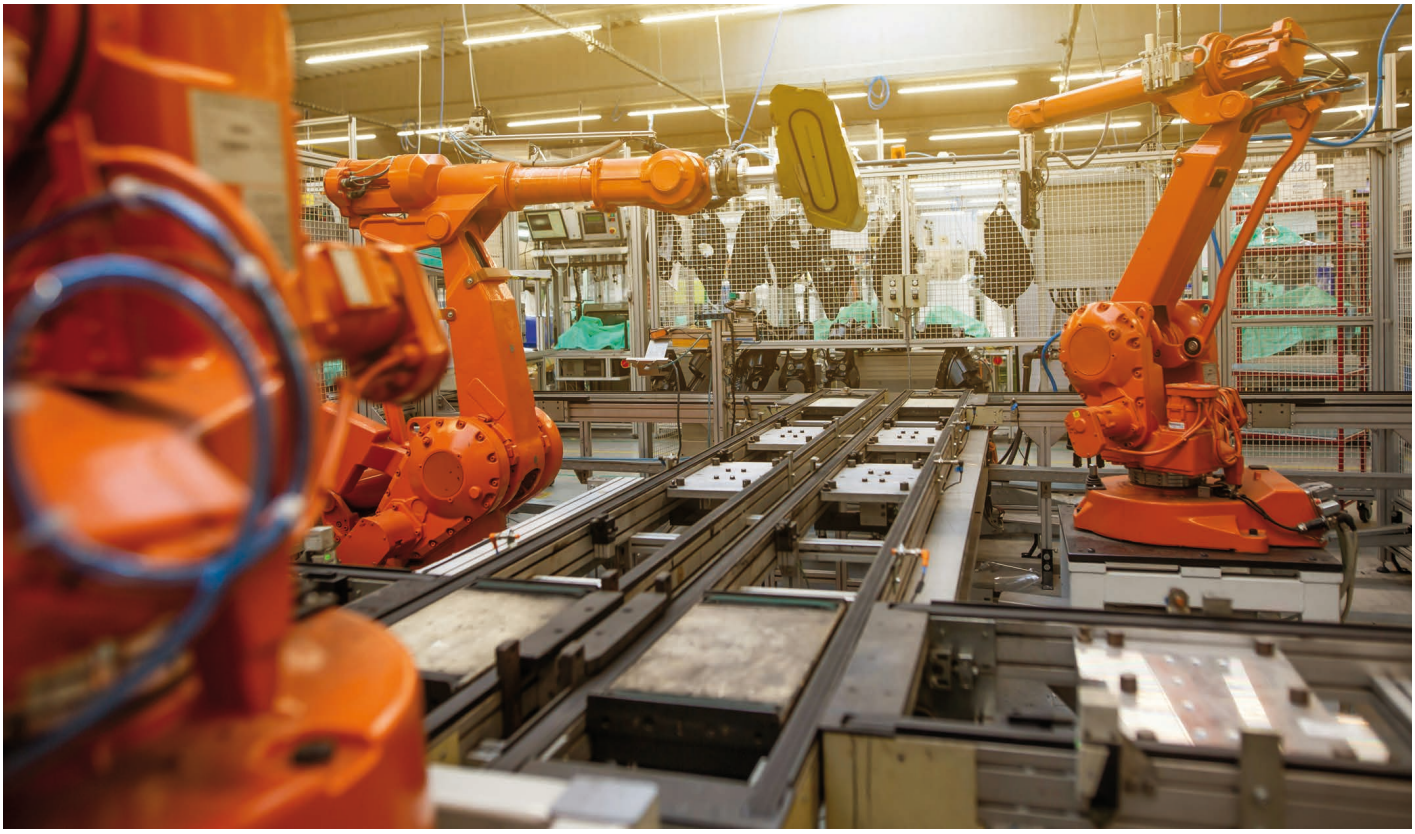
# Open Source Software sicher im Griff

***Software Composition Analysis***



***Blick in die Fertigung © Unsplash/ SimonKadula, CC0-Lizenz***

*Autorin:*
*Nicole Segerer*
*SVP & General Manager*
*Revenera*
*www.revenera.de*

Von der Kaffeemaschine über den Kernspintomographen bis zur Klimaanlage – Software findet sich heute so gut wie in jedem Produkt. Hersteller stellt dieser Digitalisierungs-Schub gleich mehrfach vor Herausforderungen – zum Beispiel was den Cyberschutz und die Compliance angeht. Analysetools und Prozesse rund um Software Composition Analyse (SCA) schaffen hier Sicherheit auf Codeebene.

Der Automobilsektor ist ein gutes Beispiel für den stetig wachsenden Softwareanteil in ehemals Hardware-Produkten. Nach Angaben von McKinsey [1] wird der weltweite Markt für Automobilsoftware und -elektronik bis 2030 auf 462 Milliarden US-Dollar ansteigen. Der zweitgrößte Anteil entfällt auf die Softwareentwicklung, einschließlich Integration, Verifizierung und Validierung (83 Milliarden US-Dollar). Schon heute sind in vernetzten Fahrzeugen 60 bis 80 Systeme integriert, die einzelne Funktionen steuern. Das entspricht zwischen 50 und 80 Millionen Zeilen Softwarecode, die die Autobauer größtenteils von Drittanbietern beziehen.

## Großer Anteil an Open Source Software

Der Automobilsektor ist kein Sonderfall. Mit zunehmender Vernetzung und Digitalisierung findet sich Software in beinah jedem Gerät. Open Source Software (OSS) macht dabei bis zu 80-90 % der Codes aus. Es ist gängige Methode, dass Entwickler bestehende (und bewährte) Codebausteine „recyclen“ und sich auf OSS-Plattformen wie GitHub oder npm bedienen. Anders wäre es kaum noch möglich, effiziente Prozesse und eine schnelle Markteinführung sicherzustellen.

