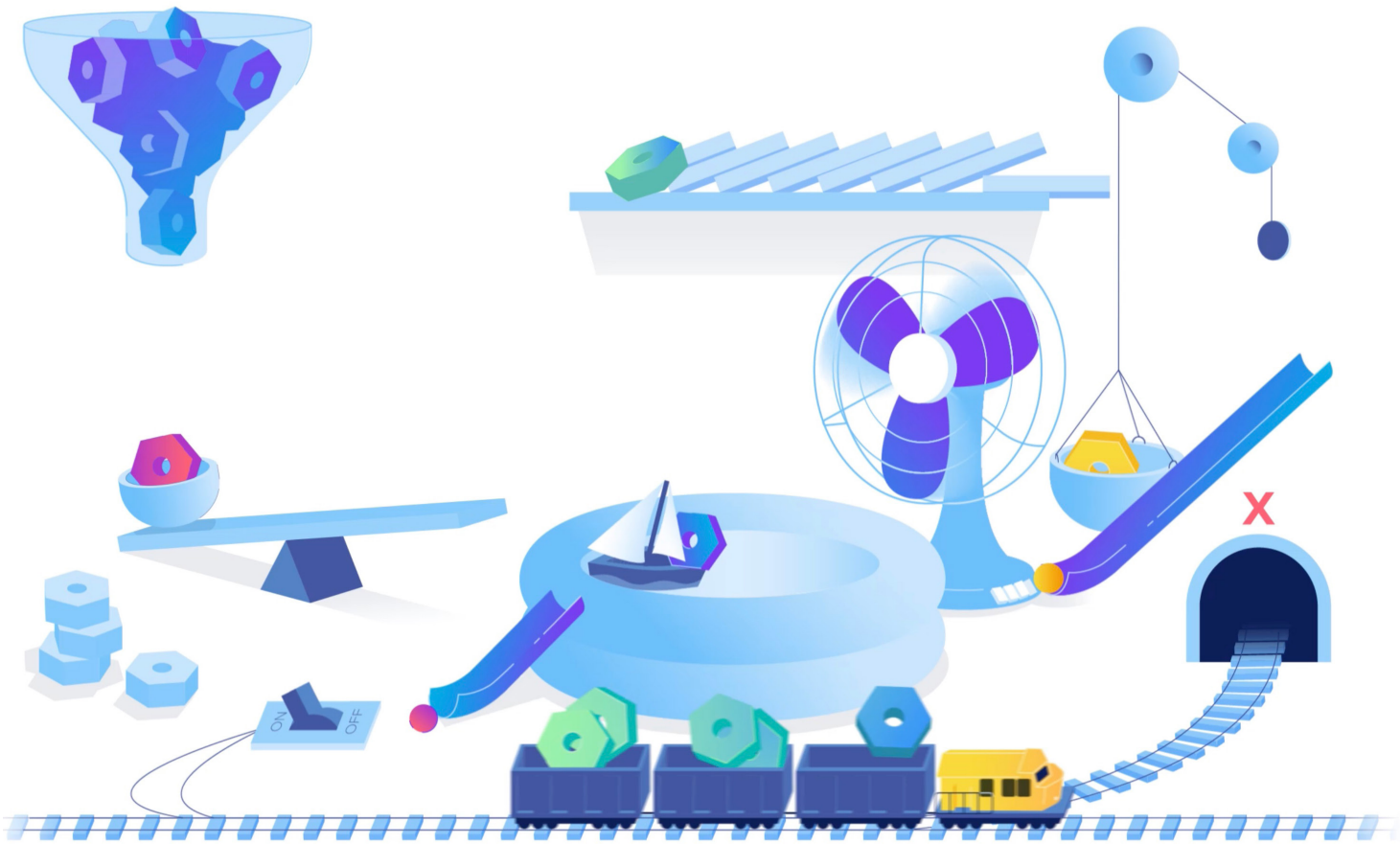
sonatype

Wie Unternehmen ihre Software-Lieferketten schützen können



sonatype

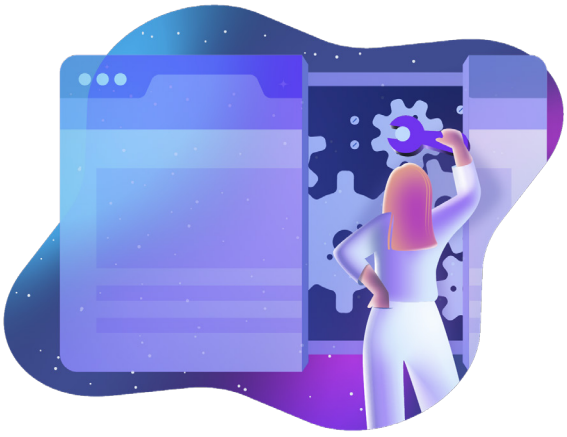
Wie Unternehmen ihre Software-Lieferketten schützen können

Software entsteht heute zu großen Teilen aus Komponenten anderer Programme. Die NIS-2-Richtlinie der EU und der geplante Cyber Resilience Act (CRA) fordern Transparenz über Hersteller, Abhängigkeiten und Versionen aller Software-Komponenten, die ein Unternehmen einsetzt. Doch wie lässt sich angesichts zahlreicher bösartiger Komponenten sicherstellen, dass sich kein Code mit verborgener Malware in die eigenen Produkte einschleicht?

Software-Steuerungen sind allgegenwärtig. Haushaltsgeräte wie Waschmaschinen oder Geschirrspüler basieren ebenso auf Software wie moderne Fernseher und Radios, Türklingeln und Heizungssteuerungen, Autos oder der öffentliche Nahverkehr. In der Industrie gilt die computergesteuerte Fertigung heute als Selbstverständlichkeit.

Im Zuge der Digitalisierung dreht sich das Rad immer schneller. Mehr und mehr Produkte erhalten eine Software zur Steuerung und/oder Überwachung. Eine Schätzung des Onlinemagazins Medium[1] aus dem Jahr 2020 ergab, dass pro Jahr mehr als 93 Milliarden Zeilen Programmcode neu entstehen, die Zuwachsraten steigen exponentiell an. Entwickler stehen unter immer stärkerem Zeitdruck. Der Fachkräftemangel und die steigende Nachfrage führen dazu, dass sie immer mehr Code in kürzeren Zeiträumen produzieren müssen.

