤️
- Am sichersten: **exakte** Versionen definieren ("Dependency Pinning")
 - z. B. statt “1.x” → “1.1.1”
 - Erhöht Wartungsaufwand, wenn Updates nicht automatisiert
- Gefahr von "Typosquatting"
 - Legitim: @azure/core-tracing **Bösartig**: core-tracing
 - Legitim: dateutil **Bösartig**: python3-dateutil



Take-home messages

1. Einhaltung von guter Sicherheitspraxis ist auch bei Closed Source **essenziell**
2. Geheimhaltung von Source Code führt **nicht** zu sicherer Software (Security through obscurity)
3. **Closed Source Code so behandeln, als wäre er öffentlich**
4. Veröffentlichung von Source Code schafft Vertrauen und Transparenz

Unsere #diwodo22 Beiträge



👉 www.karakun.com/youtube



Karakun AG

Elisabethenanlage 25
4051 Basel
Switzerland

T. +41 61 551 36 00
E. info@karakun.com
W. www.karakun.com