Wenn die Lieferkette zum Röntgen muss

Qualitätssicherung in der Software Supply Chain

Dr. Ralf Huuck, Synopsys

In den meisten Entwicklungsprojekten wird Software nicht komplett neu geschrieben, sondern baut auf bestehenden Komponenten auf. Diese Komponenten können aus vorherigen Projekten, aus Open Source Quellen oder von Zulieferern kommen. Gerade die externen Quellen bergen Herausforderungen: Wie kann man sicherstellen, dass diese Drittanbieterkomponenten den eigenen Qualitäts-, Lizenz- und Sicherheitsansprüchen gerecht werden? Dieser Artikel beleuchtet einige der Risiken beim Einbetten von Drittanbieterkomponenten und stellt die Ergebnisses einer weltweiten Untersuchung zu Sicherheitslücken in Open Source Komponenten vor. Daraus abgeleitet wird erläutert, wie automatische Software Composition Lösungen verwendet werden können, um diese Sicherheitslücken im Entwicklungsprozess zu erkennen und automatisch zu verhindern.

Einleitung

Softwaresicherheit spielt eine zunehmend größere Herausforderung bei der Entwicklung von eingebetteten Systemen. Dieses ist umso mehr der Fall, je größer Softwareentwicklungsprojekte sind und je schneller sie sich fortentwickeln. Aktuelle Beispiele finden sich in Fahrerassistenzsysteme und Industrie 4.0 Applikationen, wo die Software Millionen Zeilen von Code umfassen kann und gleichzeitig sicherheitskritische Aufgaben erfüllen muss. Diese Herausforderungen werden darüber hinaus durch weitreichende Lieferketten erschwert. Das heißt, oftmals besteht das Endsystem aus weiteren zugelieferten Komponenten, die sowohl Open Source Software als auch Code von Drittanbietern enthalten können.

Ein Beispiel für die Auswirkungen von Sicherheitslücken durch Drittanbieter ist das MIRAI Botnet [1]. Dieses Botnet umfasst in seinen verschiedenen Inkarnationen über 130.000 eingebettete Systeme, zumeist internetfähige Überwachungskameras, die durch eine gemeinsame Sicherheitslücke in einem zugelieferten Bauteil von Hackern übernommen wurden. Obwohl die Überwachungskameras von verschiedenen Herstellern vertrieben werden, basierten sie auf Lieferketten mit demselben Board und derselben Treibersoftware mit derselben Sicherheitslücke.

Im Folgenden präsentieren wird einige Ergebnisses zur Untersuchung von Risiken von eingebetteten Softwarekomponenten. Wir gehen den Fragen nach: wie häufig haben Softwarekomponenten Sicherheitslücken und wie kritisch sind diese Lücken? Darüber hinaus stellen wir einige Lösungen vor, um diese Sicherheitslücken automatisch im Entwicklungsprozess zu erkennen und zu vermeiden.

Ergebnisses zu Sicherheitslücken in Open Source Komponenten.

Synopsys publiziert regelmäßig Untersuchungen zu Sicherheitslücken in Open Source Komponenten. Dabei werden zusammenfassende Ergebnisses von Software, die auf der Protecode Plattform hochgeladen wurde, veröffentlicht [2]. Protecode ist eine Analysesoftware, die Binärdateien auf bestehen Open Source Komponenten untersucht, die Versionsnummern der Open Source Komponenten ermittelt, und

diese Komponenten mit bestehenden Datenbanken von Sicherheitslücken abgleicht. Die primäre Datenbank an Sicherheitslücken ist die National Vulnerability Database (NVD) der US NIST Behörde [3]. Diese Datenbank umfasst etwas 90.000 Einträge, die bekannte Sicherheitslücken (CVEs) dokumentieren und ihnen eine sicherheitskritische Metrik gemäß des Common Vulnerability Scoring System (CVSS) zuweist [4].

Resultate für eingebettete Open Source Komponenten von 2016/2017
Die Studie von 2016/2017 umfasst in etwa 130,000 Uploads zur Protecode Plattform. Dabei konnten über 16.000 verschiedene Komponenten und Versionen identifiziert werden. Abbildung 1 zeigt eine Untergliederung der häufigsten identifizierten Komponenten nach Aufgaben- und Nutzungsbereich. Dabei sind etwa zwei Drittel aller Komponenten Utilities für Windows- und Linux-Tools, Netzwerkprotokolle wie SLL und http, und Medienbibliotheken für jpg, png oder XML.

