Designing a Trust Chain for a Thin Client on a Live Linux CD

Megumi Nakamura
Tokyo Research Laboratory, IBM Japan
nakamegu@jp.ibm.com

Seiji Munetoh
Tokyo Research Laboratory, IBM Japan
munetoh@jp.ibm.com

ABSTRACT

CD-boot Linux[i] is a live Linux environment, which is easy to use because it is not installed in the hard disk, but simply boots directly from a CD. This helps protect the sensitive information because a clean environment can be prepared at boot time. To insure this environment protects sensitive information, we adapted the trusted computing technology to define a trustworthy environment.

Categories and Subject Descriptors

D.4.3[**Operating Systems**]: Security and Protection –*Verification*

General Terms

Security and Verification.

Keywords

Thin Client, Trusted Computing

1. INTRODUCTION

To use a live Linux environment as a thin client to protect sensitive information, we have to deal with some dangers of information leakage. Various methods have been proposed to check the integrity of the platform, but it is difficult to test it efficiently. We designed our CD-boot Linux by adapting trusted computing technology[1] to construct a trustworthy environment. Constructing the secure environment should not be a troublesome process. We propose an efficient process that checks the entire file system.

2. TRUST CHAIN

To make a system secure, we use various tools, such as a firewall, a virus scanner, secure channels, an IDS (Intrusion Detection System), and so on. These security tools are used according to the user's security policy. However such tools are useless if their behavior is manipulated by malicious software. Trusted computing technology deals with this problem by linking between trusted components. The falsification attack can be detected if the link is disrupted between the hardware and falsified component.

The Trusted Computing Group (TCG)[1] has defined a set of specifications for trusted computing technology. They use a hardware-based "root of trust" to test and report on the platform's integrity. The central module of the TCG architecture is the Trusted Platform Module (TPM). The TPM can be used to verify and report on the platform's integrity. The "trusted bootstrap" process is defined by the TCG to check the basic components on the platform during system initialization. When the system is turned on, the immutable initial bootstrap code verifies the next component and stores the verification in the TPM before transferring control to the next component. In subsequent steps, each component checks the next component and records the check in the TPM. Each component is measured by SHA-1 algorithm, and results are stored in Platform Configuration Registers (PCRs). These are tamper resistant registers within the TPM. A PCR can hold a digest for multiple components using a hash-chain mechanism. By looking at these values, the verifier can confirm the BIOS version, etc.

A secure thin client platform needs an architecture including the trust chain originated from the TPM so it is important to connect the trust chains for the security of the entire platform. It is necessary to design the client platform according to the TCG specification. It needs a TPM, a BIOS supporting the trusted bootstrap process, and a boot loader that supports TCG. The operating system includes a set of basic components such as a TPM driver and a TCG Software Stack, to support applications that need TPM functions. We targeted the creation of a live CD thin client because it can load everything from the boot loader to the OS while considering the trust chain, resulting in a trustworthy thin client.

3. INTEGRITY MEASUREMENT

For a live Linux CD, there are four crucial files. They are the boot loader, the kernel, initrd (which contains libraries and modules), and the root file system that contains the user space. After the trusted bootstrap process, Trusted GRUB[4] is started and checks the kernel and the modules. This GRUB verifies each component one by one. If we use the Integrity Measurement Architecture (IMA) [3], which is a Linux Security Module, then each executable file in the platform is checked when it is loaded into the kernel to execute. This mechanism allows recording and verifying all of the files loaded after the system bootstrap. A difficulty of validating with IMA is that a PCR value can change dynamically, so the value does not convey enough information.

A more efficient measurement approach is to check the entire file system at the time of the initial boot. The boot loader, kernel, initrd, and the entire file system as a loopback device are loaded from CD-ROM. Therefore, checking the each file in the CD-ROM at bootstrap time can reasonably assure the integrity of the entire system, and the attestation can be very simple.

SAC'07, March 11-15, 2007, Seoul, Korea.

We propose an efficient integrity check of the block file system because it takes too long time to read a large file to calculate the SHA-1 value. Such a one-shot-check takes about 7½ minutes before the system starts. To avoid this overhead time during the initial boot, the file system is measured dynamically as it is being accessed.

When a file is accessed, the kernel reads each block of data from the file system. In our proposal, we calculate the hash value of each block beforehand, and record them in a list in a file. The kernel can verify the integrity of each block using this list file. This list file must be protected from unauthorized modification. Therefore, Trusted GRUB checks the list file and records the hash value in a PCR. In this approach, the size of the files to be checked by the boot loader is reduced to 20 bytes per one block. When the block is read from the device, the hash value for the read block is calculated with the SHA-1 function of the kernel, and compared with the value recorded in the list file. If these two values are not the same, the kernel knows the read block has been tampered with, and it can take actions to stop reading the file and to block network access.

