Sauber definierte Schnittstellen in komponentenbasierten Softwarearchitekturen dienen der Sicherheit und Robustheit. Sie sind Teil des Security-by-Design-Ansatzes, den genua für seine IT-Security-Produkte seit jeher verfolgt.

Foto: genua GmbH

Cyber Resilience Act

DARAUF MÜSSEN HERSTELLER UND ZULIEFERER ACHTEN

Die EU erhöht den Druck auf Hersteller digitaler Produkte: Der Cyber Resilience Act (CRA) verlangt unter anderem dokumentierte Software-Stücklisten, längere Update-Zeiträume und erweitert den Haftungsrahmen. Das hat nicht zuletzt beim allgegenwärtigen Einsatz von Open-Source Konsequenzen.

Der Cyber Resilience Act (CRA) der Europäischen Union war bereits für Anfang 2024 geplant und wird voraussichtlich noch diesen Sommer in Kraft treten. Als Verordnung wird er unmittelbar mit dem Beschluss wirksam werden.

Mit seinen Forderungen nach mehr Software-Sicherheit, Mindest-Support-Lebenszyklen, mehr Transparenz für digitale Produkte und erweiterter Haftung für die Hersteller stößt er kaum auf Widerstand, bei dem die Stoßrichtung des CRA grundlegend abgelehnt würde. Trotzdem gibt es bis zu seiner endgültigen Fassung noch Diskussionsbedarf – etwa darüber, wer Hersteller ist, welche Pflichten für wen gelten und wer am Ende haftet. Vor allem die Open-Source-Community hatte in den vergangenen Monaten noch Nachbesserungen gefordert.

Die Lieferkette unter Kontrolle bringen

Eine der zentralen Forderungen ist, dass Hersteller von Produkten, die direkt oder indirekt mit einem anderen Gerät oder Netzwerk verbunden sind, prüfen und dokumentieren müssen, welche Fremdkomponenten in ihren Erzeugnissen zum Einsatz kommen. Dazu dienen sogenannte „Software Bill of Materials" (SBOM). Wie die Dokumentation praktisch erfolgen soll, hat das BSI für deutsche Hersteller in der Richtlinie TR-03183-2 beschrieben.

Der Nutzen liegt auf der Hand: Mit nur einem Blick können Hersteller, Anwender und auch zentrale Stellen erkennen, welche Produkte von neu entdeckten Schwachstellen betroffen sind; eine wichtige Voraussetzung, um eigene Lieferketten besser kontrollieren zu können. Verbunden mit den Pflichten, Sicherheits-Updates zur Verfügung zu stellen, ist auch klar, welcher Hersteller in der Pflicht ist, entsprechend nachzubessern.

Was in der Theorie gut und sinnvoll klingt, kann in der Praxis zu Problemen führen. Softwareabhängigkeiten etwa haben eine große Tragweite. Fremdkomponenten verwenden ihrerseits wieder andere Komponenten und so fort. Das Netz der Abhängigkeiten gewinnt schnell an Tiefe und Breite und die Wahrscheinlichkeit ist hoch, dass irgendwo außerhalb des Sichtbereichs Komponenten in das eigene Produkt Einzug gehalten haben, die weder dokumentiert noch geprüft wurden.

Um die Sicherheit seiner Software gewährleisten zu können, darf man sich nicht nur auf Zusagen seiner Zulieferer und Quellen verlassen. Ein Hersteller

muss in der Lage sein, selbst die Lieferkette zu durchleuchten und in der Tiefe bestimmen zu können, welche Komponenten tatsächlich aus welcher Quelle kommen, welche Sub-Sub-Sub-Komponenten gegebenenfalls nicht mehr supportet werden und welche Sicherheitsrisiken damit verbunden sein könnten.

Konsequentes Security by Design ist gefragt

Darüber hinaus muss ein Hersteller auch in der Lage sein, auf Probleme zu reagieren. Das eigene Produkt sollte so designed sein, dass Programmkomponenten isoliert sind, dass geschirmte Bereiche existieren und Komponenten nur über definierte Schnittstellen kommunizieren können. Dadurch lassen sich Übergänge und Kontrollmöglichkeiten klar definieren und einzelne Komponenten zum Beispiel über Sandboxen zusätzlich isolieren. Im Kontext von CRA und Security by Design dient ein solcher Aufbau der Sicherheit und Robustheit.

