Was sind Third-Party Abhängigkeiten und SBOMs?

Was sind Third-Party Komponenten?

Es gibt diesen bekannten XKCD Comic der zeigt, wie der gesamte Haufen der modernen digitalen Infrastruktur durch eine einzige Bibliothek stabilisiert wird, die von einer beliebigen Person aus Nebraska gepflegt wird. Diese einfache Satire fasst sowohl die unglaubliche Nützlichkeit von Abhängigkeiten von Drittanbietern als auch alle Gefahren, die sie mit sich bringen, perfekt zusammen. Zwar verwendet jedes Unternehmen Komponenten von Drittanbietern in seinen eigenen Produkten, doch der einfache und sichere Umgang mit ihnen ist oft noch ein Buch mit sieben Siegeln.

Einfach ausgedrückt: Alles, was kein „hausgemachter" Teil eines Softwareprodukts ist, sondern von externen Entwicklern vorgebaut wurde, gilt als Abhängigkeit von Dritten. Dazu gehören Codeteile, Bibliotheken, Frameworks und Dienste. Mit benutzerdefinierten Änderungen und Forks ist diese Grenze etwas unschärfer geworden, aber dazu später mehr. In den meisten Fällen - wenn auch nicht immer - sind diese Komponenten quelloffen, d. h. es handelt sich um Einheiten und Dienste, deren Quellcode im Internet frei zugänglich ist. Die Abhängigkeiten von Drittanbietern sind oft in Repositories organisiert, die in die Entwicklungsprozesse integriert werden, so wie Paketmanager wie Maven mit seinem Central Repository .

Was sind die Vorteile der Verwendung von Third-Party Komponenten?

Zu viele Köche können den Brei verderben, aber sie schreiben großartige Software. Wenn bewährte und gut getestete Komponenten verwendet werden, kann die Gefahr, unwissentlich angreifbare Lösungen zu bauen, minimiert werden (zumindest, wenn sie auf dem neuesten Stand gehalten werden). Einige wichtige Funktionen, wie z. B. die Kryptografie, sollten niemals im Alleingang entwickelt werden, sondern sind auf die Bemühungen einer ganzen Gemeinschaft von ausgewiesenen Experten angewiesen. Die Auslagerung der Konzeption und Entwicklung bestimmter Funktionen wie Protokollierung, Tests oder Authentifizierungsverfahren spart immense Ressourcen. Dadurch hat das Entwicklungsteam mehr Zeit, sich auf die Entwicklung völlig neuer Funktionen zu konzentrieren und den Kunden einen direkten Mehrwert zu bieten.

Wie können Komponenten von Third-Party Anbietern Unternehmen gefährden?

Natürlich hat die Verwendung externer Komponenten nicht nur Vorteile, sondern kann auch neue Probleme mit sich bringen. Eines davon ist ganz einfach: Kein Stück Software ist hundertprozentig sicher. Die weit verbreitete Verwendung einiger Bibliotheken zieht böswillige Akteure an, und ausnutzbare Schwachstellen in wichtigen Komponenten können ganze Branchen in Gefahr bringen (z. B. log4j). In der Regel ist die erste Gegenmaßnahme ein Update auf eine nicht angreifbare Version der betroffenen Software; manchmal ist eine Abschwächung der Angriffsfläche möglich, indem die Integration zwischen der Fremdkomponente und der eigenen Lösung angepasst wird. Im schlimmsten Fall muss die unsichere Abhängigkeit entweder durch ein hauseigenes Element oder eine andere Open-Source-Alternative ausgetauscht werden.
Ein weiteres erhebliches Risiko für Unternehmen, die Komponenten von Drittanbietern einsetzen, kann sich aus einer mangelnden Sorgfalt bei der Lizenzierung ergeben. Die vollständige Einhaltung der Bedingungen einiger Lizenzen kann für Unternehmen schwerwiegende Folgen haben, die ihr

Geschäftsmodell an den Rand des Zusammenbruchs bringen können. Das Schlagwort in diesem Zusammenhang ist Copyleft, das von den Herstellern von Software, die bestimmte Bibliotheken von Drittanbietern enthält, verlangt, deren offene Lizenzbedingungen einzuhalten. Dies ist vor allem bei der freien Verfügbarkeit des Quellcodes ein Problem.

Wie kann ich mein Geschäftsmodell absichern?

Man kann nicht bekämpfen, was man nicht sieht. Dies gilt insbesondere für große Unternehmen, die Tausende von Software-Assets verwalten. Daher ist die Grundlage, für alle Maßnahmen zur Sicherung der Verwendung von Komponenten von Drittanbietern in einem Unternehmen, die Erstellung eines umfassenden Überblicks über diese Komponenten; eine Art von Dokument, das besser als „Software Bill of Material“ (SBOM) bekannt ist. Ein tieferes Eintauchen in die Welt der SBOMs und wie sie zu einem der wichtigsten Werkzeuge in der modernen Softwareentwicklung wurden, wird in zukünftigen Artikeln dieser Serie behandelt.