In this approach, the trust chain recorded in the TPM is end at the list file. The checking function and the list file dynamically verify the blocks as they loaded, and the trust chain is extended into the whole platform as a result.

4. PROTOTYPE

4.1 Implementation

We implemented the CD-boot client system in Knoppix Linux[2]. Knoppix is a bootable Linux based on Debian Linux. Knoppix uses a compressed loop-back block device (CLOOP) to read the file system on the CD-ROM, which is compressed using zlib. The CLOOP file is a read-only file system.

To access the TPM, the kernel needs to support the TPM device driver. We used kernel 2.6.17 that includes the TPM driver, and the TPM Software Stack that provides basic TPM management services. When the client boots, Trusted GRUB checks the kernel and initrd, and records the results into PCRs. The initrd contains the initial environment necessary for bootstrapping. Then a script file in initrd mounts the compressed CLOOP file as a loop-back file system.
When the CLOOP file is accessed, the CLOOP module reads and uncompresses it. Therefore, this module can check the proper blocks. We make a list of the hash values of all of the blocks in advance. We added a function to check the hash value and a function to compare the value with the value in the list.

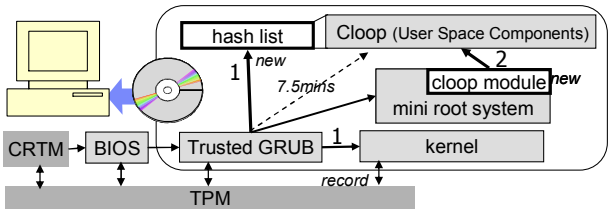

Figure 1: Integrity Measurement of CLOOP file

Figure 1 shows the verification chain. Formerly it took 7.5 minutes to check the CLOOP file. With the proposed approach, the boot loader only has to measure the list file of 300 KB. Detailed checks are done after the system starts. In those checks, the module uses the list which has already been checked by the boot loader.

4.2 Evaluation of Overhead Time

We evaluated this prototype with a server platform with a 1.6 GHz CPU and 2.0 GB memory, and with a client platform with a 2.0 GHz CPU and 512 MB memory. We evaluated three types, A) standard Knoppix, B) Knoppix with Trusted GRUB and C) Type B with the proposed verification process. Table 1 shows the times for each boot process: checking the CLOOP file, kernel, and initrd by Trusted GRUB, and then the start up of the OS. Type B shows that it takes 7.5 minutes to calculate the hash value of CLOOP file. In our approach, the list of hash values is checked instead of the entire CLOOP file. It takes two seconds to measure the list file while booting, as shown in Type C.
In our proposal, the hash value is calculated for each read block, and is compared with the value recorded in the list. Therefore, "starting up" of Type C takes longer in starting up than Type A and B. This extra time seems to come from the functions added to the CLOOP module.

Table 1: Overhead Time

	A: Original	**B: T-GRUB**	**C: Proposed**
CLOOP	-	**+7m 28s**	**+0m 02s**
kernel	0m 03s	+0m 07s	+0m 01s
initrd	0m 01s	+0m 01s	+0m 01s
starting up	**1m 34s**	**+0m 04s**	**+0m 06s**
Total	**1m 38s**	**+7m 40s**	**+0m 10s**

The time of type B and C is the overhead time compared with type A.

5. CONCLUSION

We have presented an occasional-use thin client system that assures users have a trusted operating environment. Our next step will be to develop the management mechanism of the integrity information to verify whether the file system meets specific security requirements.

6. ACKNOWLEDGMENTS

This study was sponsored by the Ministry of Economy, Trade and Industry, Japan (METI) under contract for the New-Generation Information Security R&D Program.

7. REFERENCES

[1] Trusted Computing Group (TCG), http://www.trustedcomputinggroup.org/

[2] Knoppix, http://www.knopper.net/knoppix/index-en.html

[3] R. Sailer, et.al. *Design and Implementation of a TCG-based Integrity Measurement Architecture.* USENIX Security Symposium 2004, pp.223-238.

[4] H. Maruyama, et.al. *Trusted Platform on Demand,* IBM Research Report RT0564, February 1, 2004.

[i] Linux is a trademark of Linus Torvalds in the United States and other countries.