Auf der proaktiven Seite schafft der CRA verbesserte Möglichkeiten, um Abhängigkeiten und damit die Risiken von Fremdkomponenten zu bewerten und im Vorfeld Risiken zu minimieren. Aber sowohl das Design der Software als auch die Prozesse, die Auswahl und Einbinden von Fremdkomponenten leiten, lassen sich nicht mit vertretbarem Aufwand nachträglich implementieren, um CRA-Compliance zu erreichen. Es müssen Prinzipien sein, die sich in der DNA des Herstellers und in der DNA des Produkts wiederfinden lassen.

Hinzu kommt, dass die Selbstverpflichtung für das benötigte CE-Kennzeichen bei kritischen Infrastrukturen und sicherheitsrelevanten Komponenten nicht mehr ausreichen wird. Bei besonders kritischen Systemen ist dann zwingend eine dritte Partei für die Zertifizierung erforderlich.

Spätestens dann müssen Hersteller nachweisen können, wie sie die Sicherheit ihrer Produkte tatsächlich umsetzen. Security by Design, Maintainability, Availability und Monitoring schon im Produktdesign und im Entwicklungsprozess zu verankern ist letztlich nicht nur ein Sales-Argument für sicherheitsbewusste Käufer, sondern essenziell, um den Marktzugang nicht zu verlieren.

Der Umgang mit Open Source

Die Option eines Unternehmens, auf Open-Source in seinen Produkten zu verzichten, ist kaum realisierbar. Selbst wenn dies im eigenen Unternehmen gelingt, wird es schwer sein, Auftragnehmer zu finden, die ihrerseits sich dazu verpflichten wollen. Laut verschiedenen Studien beinhalten zwischen 80 und 90 Prozent aller digitalen Lösungen Open-Source-Komponenten.

Mit Blick auf Lieferketten und CRA wirft Open-Source zwei Probleme auf: Erstens sind nicht-gewinnorientierte Akteure von der Haftung ausgenommen und zweitens gibt es einige Open-Source-Komponenten, die weit verbreitet sind.

Sicherheitslücken in Komponenten wie Log4j oder jüngst OpenSSH und XZ für Linux verdeutlichen die Gefahr, die so auch für proprietäre Software besteht: Mehr als ein Jahr nach Schließung der Log4j-Schwachstelle waren fast 40 Prozent aller Systeme immer noch betroffen. Ein Grund dafür ist sicherlich die bislang mangelnde Haftung für solche Sicherheitslücken. Systeme werden teils wider besseres Wissen aus Kostengründen nicht gepatcht. Der CRA könnte sich hier positiv auswirken. Ein zweiter Grund ist, dass das Vorhandensein dieser kompromittierten Libraries dem Inverkehrbringer – in der Regel also dem Softwarelieferanten – gar nicht bekannt ist. Die Wahrscheinlichkeit, dass irgendwo in der Lieferkette Komponenten wie Log4j eingebunden sind, ist hoch, die Wahrscheinlichkeit, dass dies bislang dokumentiert wurde, jedoch gering.

Nun gibt es aber gegenüber Open-Source-Akteuren weder Haftungsansprüche noch Nachbesserungspflichten. Es wäre auch nur schwer verständlich, wenn ein Programmierer in seiner Freizeit frei verfügbaren, nicht kommerziellen Code veröffentlicht und im Anschluss dafür haften muss, wenn ein Unternehmen damit Profit erzielen möchte. Die Python Software Foundation hatte damit gedroht, sich komplett aus dem europäischen Markt zurückzuziehen, wenn der CRA bei den Ausnahmeregelungen für Open Source nicht nachgebessert würde.

Dies bedeutet im Umkehrschluss wiederum mehr Verantwortung für die Hersteller. Denn sie müssen ihre Lieferkette in der Tiefe durchleuchten können, um eben auch die vermeintlich unkritischen, millionenfach verwendeten und mit einem Click eingebundenen Libraries im Auge zu behalten und sie müssen im Zweifel auch die eigene Kompetenz besitzen, Schwachstellen in Open-Source-Komponenten zu beheben.

Vanitas Berrymore

FÜR HERSTELLER BEDEUTET DER CRA HÖHERE ANFORDERUNGEN AN DIE SICHERHEIT IHRER PRODUKTE. FÜR KUNDEN WENIGER RISIKEN BEI DER DIGITALISIERUNG.

Steffen Ullrich,
IT-Sicherheitsforscher, genua GmbH,
www.genua.de