Die Entwickler suchen deshalb nach Möglichkeiten, ihre Arbeit effektiver und effizienter zu gestalten. Eine Möglichkeit besteht darin, auf bereits vorhandenen Code aus anderen Projekten zuzugreifen. Software muss nicht jedes Mal von Grund auf neu geschrieben werden. Viele Funktionen, Routinen und Prozesse existieren bereits in anderen Programmen und lassen sich von dort nach einer Anpassung übernehmen.



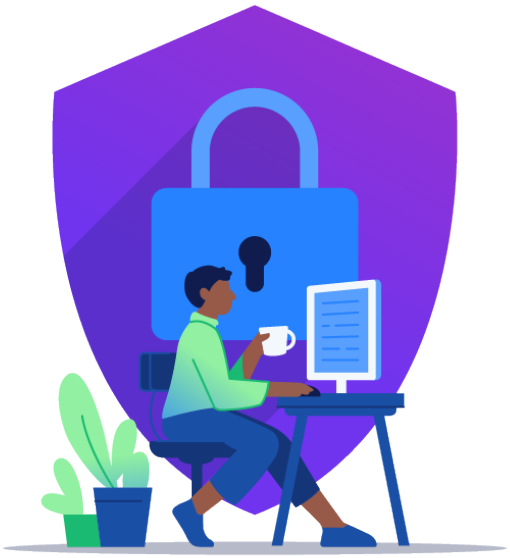
[1] **MEDIUM:** How much computer code has been written?, (https://medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459#:~:text=This%20table%20shows%20us%20that,Code%20per%20person%20each%20year)

Software unterliegt allerdings dem Urheberrecht, außerdem ist der Code fremder Anwendungen nicht freigelegt. Doch es gibt Ausnahmen: Open-Source-Software (kurz: OSS) legt nicht nur den Code offen vor, sondern erlaubt per Lizenz oftmals auch eine freie Verwendung. Seit etwa zehn Jahren geht der Trend dahin, dass Entwickler zunehmend auf quelloffene Software zurückgreifen, um sie für eigene Projekte zu verwenden. Sie nutzen dabei allerdings in den meisten Fällen nicht die komplette Open-Source-Anwendung, sondern picken sich lediglich Code-Schnipsel heraus, die sie dann in ihren eigenen Projekten weiterverwenden.

