Chapter 14

Secure Bootstrapping Using the Trusted Platform Module

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

Ahmed Mahmoud Abbas
The American University in Cairo, Egypt

ABSTRACT

The protection of Computer Hardware and Software using Cryptographic algorithms has assumed importance in the recent years. The Trusted Computing Group (TCG) has put forward certain conditions to be met by the computer hardware, software and firmware so that the devices may be considered trusted. The Trusted Platform Module is a hardware device that will authenticate the code modules contained in the Basic Input/Output System (BIOS) of a computer to ensure that the Computer System starts in a trustworthy state. This device can also protect against Memory Management attacks including Buffer Overflows and Memory Pointer attacks.

INTRODUCTION

This chapter examines the idea of Trusted Computing which is using security principles like confidentiality and integrity to the hardware and software of a computer system. The Trusted Computing Base is a set of devices that have been authenticated at the trusted source. The idea behind Trusted Computing is to secure the software and hardware of a computer system in a manner that will protect the system from malware and rootkits. The protection of physical memory during execution plays an important role in Trusted Computing. This chapter discusses a special device called the Trusted Platform Module that secures the Computer System against security attacks.

In order to lessen the demand on the CPU, many devices have built-in processors that operate autonomously-disk controllers, coprocessors, network cards, and more. Since each of these devices contains processing elements that could be replaced by compromised and infected firmware, it is important that the bootstrap process of a computer not only include the verification of the main host CPU.
Secure Bootstrapping Using the Trusted Platform Module

(Central Processing Unit) elements but also any and all firmware-driven devices that are attached to the computer at boot-up time.

Arbaugh, Farber, and Smith (1997) describe the essentials of a secure inheritance-based trust model that begins with the bootstrap. In their design, the integrity of each step of the bootstrap process is validated with digital signatures that were created using asymmetric cryptography in a secure and trusted environment back at the code origin. This approach, of course, requires that the signatures be checked by some processor, most likely the CPU. The first firmware component to run is the BIOS, or basic input-output system. This is typically stored in some form of rewriteable storage, since the BIOS must be updated to accommodate bug fixes. As such, an obvious attack avenue for this sort of implementation would be the compromise of the bootstrap process, its digital signatures (if not stored in ROM or Read Only Memory), and the BIOS that must authenticate them. Clearly, storing the digital signatures in ROM does not ultimately provide sufficient protection against such an attack if both the BIOS and the bootstrap can be altered.

The Trusted Computing Group has supported the use of a secure hardware device that can independently verify the BIOS and its verification. This allows remote verification that the operating system began with the computer in a trustworthy state. The trusted platform module (TPM) is an example of such a hardware device. In operation, the TPM employs hashing to create state signatures of code modules. During secure initialization, the TPM creates and stores a small code segment, the core root of trust for measurement (CRTM) in secure storage called the platform configuration register (PCR). Whenever the computer is reset, the CRTM is run first. The CRTM measures all executable firmware (including the BIOS) against the hash codes that were stored in the PCR during secure initialization. If successful, control is then transferred to the BIOS (which is now trusted), and the BIOS repeats the hash verification process for the initial program loader (IPL). The trusted IPL then measures code that it will load, and the process continues in this fashion through the remaining modules, such as the kernel, device drivers, and applications.

The process reflects the inheritance procedure that begins with the root and consecutively adds signatures for subsequent hardware and software modules whose integrity and trust are required in order to create a trusted computing environment. At this point, a remote administrator can issue a request to the TPM device, asking for the signatures of each module that is stored in the PCR. This enables the administrator to verify that the system was loaded correctly and can be trusted. The TPM itself cannot be verified, but, as hardware, it is more difficult to attack. Thus the degree of difficulty in attacking such a system has been raised. However, probing of the TPM is not detected, nor is the replacement of the TPM with compromised hardware that can subvert the authentication process. This is sometimes referred to as a remote attestation challenge or, more simply, attestation. Its purpose is to supply the net result of the attestation and trust-inheritance process to a remote system that seeks to ascribe a level of trust to the platform in question. The attestation process should not compromise user privacy in any way or include the transmission of personal user information.

In general it is better to trust hardware than to trust either firmware or software. Properly designed hardware can include tamper-evidence or tamper-protection capabilities, and the cost of creating a fake hardware device without the tamper protection included (or, alternatively, disabled) is quite high and often requires significant facilities and technical skill. Firmware and software, in contrast, are easily distributed, making them a much easier target for someone intent on compromising the bootstrap process. In fact, viruses and Trojan horses can be created that target exactly these firmware and software modules.