## Doppelte Bedrohung: Sicherheit und Compliance

Die OSS-Strategie birgt Risiken, die weniger mit den Komponenten selbst als mit dem nachlässigen Umgang zusammenhängen. Denn leider läuft die Nutzung von OSS im Entwicklerprozess häufig gänzlich unverwaltet ab. Eine genaue Dokumentation, die nachvollziehbar darlegt, woher die Codebausteine stammen und wo sie zum Einsatz kommen, fehlt oft oder ist lückenhaft. Zumal Entwickler Komponenten über unabhängige Quellen beziehen und die für gekaufte Software bestehenden Governance-Programme umgehen. Damit gehen auch Angaben über Lizenzen und Compliance-Richtlinien sowie bekannte Schwachstellen über kurz oder lang verloren.

## Schwachstellenmanagement im Blindflug

Auf der Risikoliste ganz oben steht die Cybersicherheit. Jede Software hat Sicherheitslücken, die in regelmäßigen Abständen nach Überprüfung, Updates und Patches verlangen. Wer aber nicht weiß, was im Code seiner Software steckt,

kann solche Risiken nicht entschärfen. Jede neue Cyberattacke oder bekannt gewordene Schwachstelle löst eine aufwändige Suche aus, wobei der gesamte Softwarecode schnellstmöglich gescannt werden muss.

Softwareschwachstellen stellen immer wieder das Einfallstor für Cyberkriminelle dar, um Ransomware-Attacken, DDoS-Angriffe und Phishing-Kampagnen anzustoßen. Das BSI [2] registrierte laut Jahresbericht 2023 zur IT-Sicherheit in Deutschland durchschnittlich rund 70 neue Schwachstellen in Software-Produkten pro Tag. Das sind nicht nur rund ein Viertel mehr als noch im Vorjahr. Jede sechste Vulnerability wurde zudem als kritisch eingestuft (Bild 1).

