Konferenzbericht AppSecUSA 2016

(Joshua Tiago, Senior Berater, cirosec GmbH)

Sichere Entwicklung in Zeiten agiler Softwareentwicklung und die immer größer werdende Flut neuer IoT-Geräte – beides sehr aktuelle Themen im Bereich IT-Sicherheit. Genau diese Aspekte standen im Mittelpunkt der diesjährigen AppSec-Konferenz in Washington, D.C. So referierten Sicherheitsexperten und Researcher aus aller Welt zu diesen Themen.

Bereits die Keynote am ersten Konferenztag befasste sich mit dem Internet of Things. Joe Jarzombek, ehemaliger Leiter der National Cyber Security Division für das U.S. Department of Homeland Security, präsentierte seine Erkenntnisse aus der Sicht eines Herstellers von IoT-Produkten. Er verdeutlichte, welchen Herausforderungen sich Hersteller solcher Geräte stellen müssen. Schwachstellen in der Software jener Produkte sind so verbreitet wie in herkömmlicher Software. Allerdings lassen sich viele IoT-Produkte nur teilweise oder gar überhaupt nicht patchen, da sie in Umgebungen betrieben werden, in denen dies nicht möglich ist. Selbst wenn der Hersteller seine Software nach strengen Vorgaben unter Berücksichtigung der IT-Sicherheit programmiert, bestehen häufig Abhängigkeiten zu Drittprodukten. Jarzombek gab zu bedenken, dass zwischen 80 und 90 Prozent aller IoT-Produkte auf irgendeine Art und Weise Open-Source-Software verwendet. Schwachstellen in diesen Open-Source-Komponenten werden teilweise nur langsam oder in den schlimmsten Fällen überhaupt nicht behoben. Das führt dazu, dass die Hersteller jener IoT-Produkte eigene Patches entwickeln müssen. Künftige Produkte müssen diese Patches ebenfalls enthalten. In der Vergangenheit wurde dies oftmals vergessen – und so enthalten aktuelle Geräte plötzlich Sicherheitslücken, die bereits seit vielen Jahren bekannt sind. Hersteller müssen verstehen, dass die Sicherheit ihrer Produkte alle Bereiche der „Supply Chain“ betrifft, so Jarzombek.

Einem anderen Thema widmete sich Zane Lackey in seinem Vortrag „Practical tips for web application security in the age of agile and DevOps“. Lange Zeit galt SDLC (Software Development Life Cycle) als Modell für

Softwareentwicklung. Sicherheitstests können in die verschiedenen Phasen des SDLC integriert werden. Doch die Entwicklung hat sich in den letzten Jahren radikal verändert. So werden Entwicklungszyklen immer kürzer. Statt monatelang an vielen neuen Funktionen zu arbeiten, werden heute oftmals einzelne Funktionen in kurzen Zyklen entwickelt und freigegeben. Der Prozess der Softwareentwicklung ist gegenwärtig sehr dynamisch. Internetgrößen wie Google oder Amazon checken bis zu tausendmal am Tag neuen Code in ihre Repositorys ein. Sicherheitstests können in einem derart dynamischen Umfeld in Zeiten agiler Softwareentwicklung nicht wie gewohnt implementiert werden. Die Herausforderung besteht darin, das Sicherheitsniveau der in kurzen Zyklen entwickelten Software aufrechtzuerhalten, ohne die Produktivität und damit die treibende Kraft hinter agiler Softwareentwicklung zu beeinträchtigen – in der Praxis eine teilweise sehr schwierige Gratwanderung. Unabdingbar, so Lackey, sei eine Kombination aus statischer wie dynamischer Code-Analyse und vielen automatischen Sicherheitschecks. Bei letzteren geht es nicht um Prüfungen der gesamten App, sondern vielmehr um kleine Checks einzelner Funktionen. Diese Checks können für künftige Applikationen oder neue Funktionen wiederverwendet werden. Sprach man vor einiger Zeit noch von DevOps, so ist heute bereits von SecDevOps die Rede: eine Verschmelzung von Entwicklung und Sicherheit im Kontext der Softwareentwicklung. Es sei zu erwarten, dass in den nächsten Jahren viele Verbesserungen auf diesem Gebiet stattfinden, so Lackey.

