Chapter 14
Building a Trusted Environment for Security Applications

Giovanni Cabiddu
Politecnico di Torino, Italy

Antonio Lioy
Politecnico di Torino, Italy

Gianluca Ramunno
Politecnico di Torino, Italy

ABSTRACT

Security controls (such as encryption endpoints, payment gateways, and firewalls) rely on correct program execution and secure storage of critical data (such as cryptographic keys and configuration files). Even when hardware security elements are used (e.g. cryptographic accelerators) software is still—in the form of drivers and libraries—critical for secure operations. This chapter introduces the features and foundations of Trusted Computing, an architecture that exploits the low-cost TPM chip to measure the integrity of a computing platform. This allows the detection of static unauthorized manipulation of binaries (be them OS components or applications) and configuration files, hence quickly detecting software attacks. For this purpose, Trusted Computing provides enhanced security controls, such as sealed keys (that can be accessed only by good applications when the system is in a safe state) and remote attestation (securely demonstrating the software state of a platform to a remote network verifier). Besides the theoretical foundation, the chapter also guides the reader towards creation of applications that enhance their security by using the features provided by the underlying PC-class trusted platform.

INTRODUCTION

Cryptography is routinely used to protect data and communication from tampering and disclosure. However cryptographic operations and keys must themselves be protected against direct attacks: if an attacker gets hold of a key or can replace a library, then he can easily bypass the cryptographic protection. In most cases, purely software protection techniques are used, but these kind of protection often fail against skilled attackers.

DOI: 10.4018/978-1-4666-4030-6.ch014

As an example, let us consider a Web server which uses the Transport Layer Security (TLS) protocol to protect its transactions. This requires access to a private asymmetric key for the server authentication and key-exchange phases. A simple solution is to store the private key in clear in the file system, relying on the access control mechanisms of the operating system to prevent unauthorized access, but this can be bypassed if an attacker can run a process impersonating the identity of the server process or has direct access to the underlining storage (e.g. being a backup operator or having physical access to the server). Even when hardware elements are used (such as a HSM, Hardware Security Module) there is still room for effective attacks based on manipulation of software building blocks (e.g. drivers, libraries) or direct memory access by privileged processes (e.g. for reading in-memory keys or sensitive data). Last but not least, if the attacker can modify the TLS configuration file then it can disable or reduce protection for certain pages by shortening the length of the negotiated symmetric encryption key.

It should be clear that a trusted environment for program execution and data storage is needed and this is actually the mission of the Trusted Computing Group (TCG) (http://www.trustedcomputinggroup.com). It introduced a set of technologies to create a "Trusted Platform," based on a hardware trust anchor capable of protecting sensitive information and identifying the components running in a computer system.

A Trusted Platform is built around a cost-effective and tamper-resistant chip called Trusted Platform Module (TPM). Most of the commercial desktop and notebook computers sold nowadays include this component, although it is usually disabled by default and rarely used by system administrators and applications developers, mostly due to ignorance of its features and difficulties in management and programming.

Key features of the TPM are its cryptographic primitives (hashing and asymmetric encryption), key and random number generation capabilities, and shielded locations to store keys and sensitive data. These features can be used for measuring the platform's integrity (and reliably reporting such data) and protecting application data and execution.

Although TPM is widely available, building a Trusted Platform is not an easy task. For a PC-class platform we aim to fill this gap with this chapter, that first briefly describes the key features of the TPM, and then explain how its cryptographic capabilities can be used to build a Trusted Platform (e.g. able to detect tampering of cryptographic libraries). Finally, we show how the TPM's capabilities are exposed to applications and how they can be used to protect the cryptographic operations and keys of a generic application.

The rest of the chapter is organized as follows: Section 2 introduces the foundation of Trusted Computing in terms of building blocks for a Trusted Platform; Section 3 defines how to build a trusted environment by leveraging the building blocks; Section 4 describes how to write an application for a trusted environment, in particular using a software library called Trusted Platform Agent; Section 5 presents related works and finally Section 6 concludes the chapter.

TRUSTED COMPUTING

Computer security is normally associated to the concepts of data confidentiality and integrity and system availability. We rarely think of security as related to trust, but for special cases (e.g. when we talk of a trusted third-party in specific protocols, such as Kerberos). However all solutions to provide security rely on some software not being altered, being executed in the proper way, and behaving as expected: these three elements can be collectively associated to the word "trust."

In social science, trust is a personal and fuzzy concept, but in the computer world, we are interested in a clear definition and exact quantitative