Dieses Verfahren zählt mittlerweile zur Normalität. OSS hat sich am Markt durchgesetzt und gilt für die Anwendungsentwicklung als unverzichtbar. Doch gleichzeitig entstand dadurch eine äußerst kritische Situation.

Welche Gefahren von Open-Source-Software ausgehen

OSS galt ursprünglich als wirksames Mittel gegen Bugs und Manipulationen. Dank des einsehbaren Quellcodes lässt sich OSS von jedermann einfach überprüfen. Da die Open-Source-Projekte jedoch immer umfangreicher wurden, löste sich dieser anfängliche Vorteil auf. Große Open-Source-Projekte bestehen aus mehreren Millionen Codezeilen und werden ständig weiterentwickelt. Bereits bei einem durchschnittlichen Projekt kommt es pro Jahr im Schnitt zu zehn Veröffentlichungen, und es entstehen nacheinander 15 Versionen. Niemand schafft es, diese Massen an Code zu kontrollieren. Entwickler verwenden deshalb in der Regel Open-Source-Code, ohne ihn zuvor eingehend zu prüfen – sie vertrauen auf die Communitys der Open-Source-Entwickler.



Aus Zeitgründen bauen sie den fremden Code teilweise ohne ausreichende Dokumentation in ihre Projekte ein. Folglich wissen Unternehmen häufig nicht, an welchen Stellen welcher Teil von welcher Open-Source-Software eingefügt wurde. Hinzu kommt, dass OSS nicht nur über die jeweilige Projekt-Website veröffentlicht wird. Oft gibt es Dutzende von Quellen, die über das gesamte Web verteilt sind, darunter auch nicht zentral erfasste Docker-Container. Das bringt gleich mehrere Probleme mit sich:

- Stammt die OSS nicht von der Projekt-Site, ist sie häufig nicht aktuell. Bekannte und bereits behobene Fehler können noch immer enthalten sein. Das stellt besonders dann ein Problem dar, wenn es sich um Sicherheitslücken handelt.

- Auch die kriminelle Szene hat die große Bedeutung von OSS für die Anwendungsentwicklung entdeckt. In mehreren Fällen gelang es Malware-Herstellern bereits, den Code von OSS so zu verändern, dass er bei der Ausführung automatisch die Rechner der Entwickler infizierte. Von dort aus drangen sie dann in weitere Systeme des Unternehmens ein, klauten Daten oder starteten Ransomware-Angriffe.

Was der Fall Log4Shell lehrt

Wie bedrohlich die Situation geworden ist, zeigt ein Fall, der sich Ende 2021 ereignete: Damals wurde bekannt, dass das Java-Logging-Framework Log4j eine Zero-Day-Sicherheitslücke enthält. Eine Injektionsschwachstelle erlaubte es Angreifern, nahezu beliebigen Code auf den betroffenen Systemen auszuführen.

Log4j zählt zu den Open-Source-Anwendungen und wird von der Apache Software Foundation gepflegt. Das Framework ist in zahlreichen Open-Source-, aber auch kommerziellen Programmen enthalten und zuständig für die Weiterleitung von Info- oder Fehlermeldungen an ein Logging-System. Unter anderem greifen Produkte von Amazon, Apple, Microsoft und Tesla auf Log4j zurück.

