PRESSEMITTEILUNG

Berlin, 22.06.2022 – Eine Publikation der WG Security der OSB Alliance

Sicherheit: Open Source Software und proprietäre Software im Vergleich

Seit es Open Source Software gibt, wird sie von der Frage begleitet, wie die Sicherheit von Open Source im Vergleich zu proprietärer Software zu bewerten ist. Durch die fortschreitende Digitalisierung und die dabei zunehmende Durchdringung auch proprietärer Lösungen mit Open-Source-Komponenten gewinnt diese Frage an Gewicht.

Dabei gibt es kaum noch Closed-Source-Produkte, die ohne Open-Source-Komponenten auskommt. 99 % der Produkte enthalten Open Source. Die durchschnittliche Anzahl der Open-Source-Komponenten stieg dabei von 298 (2018) auf 445 (2020), das ist ein Anstieg von fast 50 % in zwei Jahren. [1]

Weder bei Closed Source, noch bei Open Source Software gibt es absolute Sicherheit. Die entscheidende Frage ist deswegen die des „sicher genug“. Software, die kritische Systeme steuert, muss natürlich ein sehr viel höheres Sicherheitsniveau garantieren, als Unterhaltungssoftware, wie zum Beispiel Spiele, bei denen ein Absturz oder Bug der Software keine größere Gefährdung bedeutet.

Wie kommen wir zum „sicher genug“? Grundsätzlich gibt es fünf Aspekte, um die Sicherheit von Software zu bewerten, und wir können diese Aspekte jeweils für Open Source und Closed Source betrachten:

1. Inhalt der Software
2. Verhalten der Software, inklusive ihrer Schnittstellen
3. Vertrauen in Hersteller
4. Vertrauenswürdigkeit der Lieferkette
5. Grad der Herstellerabhängigkeit

Inhalt der Software: Sicherheit durch Offenheit

Hier hat Open Source einen klaren Vorteil: Open Source kann keine verborgenen Funktionen enthalten, das ist durch den offenen Charakter ausgeschlossen. Dadurch können die sich stetig verbessernden Analyse-Werkzeuge bereits im Quellcode nach Risiken suchen. Und dies unangemeldet, jederzeit und mit den Mitteln der eigenen Wahl.

Um dagegen die Inhalte von Closed Source zu überprüfen, müssen deutlich schwierigere Wege mit relevant großen Nachteilen beschritten werden. Im schlimmsten Fall müssen Quellcodes mühevoll aus den fertigen kompilierten Produkten rekonstruiert werden, ein aufwendiges Verfahren, das oft auf technische und rechtliche Hindernisse stößt. Im Falle von Cloud Diensten ist es meist unmöglich, überhaupt oder gar dauerhaft Zugriff auf den Quellcode zu bekommen.

Die nicht öffentliche Bereitstellung, wie sie manchmal praktiziert wird, führt meist auch nicht zum Ziel. Zudem kann nicht sichergestellt werden, dass der geprüfte Quellcode auch tatsächlich die Basis der zu untersuchenden Software ist. Die Expertise, um mit den Codes etwas anzufangen, liegt ausschließlich bei den Mitarbeitenden der Anbieter, eine unabhängige Prüfung ist daher praktisch nicht möglich.

Von dem Mangel an Expertenpersonen sind natürlich auch die Open-Source-Wirtschaft und die Communities betroffen, allerdings in einem ganz anderen Maße. Hier ist Auditierung möglich und die Inspektion der Quellen ist tägliche gelebte Praxis. Unabhängige Expertinnen und Experten zu finden, wird dadurch überhaupt erst möglich. Ein klarer Vorteil für für die Sicherheit von Open Source Software.

Vertrauen in Hersteller

Wenn wir die kritischen Schwachstellen wie die Zero-Day-Exploits der letzten Jahre betrachten, ergibt sich das eindeutige Bild, dass dass Closed-Source-Anwendungen mit weitem Abstand die größten Probleme verursachen: fast 95% dieser Sicherheitslücken treten in Closed-Source-Produkten auf. [2] Laut einer aktuellen Untersuchung sind bei Unternehmensanwendungen jedoch heute nur noch 45 % Closed-Source-Anwendungen im Einsatz [3]. Diese 45 % proprietärer Anwendungen verursachen also 95 % der schlimmsten Probleme. Daher können wir davon ausgehen, dass Closed Source Anwendungen deutlich unsicherer in der Nutzung ist als Open Source Software.