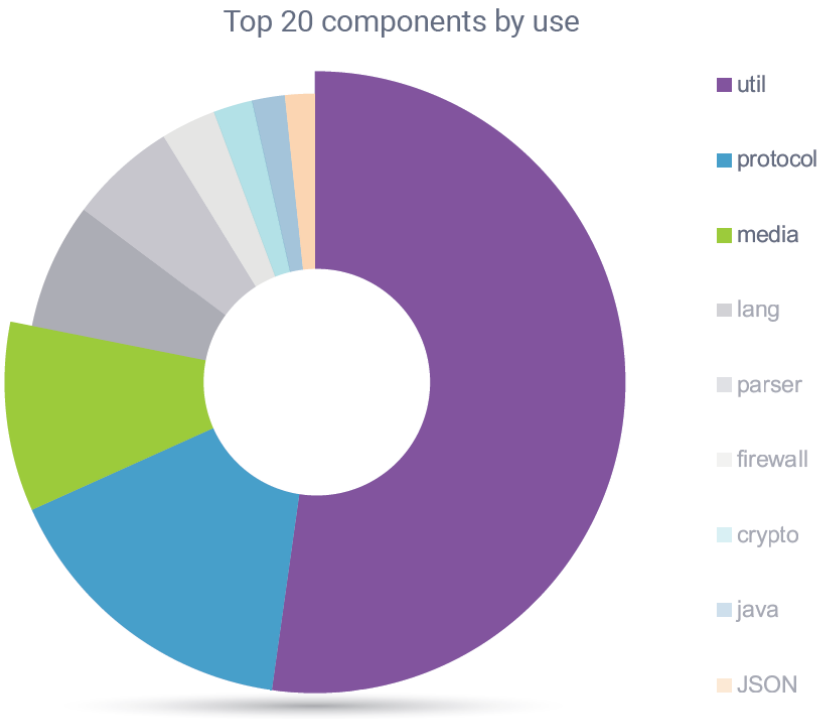

Abb. 1 Gliederung von Komponenten nach Nutzungsbereich

Diese Komponenten werden typischerweise als Open Source verwendet, da sie Standardfunktionen vereinfachen und in der Regel nicht neu implementiert werden müssen. Fälschlicherweise wird davon ausgegangen, dass diese Komponenten sicher sind oder keinen sicherheitskritischen Einfluss haben.

Es zeigt sich aber, dass aus den über 16.000 verschiedene Komponenten etwa 9.000 Sicherheitslücken (CVEs) identifizieren lassen. Das bedeutet, dass eine Großzahl der Komponenten nicht sicher ist. Darüber hinaus sind die gefundenen Sicherheitslücken nicht neu, wie Abbildung 2 zeigt:

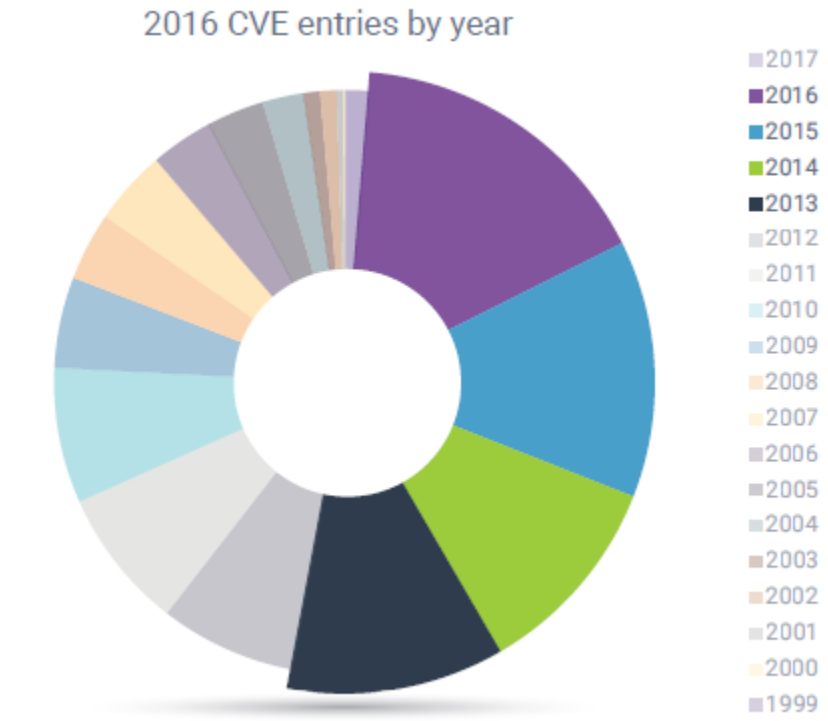

Abb. 2 Zeitraum seit Bekanntgabe der endeckten Sicherheitslücken

Dieses bedeutet, das etwas 50% aller Sicherheitslücken vier Jahre alt oder älter sind. In den meisten Fällen gibt es eine neuere und sicherere Version, die zur Verfügung steht, aber nicht benutzt wurde. Diese zeitliche Diskrepanz hängt auch damit zusammen, dass es oft einige Zeit dauert, bis Sicherheitslücken entdeckt werden. Das heißt, dass Software die heute als sicher gilt, morgen neuere Erkenntnisse haben kann. Dieses ist für Hersteller schwer zu kontrollieren und zu korrigieren.