Die Sicherheitslücke wurde am 10. Dezember 2021 erstmals beschrieben und erhielt die Bezeichnung Log4Shell. Recherchen ergaben, dass sie bereits seit 2013 in der Software enthalten war, jedoch über Jahre hinweg nicht entdeckt wurde. Bereits am 9. Dezember 2021 veröffentlichte Apache einen ersten Patch, der die Gefahr beseitigte. Doch wie immer dauerte es, bis die Anwender den Patch eingespielt hatten – in vielen Fällen geschah das bis heute nicht. Denn nach wie vor kursieren veraltete, anfällige Versionen von Log4j. Schätzungen zufolge sind bis heute 29 Prozent der Downloads der Software für Angriffe anfällig, obwohl es sich bei Log4Shell um die wohl berühmteste Software-Sicherheitslücke der Geschichte handelt.

Das Beispiel zeigt, welch hohes Risiko von der Verwendung von fehlerhaftem oder manipuliertem Code ausgeht. Die Auswirkun-

gen betreffen Firmen, staatliche Stellen und Privatleute gleichermaßen. Das Problem hat sich in den vergangenen Jahren massiv verschärft, sodass die EU den Cyber Resilience Act und die NIS-2-Richtlinie erließ, um diese Bedrohungen zu bekämpfen.

Wozu der EU Cyber Resilience Act existiert

Der Cyber Resilience Act der EU (kurz: CRA) wurde am 15. September 2022 von der Europäischen Kommission vorgestellt. Die EU reagiert somit auf die vermehrten Berichte über die Sicherheitsrisiken digitaler Produkte (etwa bei Internet-of-Things-Geräten) mit neuen Anforderungen an die Gestaltung, Entwicklung und Herstellung von Geräten mit digitalen Elementen. Die Richtlinie verlangt unter anderem, dass digitale Produkte automatisch Sicherheitsupdates benötigen. Die Hersteller müssen zudem Risikobewertungen vor der Markteinführung durchführen, ein Schwachstellen-Management für den gesamten Lebenszyklus einführen und die Produkte regelmäßigen Tests unterwerfen.

Es gilt, neu erkannte Sicherheitslücken über den gesamten Produktlebenszyklus hinweg zu schließen, maximal jedoch über fünf Jahre hinweg. Sobald der Hersteller eine Sicherheitslücke beseitigt hat, muss er die Kunden und Nutzer darüber informieren und auf damit verbundene Cybersicherheitsvorfälle hinweisen. Darüber hinaus müssen die Firmen die Vorfälle sowie aktiv ausgenutzte Schwachstellen innerhalb von 24 Stunden der Europäischen Agentur für Cybersicherheit (ENISA) melden.

Die EU unterscheidet beim Cyber Resilience Act zwischen nicht kritischen und kritischen Produkten. Zu den nicht kritischen Produkten zählen beispielsweise Festplatten/SSDs oder Computerspiele, zu den kritischen Produkten der Klasse 1 gehören unter anderem Browser und Passwortmanager. Kritische Produkte der Klasse 2 sind beispielsweise Firewalls für den industriellen Einsatz, Router, Chipkarten und Chipkartenleser. Es existiert zudem eine Kategorie mit hochkritischen Produkten, die derzeit jedoch noch leer ist. Kritische Produkte müssen die Anforderungen einer Konformitätsbewertung erfüllen, die auf Basis von EU-Standards erfolgen soll. Die Konformität wird mit einem CE-Kennzeichen dokumentiert, die Umsetzung wird von den nationalen Behörden überwacht.

Nach der Verabschiedung des Gesetzes haben die Hersteller noch zwei Jahre Zeit, die neuen Anforderungen umzusetzen. Schwachstellen und Sicherheitsvorfälle müssen sie innerhalb eines Jahres melden. Hält sich ein Unternehmen nicht an die neuen

Regelungen, kann es mit einer Geldstrafe von bis zu 15 Millionen Euro oder 2,5 Prozent seines weltweiten Jahresumsatzes im vorangegangenen Geschäftsjahr bestraft werden.

Der EU Cyber Resilience Act soll nach derzeitigem Stand 2024 in Kraft treten. Berücksichtigt man die Übergangsfrist, müssen demnach ab 2026 alle Produkte die Anforderungen erfüllen.

