Abstract

Two big trends that can be observed in the way that software is developed and used are the commoditization of software components and computing infrastructure:

It has long become the norm that software products are not entirely developed by a single trusted entity. Instead, even large software vendors routinely rely on software components from different and numerous external sources. These sources include typically public open source projects or external contractors.

Changes of similar nature can be observed in the ways that software is used and deployed; without doubt has cloud computing been one of the biggest trends in IT in recent years. The central concept of cloud computing is that the users’ software no longer runs on their own systems but rather on demand, in a cost-effective manner, in the data centers of a cloud provider. This arrangement is also referred to as “infrastructure as a service”.

Due to these developments new attack surfaces for application software arise: Classically, the security of application software is considered in adversarial settings where a trusted application is running in a largely trusted environment. The attacker is only foreseen to partly control inputs and outputs of the application. A typical concrete scenario is here for example a web browser in which the attacker triggers a buffer overflow vulnerability using a malicious web site.

This one-dimensional attacker model appears not always appropriate in the context of software components from external sources and cloud computing as it disregards important risks: a software component could contain a “backdoor” or a malicious cloud administrator could access the code and data of a cloud application at runtime.

Hence, this dissertation explores the topic of application software security in three modern adversarial settings: (i) the classic setting, (ii) the backdoor setting in which the attacker may additionally add a backdoor to a component of a software, and (iii) the cloud setting in which the attacker largely controls hardware and software and may at different places read or manipulate a cloud applications code and data.

In this dissertation, we begin with an evaluation of existing defensive approaches (e.g., control-flow integrity) in the classic setting. Thereto, we present various advanced code-reuse attacks. Our attacks break with commonly held assumptions on the nature of code-reuse attacks and as such bypass many existing academic and commercial defenses. Among others, our results here show that purely intuitive arguments or limited empirical evidence are no sufficient criteria for the security of a defense.

We discuss the challenges of the backdoor and the cloud adversarial settings and propose and evaluate novel ways to tackle them. We present a dynamic analysis approach for the detection and dismantling of backdoors in binary server applications across different processor architectures. Among others, we demonstrate how our approach can disarm real-world backdoors (e.g., in a malicious version of OpenSSH) in a fully automated manner.

Furthermore, we describe the first end-to-end secure system for the execution of distributed (MapReduce) applications in the untrusted cloud. The security of this system founds on two novel cryptographic protocols—for which we provide proof—and Intel’s SGX technology as hardware-rooted trusted computing base (TCB).