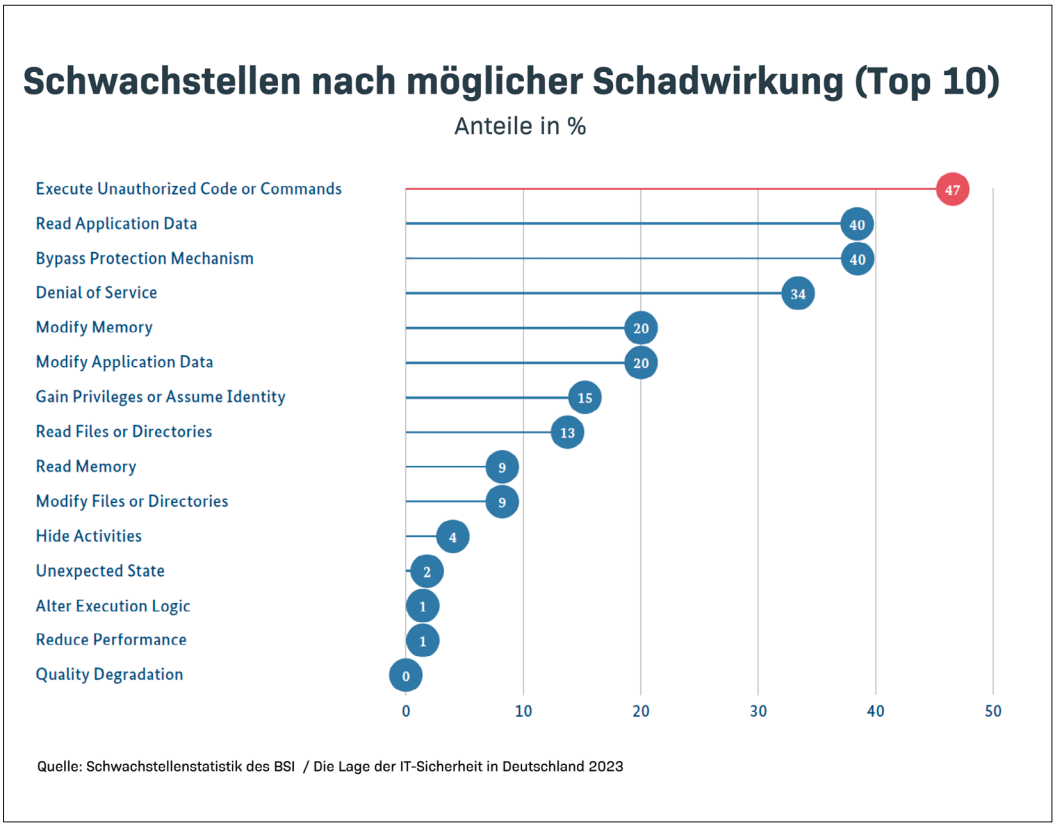



***Bild 1: Schwachstellen lassen sich für unterschiedliche Zwecke ausnutzen. © BSI***

## Cybersecurity Mindeststandards

Der Gesetzgeber hat die Auflagen in Sachen Cyberschutz dementsprechend verschärft. Zu den Wichtigsten gehört die EU-Richtlinie NIS2 [3], die Cybersecurity Mindeststandards vorlegt und in Deutschland rund 29.000 Unternehmen betrifft. Die Richtlinie muss bis Herbst 2024 in nationales Recht überführt werden. Dabei gelten die neuen und strengeren Vorschriften erstmals auch für Branchen, die bislang unter dem Radar flogen. Für Hersteller ebenfalls zentral ist der Cyber Resilience Act (CRA) [4] der EU, der von Herstellern verlangt, die Sicherheit von Hardware- und Softwareprodukte während des gesamten Lebenszyklus eines Produkts ernst nehmen.