Ein weiterer Vorteil von Open Source Software: Selbst wenn jemand eine Sicherheitslücke einbaut, wird diese über kurz oder lang entdeckt werden. Dies kann in Einzelfällen unerwünscht lang dauern, die Entdeckung ist aber meist möglich. Zudem kann sie zusätzlich durch Automation, clevere Algorithmen und KI-Verfahren immer weiter verbessert werden.

Vertrauenswürdigkeit der Lieferkette

Softwareherstellung ist eine Industrie mit Zulieferern auf der ganzen Welt. Dies gilt grundsätzlich für jede Software, für Closed-Source-Software ebenso wie für Open Source. Fast alle Closed-Source-Produkte enthalten zudem Open-Source-Bestandteile. Daher sind die Maßnahmen, mit denen eine vertrauenswürdige Lieferkette herzustellen ist, auch vergleichbar.

Viele Bestandteile von Softwareprodukten sind nicht bis zu jeder Person zurück zu verfolgen, die Codes beigetragen hat. Auch dies gilt für Open Source und Closed Source gleichermaßen. So werden viele Zeilen Code in Ländern produziert, in denen staatliche Zugriffe nicht wirksam ausgeschlossen werden können. Zudem ist auch in größeren Softwareprojekten nicht jede beitragende Person hinsichtlich der Sicherheit überprüfbar (Backgroundcheck). Das bedeutet, dass auch kein proprietärer Hersteller alle Beiträge Personen zuordnen und diese überprüfen kann.

Um eine Software-Lieferkette abzusichern, müssen sowohl das Verhalten, wie auch die Quelltexte der Software überprüft werden. Dies kann (angesichts der oft großen Menge an Codes muss es sogar) durch technische Werkzeuge unterstützt und geleistet werden. Wiederum gilt hier: proprietärer Code selbst kann meist gar nicht, und in den wenigen Ausnahmen nur sehr schwer und aufwendig überprüft werden.

Bezüglich der Lieferkette können Software-Anbieter und Communities die Resilienz erhöhen, indem sie auf die Zuverlässigkeit der Programmierer achten und unterbinden, dass anonyme Personen Änderungen am Code vornehmen. Hier hat Open Source Software den großen Vorteil gegenüber proprietärer Software, dass das Erkennen böswilliger Code-Veränderungen mit größerer Wahrscheinlichkeit auffällt, weil eine große Gruppe an Entwicklern Änderungen am Code jederzeit überprüft.

Eine weitere wirksame Maßnahme sind Vorgaben für die Code-Qualität, die Einbindung statischer Code-Analyse und mehrere Perimeter als Schutz interner Code-Repositories. Eine bewährte Technik sind außerdem Bug Bounties, bei denen Finderinnen und Finder von Schwachstellen mit teilweise sehr hohen Summen belohnt werden.

Die Sicherheit der Lieferkette erfährt durch die Einführung von Mindestanforderungen für Open Source und proprietäre Software gleichermaßen deutliche Verbesserungen, wenn einige zentrale Anforderungen berücksichtigt werden, die entscheidend zur Sicherheit der Lieferkette beitragen:

- **Multi-Faktor-Authentifizierung** – Die Authentifizierung der Entwickler sollte mindestens einen weiteren Faktor, wie z. B. einen FIDO2 nutzen, um Identitätsdiebstahl möglichst zu verhindern.
- **Integritäts-Verifikation mit Signaturen** – Alle Pakete sind zu signieren, nach Möglichkeit mit einem nicht direkt auf dem Repository liegenden Schlüssel.
- **Qualitätsanforderungen an den Code** – Mindestanforderungen an die Qualität des Codes sollten existieren und sowohl automatisiert, als auch mit manuellen Stichproben überprüft werden.
- **Statische Code Analyse** – Jeglicher Code, der ausgeliefert werden soll, muss durch eine gute, automatisierte Analyse auf einfach zu findende Schwachstellen untersucht sein.
- **Automatisierte Audits** – Der Client eines Repository muss es erlauben, alle installierten Pakete und über Abhängigkeiten installierte Pakete gegen Listen bekannter Schwachstellen abzugleichen und neue Versionen zu installieren.
- **Bug Bounties** – Finanzielle Belohnungen für die Finderinnen und Finder von Schwachstellen können große Verbesserung der Sicherheit bringen.
- **Geprüfte Identitäten der Entwickler** – Es sollte klar sein, wer Code verändert oder einfügt. Klarnamenpflicht schafft Vertrauen, auch wenn es politisch heiß umstritten ist, von vielen stark abgelehnt wird und auch der Gesetzgeber dazu verpflichtet ist, pseudonyme Nutzung zu ermöglichen. Eine Authentifizierung der pseudonymen Identität über z. B. ein asymmetrisches Schlüsselverfahren ist da denkbar, damit man zumindest etwas Schutz hat, dass „Hein23“ von gestern auch heute noch die gleiche Person ist, auch wenn diese einen neuen Account anlegt.

