Security Implications through Legacy Software Systems

Martin Grothe

Gegenwärtig verlassen sich Unternehmen in hohem Maße auf bestehende Softwaresysteme, Protokolle und Formate, die vor mehr als zehn Jahren entwickelt oder standardisiert worden sind. Fehler in diesen Systemen können sich deshalb schädigend auf den Geschäftserfolg des jeweiligen Unternehmens auswirken. Gleichzeitig gibt es zahlreiche Beispiele, wie Unternehmen selbst an so einfachen Aufgaben wie das Installieren von Patches für ältere Betriebssysteme gescheitert sind. Wenn diese simplen Aufgaben Firmen schon vor Herausforderungen stellen, wie sicher sind dann aktuell eingesetzte Softwaresysteme, die teilweise oder gänzlich vor 10 Jahren initial entwickelt worden sind, aber von Entwicklern und Unternehmen noch vertrieben werden? Entsprechen die darin enthaltenen Software-Komponenten, Protokolle und Formate noch dem aktuellen Stand der Technik in Bezug auf die IT-Sicherheit? Diese Fragen werden in der vorliegenden Arbeit anhand von drei Beispielen aus den Bereichen Wirtschaft, Finanzen und Kommunikation exemplarisch beantwortet.

Das erste Beispiel aus dem Bereich Enterprise Rights (ERM) Management ist Microsofts Implementierung Rights Management Services (RMS). Das ERM System von Microsoft ist ursprünglich im Jahr 2003 veröffentlicht worden und wird seither ständig weiter entwickelt. RMS wird von vielen großen Firmen auf der Welt zum Schutz sensibler Daten eingesetzt. Dennoch konnten zwei kritische Angriffe auf RMS, sowie auf die verwandten Produkte Azure RMS, Office 365 und Azure Information Protection identifiziert und praktisch umgesetzt werden. Diese Angriffe erlauben es einem authentifizierten Benutzer mit Nur-Lese-Rechten den RMS Schutz von Dokumenten komplett zu entfernen, beziehungsweise die Dokumente nach Belieben zu manipulieren. Um diese Manipulation von Dokumenten zu verhindern, wird eine Gegenmaßnahme beschrieben, welche auf der vorhandenen RMS Public-Key Infrastruktur aufsetzt und somit schnell umgesetzt werden kann.

Das zweite Beispiel in der Dissertation behandelt Gridcoin, eine Kryptowährung aus dem Finanzbereich. Diese Kryptowährung hat den Zweck, die energieverschwendenden Berechnungen der Kryptowährung Bitcoin durch sinnvollere Berechnungen zu ersetzen. Diese sinnvollen Berechnungen werden durch das BOINC System durchgeführt. BOINC gibt Forschern auf der ganzen Welt die Möglichkeit, ihre Berechnungen zur Suche nach neuen Materialien, Heilmitteln für Krebs oder Covid-19, auf den Computern von Freiwilligen auszuführen. Die Idee hinter Gridcoin ist es, auf Basis von BOINC eine Kryptowährung zu erschaffen, welche die Freiwilligen für die BOINC-Berechnung auf ihren Computern monetär entlohnt. Jedoch unterliefen den Entwicklern dabei Fehler, welche zu zwei Sicherheitsproblemen führten. Durch den ersten Fehler sind teilweise E-Mail Adressen der Gridcoin Nutzer öffentlich in der Blockchain gespeichert worden. Mit dem zweiten Fehler kann ein Angreifer unberechtigterweise den Großteil der noch zu erstellenden Währung für sich beanspruchen. An diesem Beispiel zeigt die Dissertation, dass selbst eine Open-Source Kryptowährung mit einer Marktkapitalisierung von mehr als 80 Million € im Jahr 2018, kritische Sicherheitslücken aufweist. Die in dieser Arbeit vorgeschlagene und beschriebene Gegenmaßnahme basiert auf dem Prinzip des Trusted-On-First-Use und behebt die zuletzt genannte Schwachstelle.

Im dritten und letzten Beispiel wird gezeigt, dass das gut etablierte Schlüsselaustauschprotokoll für VPNs, mit der Bezeichnung IKEv1 ebenfalls kritische Schwachstellen enthalten kann. Bekannte Firmen aus dem Kommunikationsbereich wie Huawei, ZyXEL, Clavister und Cisco enthielten in ihren Implementierungen sogenannte Bleichenbacher-Padding-Orakel, welche angegriffen werden konnten. Darüber hinaus wird in der vorliegenden Arbeit eine Schwachstelle im Standard RFC 2409 für die Authentifizierung mittels Pre-Shared Keys beschrieben und am Beispiel von Openswan gezeigt, wie diese Schwachstelle praktisch ausgenutzt werden kann.

An diesen drei Beispielen wird exemplarisch aufgezeigt, dass, trotz der signifikanten Bedeutung und breiten Anwendung in der IT, diese Softwaresysteme, Protokolle und Formate kritische Schwachstellen enthalten können. Die meisten dieser Schwachstellen sind in weniger als drei Wochen nach der Analyse der Systeme oder Implementierungen identifiziert worden. Aus der Arbeit resultiert daher die Empfehlung, dass Softwareprodukte während ihrer Lebensdauer einer initialen und bei Veränderung einer fortwährenden Sicherheitsüberprüfung unterzogen werden müssten. Gefundene Schwachstellen sollten innerhalb von 30 Tagen behoben und die betroffenen Nutzer im Anschluss benachrichtigt werden.

Im Rahmen einer guten wissenschaftlichen Praxis sind die in dieser Arbeit identifizierten Schwachstellen den entsprechenden Entwicklern oder Unternehmen verantwortungsbewusst offengelegt und die Gegenmaßnahmen zur Verfügung gestellt worden.