## Compliance: Freiheit mit Grenzen

Während das Sicherheitsbewusstsein für OSS wächst, tut sich die Compliance weiterhin schwer. Hartnäckig hält sich die Ansicht, Open Source sei frei verfügbar. Dass diese Freiheit jedoch an komplexe und manchmal restriktive Lizenzen und Nutzungsrichtlinien geknüpft sein kann, fällt oft unter den Tisch. So können zum Beispiel Entwickler Quellcode, der unter die GNU-Lizenz (General Public License GPL) fällt, kostenfrei und legal nutzen, kopieren, weiterverbreiten und ändern. Jedoch sind sie verpflichtet alle auf GPL-basierten Entwicklungen ebenfalls offenzulegen. Bei kommerziellen Produkten ist der Konflikt mit der unternehmenseigenen Intellectual Property vorprogrammiert.

## Offene Technologieplattformen

Der Autoriese BMW beispielsweise musste den Quellcode einer Fahrzeugsoftware veröffentlichen, da Teile davon unter GPL lizenziert waren. Der Code einschließlich Sicherheitslücken landete schließlich auf GitHub. Initiativen, die eine Standardisierung von Open Source sowie offene Technologieplattformen im Automotive-Sektor vorantreiben, gibt es schon seit Jahren. Dazu gehört u. a. COVESA [5] (vormals GENIVI Alliance) und Automotive Grade Linux (AGL) [6] der Linux Foundation, die gemeinsam mit Autoherstellern an einer kollaborative Open-Source-Anwendungsplattform arbeiten. Branchenunabhängige Standards rund um die Open Source Lizenzierung und Sicherheit werden unter anderem vom Open-Chain-Projekt [7] vorangetrieben.

## Software Composition Analysis

Der Nachholbedarf beim Open Source-Management ist trotz dieser Bemühungen nach wie vor groß wie eine branchenübergreifenden Studie [8] zeigt. Der Softwareexperte Revenera analysierte mehr als 2,6 Milliarden Codezeilen und stieß durchschnittlich alle 11.500 Codezeilen auf einen Compliance-Verstoß, eine Sicherheitsschwachstelle oder Ähnliches. 83 % der in den Audits aufgedeckten Risiken war den Unternehmen im Vorfeld der Untersuchung nicht einmal bekannt (Bild 2).

Ein Grund für diesen Mangel an Ein- und Durchblick ist laut der Experten nicht nur die wachsende Komplexität von Software-Codes und Software Supply Chain. In Unternehmen fehlen zudem Prozesse, Richtlinien und Tools, um Anwendungen auf Codeebene zu scannen und Compliance-Verstöße sowie Sicherheitslücken frühzeitig zu erkennen.

## Best Practices

Software Composition Analyse (SCA) bietet hier als Teil eines umfangreichen Application-Security-Tests sowohl Lösungen als auch Best Practices, um die Analyse der Sicherheit, Lizenzkonformität und Codequalität zu automatisieren. Die Analyse auf Codeebene liefert darüber hinaus die Grundlage für andere zentrale Aufgaben der Softwareentwicklungspraxis.

- **Shift-Left in der Softwareentwicklung**
  In modernen DevOps- oder DevSecOps-Umgebung findet SCA möglichst früh statt, d. h. zu Beginn der Softwareentwicklung. Entwickler und Sicherheitsteams widmen sich damit proaktiv und kontinuierlich Aufgaben rund um das Testen, Scannen, Beheben und Nachverfolgen des Codes. Je früher sich das Software Vulnerability Management und die Lizenz-Compliance als feste Bestandteile des Entwicklungsprozesses bzw. Build-Prozess verankern, desto effizienter können im Folgenden Arbeitsläufe ablaufen.

