一种不依赖TPM的安全引导方式的设计与实现

姚金魁,张 涛,王金双,陈 融,施祖清
(解放军理工大学 指挥自动化学院,江苏 南京 210007)

摘 要:在深入研究可信平台模块(TPM)和kexec相关技术的基础上,针对未安装TPM芯片的计算终端设计并实现了一种安全引导方式。该引导方式参考了可信平台模块的安全引导机制,在操作系统启动前,预先启动一个受保护的小型Linux系统,对操作系统的运行环境进行信任度量,验证BIOS和操作系统的完整性,为操作系统的启动提供一个可信赖的计算环境,然后使用kexec工具切换到磁盘操作系统。经过实验验证,文中设计的引导方式可以为Windows和Linux等多种平台的计算机终端提供安全引导支持,预引导系统验证系统运行环境的时延对系统启动总耗时的影响不明显。该引导方式是一种在不改变现有计算机硬件的基础上实现安全引导,是对现有计算机系统的安全增强措施。

关键词:信息安全;安全引导;预引导系统;kexec;完整性度量
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2012)06-0143-04

Design and Implementation of a Secure Boot without TPM Support

YAO Jin-kui, ZHANG Tao, WANG Jin-shuang, CHEN Rong, SHI Zu-qing
(College of Command Automation, PLA University of Technology, Nanjing 210007, China)

Abstract: Investigating in depth the technology of trusted platform module (TPM) and kexec, design and implement a secure boot mechanism for computers without TPM support. In this scheme, referencing the secure boot mechanism based on TPM, a small protected Linux will start first, which can then measure the real operating system needed to boot at the early stage. After the measurement, can boot the operating system on disk by utilizing kexec. The mechanism designed by this paper can support Windows and Linux according to the experiments, and it was also shown that no obvious burden has been added to the whole system boot time. This mechanism is one of secure boots for existing computer without changing hardware, is a kind of security enhancement measures for computer system.

Key words: information security; secure boot; pre-boot system; kexec; integrity measure

0 引 言

安全引导[1,2]是计算机终端安全的基础,其要求只有在上一步骤的引导实体确认下一步骤的引导实体完整性之后,才交出控制权。基于可信计算的安全引导以可信平台模块(Trusted Platform Module, TPM)[3]作为可信根,进而构建TPM到BIOS,BIOS到bootloader,bootloader到操作系统,再到所有应用程序的信任链[4]。

然而,目前仍有大量没有安装TPM芯片的计算机,文中针对这类计算机的安全引导问题,在不改变现有硬件的基础上,设计实现了一种安全引导方式。文献[5]通过便携式TPM完成可信引导,但是其要求计算机拥有EFI(Extensible Firmware Interface,可扩展固件接口),并需要修改部分固件代码,但是EFI尚未普及。文中通过先启动一个小型受保护的Linux操作系统(文中称之为预引导Linux系统,Pre-Boot Linux, PBL)来完成初始启动过程,然后对引导的操作系统内核等部件进行完整性度量,基于kexec[6,7]完成从受保护系统到待引导操作系统(主系统)的切换。文中基于X86体系结构实现了该引导方式,并分析了其对计算机性能的影响。

1 设计与实现

采用预引导系统启动计算机,其系统架构如图1所示,整个系统分为固件、只读文件系统、主文件系统3个部分。配置固件BIOS引导序列,设定首先启动的文件系统为只读文件系统。Bootloader和预引导系统存储在受保护的区域中,该区域为只读文件系统,预引导系统包含主系统内核和initrd的备份、BIOS的备份、完整性校验工具和系统切换工具kexec。主系统启动脚本提供一个类似于Grub启动菜单的交互界面供用户选择。

收稿日期:2011-11-15;**修回日期**:2012-02-18
基金项目:国家高技术研究发展计划"863"项目(2009AA01Z40)
作者简介:姚金魁(1979-),男,硕士研究生,研究方向为信息安全、可信计算;张 涛,博士,教授,研究方向为操作系统、信息安全、嵌入式系统。

