Trusted Computing and Linux

Kylene Hall
IBM
kylene@us.ibm.com

Tom Lendacky
IBM
toml@us.ibm.com

Emily Ratliff
IBM
emilyr@us.ibm.com

Kent Yoder
IBM
yoder1@us.ibm.com

Abstract

While Trusted Computing and Linux® may seem antithetical on the surface, Linux users can benefit from the security features, including system integrity and key confidentiality, provided by Trusted Computing. The purpose of this paper is to discuss the work that has been done to enable Linux users to make use of their Trusted Platform Module (TPM) in a non-evil manner. The paper describes the individual software components that are required to enable the use of the TPM, including the TPM device driver and TrouSerS, the Trusted Software Stack, and TPM management. Key concerns with Trusted Computing are highlighted along with what the Trusted Computing Group has done and what individual TPM owners can do to mitigate these concerns. Example beneficial uses for individuals and enterprises are discussed including eCryptfs and GnuPG usage of the TPM. There is a tremendous opportunity for enhanced security through enabling projects to use the TPM so there is a discussion on the most promising avenues.

1 Introduction

The Trusted Computing Group (TCG) released the first set of hardware and software specifications shortly after the creation of that group in 2003.[1] This year, a short two years later, 20 million computers will be sold containing a Trusted Platform Module (TPM) [Mohamed], which will largely go unused. Despite the controversy surrounding abuses potentially enabled by the TPM, Linux has the opportunity to build controls into the enablement of the Trusted Computing technology to help the end user control the TPM and take advantage of security gains that can be made by exercising the TPM properly. This paper will cover the pieces needed for a Linux user to begin to make use of the TPM.

This paper is organized into sections covering the goals of Trusted Computing, a brief introduction to Trusted Computing, the components required to make an operating system a trusted operating system from the TCG perspective, the current state of Trusted Computing, uses of the TPM, clarification of common technical misperceptions, and finally concludes with

[1] See [Fisher] and [TCGFAQ] for more history of the Trusted Computing Group.

a section on future work.

2 Goals of Trusted Computing

The Trusted Computing Group (TCG) has created the Trusted Computing specifications in response to growing security problems in the technology field.

"The purpose of TCG is to develop, define, and promote open, vendor-neutral industry specifications for trusted computing. These include hardware building block and software interface specifications across multiple platforms and operating environments. Implementation of these specifications will help manage data and digital identities more securely, protecting them from external software attack and physical theft. TCG specifications can also provide capabilities that can be used for more secure remote access by the user and enable the user's system to be used as a security token."[TCGBackground]

Fundamentally, the goal of the Trusted Computing Group's specifications is to increase assurance of trust by adding a level of verifiability beyond what is provided by the operating system. This does not reduce the requirement for a secure operating system.

3 Introduction to Trusted Computing

The Trusted Computing Group (TCG) has released specifications about the Trusted Platform Module (TPM), which is a "smartcard-like device," one per platform, typically realized in hardware that has a small amount of both volatile and non-volatile storage and cryptographic execution engines. Figure 1 shows