- **Software Bill of Materials (SBOM) auf Knopfdruck**
  SBOMs liefern eine Übersicht aller Top-Level-Komponenten, Sub-Komponenten einer Software sowie deren direkte und transitive Abhängigkeiten. Das Erstellen solcher Stücklisten läuft mit Hilfe von SCA-Tools mittlerweile

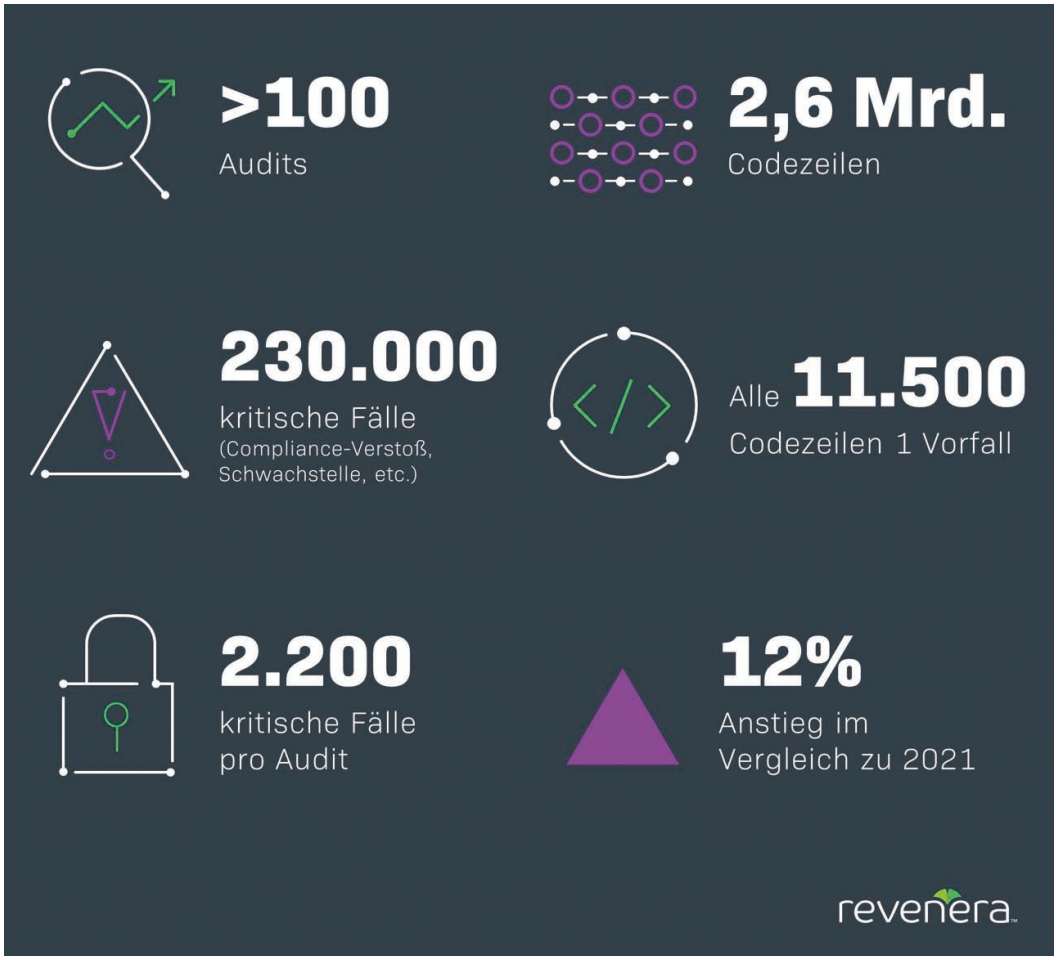



***Bild 2: Ergebnisse des Statusreport 2022 zu Open Source Software (© Revenera)***

weitestgehend automatisiert ab. Die Lösungen aggregieren Daten aus unterschiedlichen Quellen und fassen sie in einem Standardformat (z. B. SPDX, Cyclone DX) zusammen. Die SBOMs enthalten damit nicht nur interne Informationen, sondern auch SBOMs von Upstream-Partner und Drittanbieter sowie Daten aus SCA-Scans, Open Source Software-Libraries und anderen Data Services. Immer häufiger werden SBOMs zudem um Sicherheitsberichte wie Vulnerability Disclosure Report (VDR) und Vulnerability Exploitability eXchange (VEX) ergänzt, um aktuelle Informationen über Schwachstellen sicherzustellen. Je umfangreicher, vollständiger und genauer die SBOM, desto besser lässt sich auch die Gesamtqualität des Codes bewerten.