Was die NIS-2-Richtlinie besagt

In Jahr 2023 veröffentlichte die Europäische Kommission zudem die zweite Fassung der Richtlinie zur Netz- und Informationssicherheit: NIS 2. Sie betrifft alle Unternehmen, die kritische Infrastruktur betreiben, also Einrichtungen, Anlagen und Systeme, deren Funktionsfähigkeit wichtige Rollen für die Gesellschaft, die Sicherheit des Landes und der Wirtschaft spielen.

NIS 2 enthält unter anderem Bestimmungen für die Sicherheit von Lieferketten und für Meldepflichten. Firmen müssen demnach regelmäßig Penetrationstests durchführen, Systeme zur Meldung von Cybervorfällen einrichten und eine Risikobewertung vornehmen, welche die potenziellen Gefahren für die IT-Sicherheit aufzeigt.

Die NIS-2-Richtlinie ist bereits EU-Recht, die Mitgliedstaaten müssen sie bis zum Oktober 2024 in nationales Recht überführen. Wie die Richtlinie jeweils umgesetzt wird und welche Unternehmen zur kritischen Infrastruktur zählen, legt jeder Mitgliedstaat selbst fest.

Was ein Inhaltsverzeichnis für Software-Komponenten (SBOM) leistet

Nicht nur wegen der kommenden gesetzlichen Regelungen sollten Unternehmen Buch darüber führen, welche Software-Komponenten in welchen Anwendungen enthalten sind. Da vor allem große Firmen solche Programme oft für den eigenen Bedarf produzieren, sind sie ebenfalls darauf angewiesen, keine fehlerhaften Komponenten zu verwenden.

Seit der Entdeckung der Log4j-Sicherheitslücke gehen mehr und mehr Betriebe dazu über, für ihre Software-Produkte eine SBOM (Software Bill of Materials) zu führen. Sie listet alle verwendeten Komponenten wie beispielsweise Bibliotheken oder Pakete auf und umfasst auch den aus anderen Produkten übernommenen Open-Source-Code und die Software-Komponenten von Drittherstellern. Für jede Komponente nennt die SBOM die Version, eventuelle Abhängigkeiten, Lizenzinformationen, Patchlevel und bekannte Schwachstellen. Die US-amerikanische National Telecommunications and Information

Administration definierte 2021 die Minimalanforderungen an eine SBOM und legte für jede Komponente den Namen des Anbieters, den Namen der Komponente, die Version, eindeutige IDs, eventuelle Abhängigkeitsbeziehungen, den Autor der SBOM-Daten sowie Datum und Uhrzeit der Zusammenstellung der SBOM-Daten als Pflichtfelder fest. Die SBOM muss für jede neue Version eines Software-Produkts auf den aktuellen Stand gebracht werden.

Durch das konsequente Führen von SBOMs weiß ein Software-Hersteller bei Bekanntwerden einer Sicherheitslücke sofort, ob seine Produkte betroffen sind. Die Verwaltung der SBOM-Listen erfordert jedoch einen hohen manuellen Aufwand, der in den meisten Fällen an den Entwicklern hängen bleibt. Die Folgen: Die Entwicklung verzögert sich, die Time to Market wird länger, es entstehen zusätzliche Kosten.

Außerdem müssen sich Unternehmen darüber im Klaren sein, dass es sich bei einer SBOM um ein rein reaktives Mittel handelt. Sinnvoller wäre dagegen ein proaktiver Ansatz, mit dem sich verhindern ließe, dass Schadcode überhaupt ins Spiel kommt. Darüber hinaus gibt es noch weitere Anforderungen:

- Es soll im Entstehungsprozess von Anwendungen verhindert werden, dass in ein Produkt gefährliche oder unsichere Komponenten eingebaut werden.
- Die Gefahrenbewertung von Komponenten soll an externe Stellen ausgelagert werden, da Firmen diese Aufgabe selbst nicht effizient erledigen können.
- Die Entscheidung, ob ein Code verwendbar ist oder nicht, sollte weitgehend automatisiert getroffen werden.
- Die Lösung sollte sich nahtlos in CI/CD-Pipelines integrieren lassen.