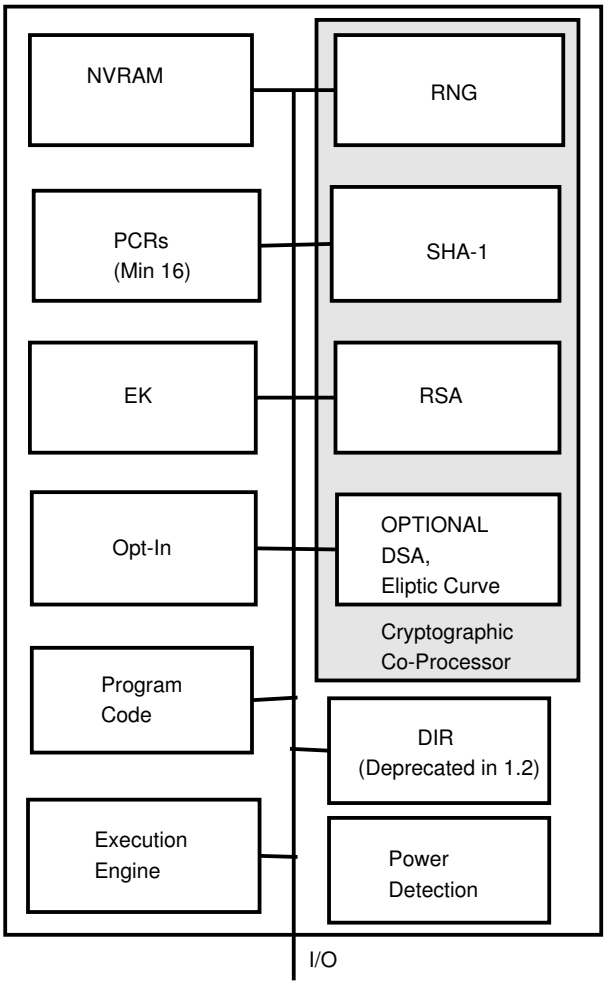

Figure 1: Trusted Platform Module

a logical view of a TPM. The TCG has also released a specification for APIs to allow programs to interact with the TPM. The next section details the components needed to create a completely enabled operating system. The interaction between the components is graphically shown in Figure 2.

For a rigorous treatment of Trusted Computing and how it compares to other hardware security designs, please read Sean W. Smith's "Trusted Computing Platforms Design and Applications" [Smith:2005].

3.1 Key Concepts

There are a few key concepts that are essential to understanding the Trusted Computing specifications.

3.1.1 Measurement

A measurement is a SHA-1 hash that is then stored in a Platform Configuration Register (PCR) within the TPM. Storing a value in a PCR can only be done through what is known as an extend operation. The extend operation takes the SHA-1 hash currently stored in the PCR, concatenates the new SHA-1 value to it, and performs a SHA-1 hash on that concatenated string. The resulting value is then stored in the PCR.

3.1.2 Roots of Trust

In the Trusted Computing Group's model, trusting the operating system is replaced by trusting the roots of trust. There are three roots of trust:

- root of trust for measurement
- root of trust for storage
- root of trust for reporting

The root of trust for measurement is the code that represents the "bottom turtle"[2]. The root of trust for measurement is not itself measured; it is expected to be very simple and immutable. It is the foundation of the chain of trust. It performs an initial PCR extend and then the performs the first measurement.

[2]This is an allusion to the folk knowledge of how the universe is supported. `http://en.wikipedia.org/wiki/Turtles_all_the_way_down`

The root of trust for storage is the area where the keys and platform measurements are stored. It is trusted to prevent tampering with this data.

The root of trust for reporting is the mechanism by which the measurements are reliably conveyed out of the root of trust for storage. This is the execution engine on the TPM.[TCGArch]

3.1.3 Chain of Trust

The chain of trust is a concept used by trusted computing that encompasses the idea that no code other than the root of trust for measurement may execute without first being measured. This is also known as transitive trust or inductive trust.

3.1.4 Attestation

Attestation is a mechanism for proving something about a system. The values of the PCRs are signed by an Attestation Identity Key and sent to the challenger along with the measurement log. To verify the results, the challenger must verify the signature, then verify the values of the PCRs by replaying the measurement log.

3.1.5 Binding Data to a TPM

Bound data is data that has been encrypted by a TPM using a key that is part of the root of trust for storage. Since the root of trust of storage is different for every TPM, the data can only be decrypted by the TPM that originally encrypted the data. If the key used is a migratable key, however, then it can be migrated to the root of trust for storage of a different TPM allowing the data to be decrypted by a different TPM.

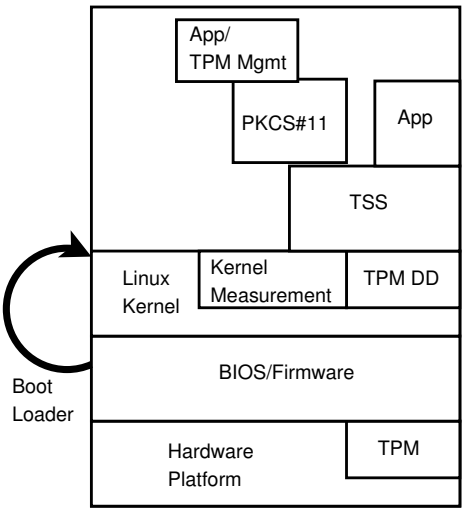

Figure 2: Trusted Computing Enabled Operating System

3.1.6 Sealing Data to a TPM

Sealed data is bound data that additionally records the values of selected PCRs at the time the data is encrypted. In addition to the restrictions associated with bound data, sealed data can only be decrypted when the selected PCRs have the same values they had at the time of encryption.