- **Schwachstellen erkennen und priorisieren**
  Im Zusammenhang mit Schwachstellenmanagement hilft SCA, Updates und Patches bedarfsgerecht und nicht nach dem Gießkannenprinzip zu verteilen. SBOMs in Kombination mit VDR und VEX dienen als Ausgangspunkt. Der Abgleich mit einschlägigen Datenbanken (z. B. National Vulnerability Database, NVD) liefert dann Sicherheitsteams das Risikolevel und die Kritikalität der Schwachstellen. Was jedoch fast noch wichtiger ist: Die Analyse zeigt, ob die eigene Anwendung überhaupt von der Sicherheitslücke betroffen ist. In der Cybersecurity, in der jede Minute zählt, kann das ein entscheidender Vorsprung sein.

- **Open Source Tracking**
  SCA läuft kontinuierlich ab und zielt darauf, Software und damit verbundene Produkte über den kompletten Lebenszyklus hinweg zu schützen. Die Analyse beinhaltet daher auch die Nachverfolgung und Dokumentation aller Open Source-Komponenten über einen längeren Zeitraum hinweg. Ziel ist es, die Transparenz entlang der Software Supply Chain zu erhöhen und langfristig sicherer zu machen. Entsprechende SCA-Tools unterstützen Entwickler darüber hinaus, an Open-Source-Komponenten vorgenommene Änderungen ordnungsgemäß zu dokumentieren und gemäß der Lizenzbestimmungen mit der OS-Community zu teilen.

- **Richtlinien festlegen und durchsetzen**
  Um Workflows und Best Practices im Umgang mit Open Source im täglichen Betrieb durchzusetzen, braucht es mehr als nur interne Richtlinien. Trainings und Fortbildungen helfen, das Bewusstsein zu schärfen und internes Know-how aufzubauen. Dedizierte Teams für das Management von Open Source Software entwickeln eine unternehmensweite Open Source-Strategie und sorgen für deren Durchsetzung. Ein Open Source Program Office (OSPO) versammelt dabei Experten aus unterschiedlichen Bereichen, darunter Recht, Software-Engineering, Sicherheit und Produktmanagement.

## Wer schreibt:

Nicole Segerer blickt auf über 15 Jahre Erfahrung in den Bereichen Softwareproduktstrategie und Marketing zurück. Bei ihr dreht sich alles um die Analyse von Softwareprodukten und darum, den Mehrwert der Lösungen sowie das Kundenerlebnis zu steigern.

Als SVP und General Manager von Revenera bei Revenera unterstützt sie Softwareanbieter und IoT-Hersteller bei der Umstellung auf neue digitale Geschäftsmodell und der Optimierung der Softwaremonetarisierung.

## Links

[1] McKinsey:
www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030

[2] BSI:
www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2023.html

[3] NIS2:
www.consilium.europa.eu/en/press/press-releases/2022/11/28/eu-decides-to-strengthen-cybersecurity-and-resilience-across-the-union-council-adopts-new-legislation

[4] Cyber Resilience Act (CRA):
https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act

[5] COVESA:
https://covesa.global/

[6] Automotive Grade Linux:
https://www.automotivelinux.org

[7] OpenChain-Projekt:
www.openchainproject.org

[8] Studie:
https://info.revenera.com/SCA-RPT-OSS-License-Compliance-2022/?lead_source=PR ◄