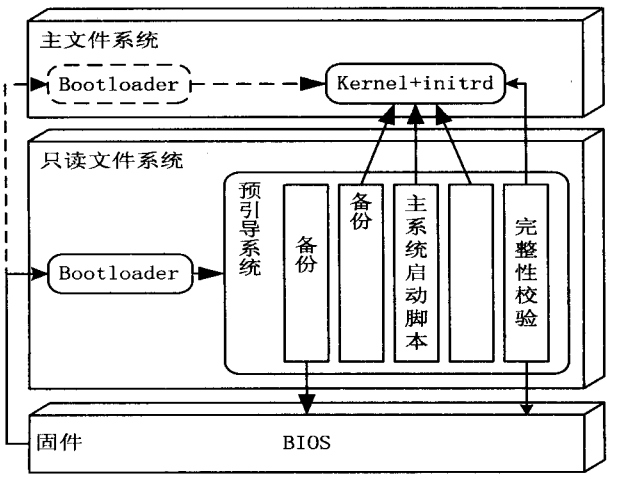

图 1 系统架构图

系统的引导流程如图 2 所示，系统加电后，BIOS 启动只读文件系统，bootloader 加载预引导系统的内核、initrd，进入预引导系统，自动运行主系统的启动脚本程序。然后度量 BIOS 的完整性，如果发现 BIOS 被篡改，使用 BIOS 备份恢复，如果用户不同意恢复，则关机处理，恢复 BIOS 后也应重新启动计算机，避免内存被驻留固件的恶意程序如 BIOS Rootkit[8] 的污染。若 BIOS 度量通过，则对需要启动的内核和 initrd 进行完整性度量，如果完整性度量通过则使用 kexec 启动主系统内核和 initrd，否则拒绝启动，并根据内核和 initrd 的备份恢复系统或者关机。

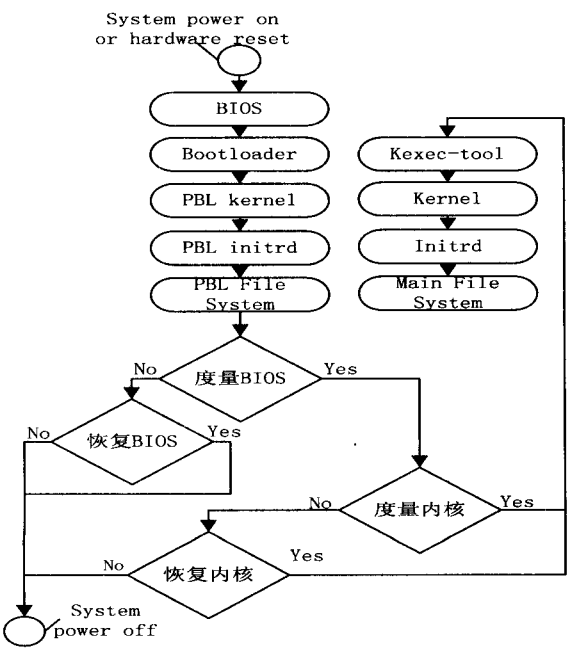

图 2 系统启动流程图

1.1 预引导系统

预引导系统在主系统启动前对计算机环境的安全性进行评估，相当于一个提供安全性能的 bootloader。要求启动和运行速度尽可能快，减小对整个系统启动时间耗费的影响。文中基于 Tiny Core Linux[9] 制作预引导系统，Tiny Core Linux 是一个小型的 Linux 系统，是遵循 GNU 许可的开源项目。其体积很小，一般 10MB 左右，运行在内存中，系统启动速度快。

