Kurzfassung der Dissertation „Securing Application Software in Modern Adversarial Settings“ von Felix Schuster

Die Art und Weise in der Software entwickelt und genutzt wird hat sich in der näheren Vergangenheit gewandelt. So werden Softwareanwendungen heutzutage nur noch selten komplett von einer einzelnen vertrauenswürdigen Partei entwickelt. Selbst große Softwarehersteller bauen mittlerweile in ihren Produkten häufig auf Softwarekomponenten aus unterschiedlichen externen Quellen. Veränderungen ähnlicher Art gibt es auch bei der Nutzung von Software: Cloud-Computing ist einer der großen IT-Trends der letzten Jahre. Das zentrale Konzept von Cloud-Computing ist, dass Software nicht mehr lokal auf den Systemen eines Anwenders läuft, sondern *on-demand* und kosteneffektiv in den Rechenzentren eines Cloud-Computing-Anbieters. Durch diese Veränderungen ergeben sich neue Angriffsflächen für Software: Klassische Angreifermodelle in der Softwaresicherheit betrachten den Programmcode und die Ausführungsumgebung einer Anwendung typischerweise als vertrauenswürdig. Dem Angreifer wird nur die Möglichkeit zugestanden die Ein- und Ausgaben einer Anwendung in Teilen zu kontrollieren. Ein klassisches Szenario ist hier z.B. ein Web-Browser in dem vom Angreifer mit Hilfe einer bösartigen Webseite ein Pufferüberlauf erzeugt wird. Dieses eindimensionale Angreifermodell erscheint im Kontext von Softwarekomponenten aus unterschiedlichen externen Quellen und von Cloud-Computing nicht immer weitgreifend genug – es lässt wichtige Fragestellungen außer Acht: *Was ist wenn eine Softwarekomponente eine Hintertür enthält? Was ist wenn ein Cloud-Administrator mit Hardwarezugriff den Programmcode oder die Daten einer Cloud-Anwendung zur Laufzeit manipuliert oder liest?*

In dieser Dissertation wird daher das Thema Softwaresicherheit in drei Angreifermodellen bearbeitet: *(A)* Das klassische Modell, *(B)* das Modell „Externe Softwarekomponenten“ in dem der Angreifer zusätzlich einmalig Hintertüren in bestimmte Komponenten einer Anwendung einzubauen kann und *(C)* das Modell „Cloud-Computing“ in dem der Angreifer weite Teile von Soft- und Hardware kontrolliert und an vielen Stellen Programmcode und Daten einer Cloud-Anwendung lesen und manipulieren kann. Es werden zunächst existierenden Defensivansätzen (z.B. *Control-Flow Integrity*) im Modell A analysiert und bewertet. Dazu werden mehrere neuartige Code-Reuse-Angriffe präsentiert, die mit weit verbreiteten Annahmen brechen und so viele existierende akademische und kommerzielle Defensivmaßnahmen umgehen. Unter anderem zeigen die Ergebnisse hier, dass informelle Argumente oder rein empirische Belege kein hinreichendes Kriterium für die Sicherheit von Defensivmaßnahmen sind. Im Anschluss werden Herausforderungen der erweiterten Modelle B und C diskutiert und entsprechende neuartige Defensivmaßnahmen vorgeschlagen und evaluiert. Konkret wird für Modell B ein System zum Auffinden von bestimmten Arten von Hintertüren in Binärsoftware mittels dynamischer Analysen beschrieben. Für Modell C wird ein Ende-zu-Ende sicheres System zur Ausführung von verteilten Anwendungen in der Cloud beschrieben. Die Sicherheit dieses Systems baut auf zwei kryptographische Protokollen und Intels SGX-Technologie als *Trusted Computing Base* (TCB).