4 Components of Trusted Computing on Linux

Several components are required to enable an operating system to use the Trusted Computing concepts. These components are described in this section.

4.1 TPM

The Trusted Platform Module (TPM) is a hardware component that provides the ability to securely protect and store keys, certificates, passwords, and data in general. The TPM enables more secure storage of data through asymmetric key operations that include on-chip key generation (using a hardware random number generator), and public/private key pair encryption and signature operations. The TPM provides hardware-based protection of data because the private key used to protect the data is never exposed in the clear outside of the TPM. Additionally, the key is only valid on the TPM on which it was created unless created migratable and migrated by the user to a new TPM.

The TPM provides functionality to securely store hash values that represent platform configuration information. The secure reporting of these values, if authorized by the platform owner, enables verifiable attestation of a platform configuration. Data can also be protected under these values, requiring the platform to be in the same configuration to access the data as when the data was first protected.

The owner of the platform controls the TPM. There are initialization and management functions that allow the owner to turn on and off functionality, reset the TPM, and take ownership of the TPM. There are strong controls to protect the privacy of an owner and user.[3] The platform owner must opt-in. Any user, even if different from the owner, may opt-out.

Each TPM contains a unique Endorsement Key. This key can be used by a TPM owner to anonymously establish Attestation Identity Keys (AIKs). Since privacy concerns prevent the Endorsement Key from being used to sign data generated internally by the TPM, an AIK is used. An AIK is an alias to the Endorsement Key. The TPM owner controls the creation and activation of an AIK as well as the data associated with the AIK.[TCGMain],[TPM]

[3] See Section 7.1 for more details.

4.1.1 A Software-based TPM Emulator for Linux

If you don't have a machine that has a TPM but you'd like to start experimenting with Trusted Computing and the TSS API, a software TPM emulator can provide a development environment in which to test your program. While a software TPM will provide you with a development environment, it can't provide you with the "trust" that a hardware TPM can provide.

The advantage of having the TPM be a hardware component is the ability to begin measuring a system almost immediately at boot time. This is the start of the "chain of trust." By measuring as early in the boot cycle as possible, you lessen the chance that an untrusted component (hardware or software) can be introduced without being noticed. There must be an initial "trusted" measurement established, known as the root of trust for measurement, and the measurement "chain" must not be interruped.

With a software TPM emulator, you have delayed the initial measurement long into the boot cycle of the system. Many measurements have not occurred and so the trust of the system can not be fully validated. So while you would not want to rely on a software TPM to validate the trust of your systemm, it does provide you with a development environment to begin preparing to take advantage of trusted computing.

Mario Sasser, a student at the Swiss Federal Institute of Technology has created a TPM emulator that runs as a kernel module.[Strasser] It is not a full implementation of the specification and it is still under development. It is available from `http://www.infsec.ethz.ch/people/psevinc/` or `https://developer.berlios.de/projects/tpm-emulator`.

4.2 TPM Device Driver

The TPM device driver is a driver for the Linux kernel to communicate TPM commands and their results between the TCG Software Stack (TSS) and the TPM device. Today's TPMs are connected to the LPC bus. The TPM hardware is located by the driver from the PCI device for the LPC bus and attempts to read manufacturer specific information at manufacturer specific offsets from the standard TPM address. Since the TPM device can only handle one command at a time and the result must be cleared before another command is issued, the TPM device driver takes special care to provide that only one command is in-flight at a time and that the data is returned to only the requester. Rather than tie up all system resources with an ioctl, the command is transmitted and the result gathered into a driver buffer on a write call. Then the result is copied to the same user on a subsequent read call. This coupling of write and read calls is enforced by locks, the file structure's private data pointer and timeouts. At the direction of the Trusted Computing Group Specification, the TSS is the only interface allowed to communicate with the TPM thus, the driver only allows one open at a time, which is done by the TSS at boot time. The driver allows canceling an in-flight command with its sysfs file `cancel`. Other sysfs files provided by the driver are `pcrs` for reading current pcr values, `caps` for reading some basic capability information about the TPM such as manufacturer and version and `pubek` for reading the public portion of the Endorsement Key if allowed by the device. The current driver supports the Atmel and National Semiconductor version 1.1 TPMs, which are polled to determine when the result is available. The common functionality of the driver is in the `tpm` kernel module, and the vendor specifics are in a separate module. The driver is available on Sourceforge at `http://sourceforge.`

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.