Warum die Hersteller bei Software-Lieferketten Hilfe benötigen

Mittlerweile existieren mehrere Lösungen, die sich genau um dieses Problem kümmern. Sie bieten ein Software-Supply-Chain-Management, das ähnlich funktioniert wie die Verwaltung einer Lieferkette beispielsweise in der Automobilindustrie. Denn genauso wie bei der Produktion von Autos werden beim Entwickeln von Software mehrere Teile von verschiedenen Herstellern zusammengesetzt. Zum intern erzeugten Code kommen externe Code-Anteile hinzu, bei denen es sich heute häufig um Open Source handelt. Die einzelnen Code-Teile müssen dabei von bestmöglicher Qualität sein und dürfen keine Fehler aufweisen. Eine Qualitätskontrolle, wie sie im vorherigen Abschnitt beschrieben wurde, zählt daher zum Pflichtprogramm und muss Teil der Software-Lieferkette sein.

Die Unternehmen selbst können diese Aufgabe nicht im Alleingang bewältigen. Das gilt umso mehr für Firmen, deren Kerngeschäft in einem ganz anderen Gebiet liegt, etwa bei der Automobilproduktion oder dem Bau von Haushaltsgeräten. Sie benötigen eine weitgehend automatisiert arbeitende Lösung für das Management ihrer Software-Supply-Chain, die darüber hinaus noch weitere Dienste zur Verfügung stellt:

- Sie verhindert die Nutzung gefährlicher oder unsicherer Software-Pakete und -Aktualisierungen.
- Sie gibt bei Bekanntwerden einer Sicherheitslücke Auskunft, welche Anwendungen des Unternehmens betroffen sind.
- Sie weist auf neue Versionen und Patches hin, mit denen sich problematische Komponenten ersetzen oder patchen lassen.
- Sie erbringt diese Mehrwerte, ohne dass zusätzliche Komponenten installiert oder Konfigurationen geändert werden müssen, da sie sich nahtlos in CI/CD-Pipelines integriert.

Wie die Sonatype-Plattform Schutz bietet

Sonatype hat sich ganz und gar dem Software-Supply-Chain-Management, der Erkennung und dem Schutz vor fehlerhaftem oder mit Sicherheitsrisiken behaftetem Open-Source-Code verschrieben. Die Sonatype-Plattform besteht aus drei Komponenten:

- Die Repository Firewall von Sonytype identifiziert Schwachstellen und gibt entsprechende Warnungen aus. Sie blockt automatisch bösartigen Code und gibt bereinigte Versionen wieder für die Entwicklungspipeline frei. Außerdem setzt sie Richtlinien bereits vor dem Download um.

- Bei Nexus Repository Pro handelt es sich um einen Repository-Manager, der einerseits eine Verwaltung lokal vorgehaltener Komponenten erlaubt und den Entwicklern einen geprüften, sicheren Code-Pool zur Verfügung stellt. Andererseits enthält die Software einen Cache, mit dem sich öffentliche Repositorys lokal zwischenspeichern lassen. Dadurch sind sie für die Entwickler ständig und schnell verfügbar.

- Sonatype Lifecycle warnt vor neu erkannten Sicherheitslücken und gibt den genauen Ort an, wo die Schwachstelle sitzt. Er führt automatisierte Compliance-Prüfungen durch und stellt Entwicklern die Werkzeuge zur Verfügung, um sichere Open-Source-Komponenten auswählen zu können.

Die Schwierigkeit besteht darin, die Datenbank der bekannten Sicherheitslücken in OSS ständig auf dem neuesten, möglichst tagesaktuellen Stand zu halten. Denn Verzeichnisse wie beispielsweise die National Vulnerability Database unter https://nvd.nist.gov hinken den Erkenntnissen der Sicherheitsforscher meist einige Zeit hinterher.