Welche Probleme können SBOMs lösen?

Für Unternehmen, welche die in ihren Produkten verwendeten Komponenten von Drittanbietern nicht nachverfolgen, gibt es einen besonderen Ort, der so genannt wird: „Dependency Hell“. Ursprünglich beschrieb er einen Zustand der Frustration, der durch Inkompatibilitäten zwischen Softwareversionen entsteht. Heute kann seine Bedeutung jedoch leicht erweitert werden, um Sicherheits- und Compliance-Probleme einzubeziehen, die durch ein schlechtes Abhängigkeitsmanagement entstehen.

Halte deine Freunde nah und deine Feinde näher

Komponenten von Drittanbietern können beides gleichzeitig sein: Freunde, da sie helfen können, Ressourcen zu sparen und oft eine einfache Möglichkeit bieten, Best Practices der Branche zu implementieren; und Feinde, da sie (manchmal bekannte) Schwachstellen einführen können. Das obige Sprichwort beschreibt sogar die Lösung für dieses Problem: „Sie in der Nähe halten“ sollte als „Wissen, wo sie sind“ interpretiert werden - und nicht nur wo sie sind, sondern auch wer und warum. Ein umfassender Überblick über die Abhängigkeiten von Drittanbietern ist für jedes Unternehmen von entscheidender Bedeutung, um Risiken zu erkennen und zu entschärfen, bevor sie zu ausnutzbaren Schwachstellen werden.

Ein Dokument, um sie alle zu beherrschen

Dokumentation ist der Schlüssel zu allen Bemühungen, die Verwendung von Komponenten von Drittanbietern sicherer und konformer zu machen. Das Mittel der Wahl hierfür ist die Software Bill of Material, kurz SBOM. Sie identifiziert alle Abhängigkeiten eines Softwareprodukts eindeutig, in der Regel durch die Angabe ihrer Anbieter, Namen und Versionsnummern. Zusätzliche Felder zur Beschreibung einer Komponente können einen Hash des Quellcodes oder die jeweiligen Lizenzbedingungen beinhalten. Außerdem sollten die Beziehungen zwischen den Elementen enthalten sein, um das Potenzial des Dokuments voll auszuschöpfen. Dadurch werden auch die so genannten transitiven Abhängigkeiten sichtbar, d. h. Komponenten, die nicht direkt von einem Produkt selbst implementiert werden, aber von anderen Komponenten Dritter benötigt werden. Das Endergebnis sind oft ganze Abhängigkeitsbäume, die Hunderte oder Tausende von Einheiten enthalten, einige davon sogar mehrfach. Es ist ein nicht zu unterschätzender Vorteil, eine einzelne Abhängigkeit (zusammen mit möglichen über- oder untergeordneten Komponenten) zu lokalisieren, ihren Anwendungsfall zu analysieren und die Auswirkungen einer Aktualisierung oder Substitution abzuschätzen.

Es mag sinnvoll sein, ein Dokument für jede Anwendung oder jedes Modul zu erstellen, um den Fokus zu schärfen, aber SBOMs können leicht definiert werden, um ganze Produktfamilien oder Organisationen zu beschreiben, um einen umfassenden Überblick über die Abhängigkeiten zu schaffen, Schwachstellen zu erkennen und die Rationalisierungsbemühungen zu unterstützen.

Wie erstellt man eine SBOM?

Hier muss das Rad nicht neu erfunden werden - für fast jedes Software-Ökosystem gibt es großartige Werkzeuge, deren ordnungsgemäße Umsetzung jedoch manchmal eine ziemliche Herausforderung darstellt. Hinzu kommt, dass die Anzahl der Einschränkungen und Anforderungen, die von verschiedenen Gesetzgebern für die Erstellung von SBOMs auferlegt werden, mit jedem Tag wächst. Bald wird die Verteilung einer entsprechenden SBOM zusammen mit jedem Produkt zu einer Notwendigkeit, die eine vollständige Sorgfaltspflicht erfordert - aber mehr dazu in unserem Artikel über das Cyber Resilience Act.

Entscheidend ist, die Analyse durch Dritte so früh und so automatisiert wie möglich in den Secure Software Development Lifecycle (SSDLC) zu implementieren - eine Vernachlässigung der Verantwortung in diesem Bereich hat enorme Konsequenzen. SBOMs als eines der wichtigsten Werkzeuge in der modernen Softwareentwicklung zu bezeichnen, ist daher sicher nicht übertrieben.

WoBeeCon GmbH - All rights reserved 2024