Ein Highlight der Konferenz war der von Professor Matthew D. Green von der Johns Hopkins University präsentierte Vortag „Cryptography in the age of Heartbleed“. Green, der im vergangenen März eine Schwachstelle in Apples Messenger entdeckte und sie publik machte, gilt in der Szene als Referenz im Bereich Kryptografie. Bereits in der Einleitung zu seinem Vortrag begrüßte Green, dass die Verwendung von Kryptografie in den vergangenen drei Jahren enorm zugenommen hat. Als ausgewiesener NSA-Kritiker wies er auf die Enthüllungen zu den bekanntgewordenen Überwachungsmaßnahmen zurück. Viele Menschen möchten einen Teil ihrer Privatsphäre zurück und nutzen bewusst Applikationen und Systeme mit starker Verschlüsselung. Mehr und mehr Entwickler stehen unter Druck und müssen auf Kryptografie für ihre Anwendungen zurückgreifen. Obwohl für nahezu jede Programmiersprache Krypto-Bibliotheken zur Verfügung stehen, ist die Implementierung von Kryptografie alles andere als trivial. Entwickler wissen häufig nicht, wie sie

die Klassen und Methoden solcher Bibliotheken sicher verwenden sollen, da das Hintergrundwissen zu den gewählten kryptografischen Verfahren fehlt. Dies führt in der Regel dazu, dass sich Entwickler auf Standardvorgaben der Bibliotheken verlassen. Ein fataler Fehler, so Green, da sehr viele Bibliotheken in der Standardeinstellung unsichere Algorithmen, Schlüssellängen oder Betriebsmodi verwenden. In der Folge weisen solche Anwendungen gravierende Lücken auf. Dem Benutzer der Applikation wird ein falsches Gefühl der Sicherheit vermittelt. Green macht in erster Linie nicht die Entwickler dafür verantwortlich, sondern vielmehr Kryptografen, die diese Bibliotheken programmieren. Während ein Kryptografie-Experte durchaus versteht, welche Parameter es für eine sichere Verschlüsselung zu beachten gilt, sind die meisten Benutzer solcher Bibliotheken damit überfordert. Um dies anhand eines Beispiels zu verdeutlichen, griff Green auf eine Funktion aus der Windows Cryptography API zurück. Konkret benannte er die Funktion „CertVerifyCertificateChainPolicy“. Wie der Name der Funktion und die Beschreibung in der Dokumentation suggerieren, dient diese Funktion dazu, die Gültigkeit einer Zertifikatskette für ein bestimmtes Zertifikat zu prüfen. Schaut man sich in der Dokumentation den Rückgabewert an, so wird schnell klar, dass dies nicht der Fall ist. Die Funktion gibt lediglich einen Wert zurück, der besagt, ob die Prüfung durchgeführt werden konnte – nicht jedoch das Ergebnis. Daher plädiert Green für Bibliotheken, die standardmäßig sicher sind. Entwickler von Krypto-Bibliotheken sollten seiner Meinung nach darauf achten, die Komplexität der Kryptografie vor dem eigentlichen Anwender zu verbergen und stattdessen auf sichere Algorithmen, Schlüssellängen und Modi zu setzen. Als positives Beispiel erwähnte Green die von Daniel J. Bernstein, Tanja Lange und Peter Schwabe entwickelte Bibliothek NaCl. Trotz aller Schwierigkeiten zeigte sich Green zuversichtlich und wagte eine Prognose: Kryptografie werde eine zunehmend wichtige Rolle spielen.

Joshua Tiago, Senior Berater, cirosec GmbH