Sonatype hat aus diesem Grund eigene Mechanismen entwickelt, die mögliche Manipulationen an Software erkennen. Für menschliche Sicherheitsspezialisten wäre das allerdings aufgrund der täglich veröffentlichten Open-Source-Menge eine unlösbare Aufgabe – allein schon die Versionsverwaltung GitHub meldet in ihrem Octoverse Report[2], dass bei ihr im Jahr 2022 rund 413 Millionen Beiträge zu Open-Source-Projekten eingegangen sind.

Sonatype setzt deshalb bei der Sichtung von OSS auf künstliche Intelligenz. Diese achtet auf etwa 50 Merkmale, die auf böswillige Manipulationen hinweisen. Das kann beispielsweise ein bislang unbekannter Mitarbeiter mit einer russischen E-Mail-Adresse sein, der in den vergangenen drei Jahren lediglich einen einzigen Beitrag zu einem Projekt geliefert hat. Oder jemand hat den Namen einer Komponente leicht verändert. Anstatt auf ein Paket namens „Colors“ weist sie nun auf ein nicht existierendes Paket mit dem englischen Namen „Colours“ hin. Letzteres verleitet Entwickler dann dazu, das vermeintlich fehlende Paket von einer unsicheren Quelle im Internet herunterzuladen.

Wenn die KI einen solchen Hinweis auf eventuelle bösartige Manipulationen identifiziert, markiert sie die Komponente als verdächtig. Anschließend sieht sich ein Security-Forscher den Code an und entscheidet, ob tatsächlich ein Sicherheitsrisiko besteht. Sonatype gibt an, dass die Erkennungsrate der KI mittlerweile bei über 90 Prozent liegt.

[2] **GITHUB:** The state of open source software, (https://octoverse.github.com)

Glossar

CI/CD-Pipeline (Continuous Integration/Continuous Deployment):
Eine solche Pipeline umfasst mehrere definierte Schritte, die für die Fertigstellung einer neuen Software-Version ausgeführt werden müssen.

EU Cyber Resilience Act (CRA):
Ein Gesetz mit von der EU entwickelten Richtlinien für die Gestaltung, Entwicklung und Herstellung von Produkten mit digitalen Elementen.

NIS 2 (Netz- und Informationssicherheit):
Von der EU formulierte Richtlinien mit Bestimmungen für die Sicherheit von Lieferketten und für Meldepflichten. Sie betreffen ausschließlich Unternehmen und Einrichtungen der kritischen Infrastruktur.

OSS (Open-Source-Software):
Software mit offenem, frei zugänglichem Code.

SBOM (Software Bill of Materials):
Ein Verzeichnis aller verwendeten Komponenten in einer Software. Sie umfasst Bibliotheken und Pakete ebenso wie den aus anderen Produkten übernommenen Open-Source-Code und die Software-Komponenten von Drittherstellern.

Über Sonatype

Sonatype ist das Unternehmen für Software-Supply-Chain-Management. Wir geben Entwicklern und Sicherheitsexperten intelligente Tools an die Hand, um Innovationen in größerem Umfang sicherer zu machen. Unsere Plattform deckt jedes Element des gesamten Lebenszyklus der Softwareentwicklung eines Unternehmens ab, einschließlich Open-Source-Code von Drittanbietern, Quellcode von Erstanbietern, Infrastruktur als Code und containerisierter Code. Sonatype identifiziert kritische Sicherheitsschwachstellen und Codequalitätsprobleme und meldet die Ergebnisse direkt an die Entwickler, wenn diese sie am effektivsten beheben können. Dies hilft Unternehmen bei der Entwicklung qualitativ hochwertiger, sicherer Software, die ihre geschäftlichen Anforderungen und die ihrer Endkunden und Partner voll erfüllt. Mehr als 2.000 Unternehmen, darunter 70 % der Fortune-100-Unternehmen, und 15 Millionen Softwareentwickler vertrauen bereits auf unsere Tools und Anleitungen, die ihnen helfen, hervorragende und sichere Software zu entwickeln und zu pflegen.

Für weitere Informationen besuchen Sie bitte Sonatype.com oder kontaktieren Sie uns auf Facebook, Twitter oder LinkedIn.

Kontakt