Automische Einbindung von Composition Analysis in den SDLC
Es ist unrealistisch, auf Komponenten von Open Source Quellen und Drittanbietern zu verzichten. Dieses ist ein nützlicher Bestandteil, um Produkte kostengünstig und relativ schnell herzustellen. Darüber hinaus ist es keineswegs erwiesen, dass Open Source Komponenten schlechter als eigenentwickelte Software ist. In der Tat, das Umgekehrte ist häufig der Fall.

Dennoch ist es nicht empfehlenswert, Komponenten von Drittanbietern ohne vorherige Prüfungen einzubauen. Glücklicherweise gibt es auf dem Markt heutzutage Softwarelösungen wie Protecode, Sonatype oder Black Duck, die diese Überprüfungen automatisch vornehmen können [5]. Nicht nur das, sondern diese Softwarelösungen können auch direkt in den Entwicklungsprozess automatisiert eingebunden werden. Das heißt, es lässt sich zum Beispiel ein DevOps Jenkins Prozess starten, der bei jeder Produkterstellung eine automatisierte Analyse fährt, die Open Source Komponenten entdecken und mit ihren bekannten Sicherheitslücken abgleicht. Diese Ergebnisse können dann zeitnah den Softwareentwicklungsteams und den Qualitätsteams zur Verfügung gestellt werden.

Zusammenfassung und Ausblick
Die hier vorgestellten Erkenntnisse zeigen, dass sich routinemäßig Sicherheitslücken in Komponenten von Drittanbietern finden lassen. Dennoch sind Drittkomponenten und Open Source Pakete für die moderne Softwareentwicklung unerlässlich. Wichtig ist, dass die Lieferkette nicht mit blindem Vertrauen hingekommen wird, sondern auf ihre Sicherheitsaspekte kontinuierlich überprüft wird. Softwarelösungen im Bereich der Composition Analysis ermöglichen dieses heutzutage effizient und kostengünstig.

Mittelfristig lassen sich Lieferketten in ihrer Qualität durch Composition Analysis lenken und verbessern. Dieses kann zum Beispiel dadurch geschehen, einen engen Rückmeldungsprozess an die Zulieferer herzustellen, eine Abnahme bei unzureichender Sicherheit zu verweigern oder eine automatisierte Vorprüfung auf der Zuliefererseite zu fordern. Dieses kann zum Vorteil für alle Beteiligten gestaltet werden.

References
[1] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou and Jeffrey Voas. DDoS in the IoT: Mirai and Other Botnets. IEEE Computer, Volume 50/7, 2017.
[2] Synopsys Software Integrity Group. The State of Software Composition 2017. https://www.synopsys.com/software-integrity/resources/analyst-reports/state-of-software-composition-2017.html
[3] Harold Booth, Doug Rike, Gregory A. Witte. The National Vulnerability Database (NVD): Overview. ITL Bulletin, December 2013.
[4] Peter Mell, Karen Scarfone, and Sasha Romanosky. 2006. Common Vulnerability Scoring System. IEEE Security and Privacy 4, 6 (November 2006), 85-89.
[5] Millar S. Vulnerability Detection in Open Source Software: The Cure and the Cause. Queen's University Belfast, 2017.

Autor
Dr. Ralf Huuck ist ein Technischer Direktor bei Synopsys, Australien. Dr. Huuck arbeitet im Werkzeugbereich der Synopsys Software Integrity Group und unterstützt die Werkzeugentwicklung, um standardkonforme Produkte sicher und schnell auf den Markt zu bringen. Zuvor war Dr. Huuck Geschäftsführer von Red Lizard Software, Sydney, und langjähriger Forschungs- und Entwicklungsleiter am Forschungsinstitut, NICTA. Zeitglich ist Dr. Huuck Adjunct Associate Professor an der UNSW, Australien, und ist im Bereich Softwaresicherheit tätig.

Kontakt
Email: ralf.huuck@synopsys.com
Internet: http://www.synopsys.com.software