Grad der Herstellerabhängigkeit

Sicherheit bedeutet zunehmend auch hohe Verfügbarkeit. Die Digitalisierung macht uns immer abhängiger davon, dass wir Zugriff auf Kerntechnologien haben. Open-Source-Komponenten sind die Rohstoffe der Digitalisierung, über die wir souverän verfügen können. Das Gegenteil davon sind Abhängigkeiten, die sich dort zu Monopolen verdichten, wo wir nicht gegensteuern.

Wer sich in diese technischen Abhängigkeiten begibt, gerät auch in wirtschaftliche und politische Zwickmühlen – das gilt gleichermaßen für Unternehmen wie Organisationen im öffentlichen Bereich. Abhängigkeiten so weit wie möglich zu vermeiden, stärkt die Verfügbarkeit, Sicherheit und nicht zuletzt auch die eigene Verhandlungsposition.

Zudem erhöht ein lebendiges Ökosystem gegenüber Monopolisten die Resilienz gegen Angriffe. Das Bundesamt für Sicherheit in der Informationstechnologie (BSI) stellt dazu fest: „Da Software-Monokulturen mit weniger Aufwand angegriffen werden können und so schneller ein höheres Schadensausmaß erreichen, sieht die IT-Strategie des Bundes daher vor, die Vielfalt von Software zu erhöhen und so Monokulturen zu reduzieren. Eine größere Auswahl an Software führt auch zu mehr Hersteller-Unabhängigkeit.“ [4].

Fazit

Die Sicherheit von Software ist immer kritisch zu betrachten, bei Closed Source Software wie bei Open Source. Jedoch gibt es Werkzeuge und Methoden, die bei Betrieb, aber auch Herstellung von Software ein höheres, und oft auch hinreichendes Sicherheitsniveau befördern können. Diese Werkzeuge und Methoden sollten immer eingesetzt werden, egal ob Open Source oder Closed Source verbreitet wird. Letztere ist jedoch weniger gut prüfbar, zudem birgt sie Verfügbarkeitsrisiken. Open Source ist daher durch die Quelloffenheit und Verfügbarkeit immer im Vorteil. Dies gilt für Hersteller auch dann, wenn – wie so häufig - Open Source in proprietäre Produkte eingebaut wird. Nutzer von Software sind jedoch mit Open Source besser geschützt, und sollten daher beim Einsatz von Software immer auf einen möglichst großen Anteil von Open Source zielen.

Referenzen

[1] https://www.securitymagazine.com/articles/92368-synopsys-study-shows-91-of-commercialapplications-contain-outdated-or-abandoned-open-source-components

[2] https://googleprojectzero.blogspot.com/p/0day.html

[3] https://www.redhat.com/en/enterprise-open-source-report/2022

[4] https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Freie-Software/freie-software.html

Über die Open Source Business Alliance (OSB Alliance)

Die Open Source Business Alliance (OSB Alliance) ist der Verband der Open Source Industrie in Deutschland. Dabei vertreten wir über 180 Mitgliedsunternehmen, die in Deutschland ca. 10.000 Mitarbeiter beschäftigen und jährlich mehr als 1,7 Milliarden Euro erwirtschaften. Gemeinsam mit wissenschaftlichen Einrichtungen und Anwenderorganisationen setzen wir uns dafür ein, die zentrale Bedeutung von Open Source Software und offenen Standards für einen erfolgreichen digitalen Wandel im öffentlichen Bewusstsein nachhaltig zu verankern. Dieser digitale Wandel soll Unternehmen, Staat und Gesellschaft gleicherweise zugutekommen. Zudem sollen Innovationen im Bereich Open Source vorangetrieben werden. Unser Ziel ist es, Open Source als Standard in der öffentlichen Beschaffung und bei der Forschungs- und Wirtschaftsförderung zu etablieren. Denn Open Source und offene Standards sind zwingende Grundlagen für digitale Souveränität, Innovationsfähigkeit und Sicherheit im digitalen Wandel und damit die Antwort auf eine der größten Herausforderungen unserer Zeit.

OSB Alliance – Bundesverband für digitale Souveränität e.V.
Pariser Platz 6a
10117 Berlin

Tel.: +49 (30) 300 149 3377
E-Mail: presse@osb-alliance.com
Internet: www.osb-alliance.com