1.1.1 系统定制

Tiny Core Linux 本身支持的硬件有限，不支持 kexec，需要对内核进行重新编译，使其能够支持 kexec 以及多种类型的文件系统和存储硬件。重新编译的内核去除不必要的功能和设备驱动可以减小内核镜像文件的大小、缩短读入内存的时间、提高内核的运行速度、缩短启动运行时间。initrd 采用 CPIO 文件格式，基于 busybox 制作，仅提供控制台工作环境，用于运行主系统启动脚本，以及提供信任度量、备份恢复等功能。主系统可能存在多个内核和相应的 initrd，预引导系统需要为用户提供选择菜单。文中自定义的 initrd 生成脚本可以分析主系统 grub 的配置文件 grub. cfg，备份主系统的内核和 initrd，并生成主系统启动脚本。主系统启动脚本使用 dialog 工具为用户提供交互界面，通过键盘选择启动项。预引导系统的 initrd 还包含信任度量工具、系统预存散列值文件、主系统内核和 initrd 备份、BIOS 备份二进制文件等。内核和 initrd 制作完成后，生成预引导系统的镜像文件。

1.1.2 备份和恢复 BIOS

要度量 BIOS 需要先获得其 flash ROM 中的内容，一种可行的方法是读取 flash ROM 输出到二进制文件，而后对该文件进行度量等操作。目前各个厂商生产的 BIOS 的型号种类繁多，ROM 的容量、ROM 在存储空间的映射地址存在差别，没有一种固定地址读取方式来操作所有的 BIOS。文中借鉴开源项目 Coreboot[10] 提供的 BIOS ROM 的支持库，其支持 296 种不同品牌和型号的 BIOS ROM。当需要读取 BIOS ROM 时，先创建 flashchip 结构体，使用函数 probe_flash( ) 遍历支持库，并匹配与该型号的项，再用该项初始化 flashchip，而后基于该 flashchip 进行后续的操作。flashchip 结构体的内容如下：

```
struct flashchip {
const char * vendor;//设备供应商
const char * name;//设备名称
enum chipbustype bustype;//总线类型
uint32_t manufacture_id; //产品编号
uint32_t model_id; //flash 设备型号
int total_size;//flash 的 ROM 容量
int page_size;//flash 设备页面大小
int feature_bits;//设备属性位
uint32_t tested;//设备是否已经被测试
int probe_timing;//操作时延
chipaddr virtual_memory;//虚拟内存地址
chipaddr virtual_registers;//虚拟寄存器地址
```

|;

系统进入保护模式后,BIOS 的地址位于地址空间的顶部,读取 BIOS ROM 时需要设置读取基地址,如果某型 BIOS 已经定义了 flashbase 就用该地址作为 BIOS 的基地址,如果未定义,则通过 BIOS ROM 的大小来定位基地址。如下语句描述了 32 位的地址空间中基地址的设置:

base = flashbase ? flashbase : (0xffffffff - size + 1);

读取 BIOS ROM 内容到二进制文件的流程如下所示:

1)初始化 flashchip 结构体;

2)识别 BIOS 型号,遍历 BIOS ROM 支持库,查找到相应的项,并用该项初始化 1)定义的结构体;

3)进行一次 BIOS 自检;

4)根据 ROM 大小分配内存空间;

5)从基地址 base 开始,以地址空间的顶端为结束地址,读取 ROM 内容到内存;

6)写入指定二进制文件;

7)释放 4)分配的内存空间。

将二进制文件写入 BIOS 的流程类似,不再赘述。

1.2 信任度量和系统切换

信任度量通过 SHA-1[11] 散列算法对 BIOS 以及主系统的内核和 initrd 进行完整性度量。系统初始化时,将文件名(含全路径)和相应的校验值保存在散列值存储文件 hash. properties 中,并写入预引导系统的 initrd,每次启动系统都将计算散列值并与预存值比较。从效率和安全性考虑,内核文件一次读入内存后,须完成度量、分析和引导的一系列操作。在 kexec 中扩充完整性度量功能,可以减少将内核与 initrd 文件读入内存的次数。

kexec 分为两部分,分别工作在用户态和内核态,用户态的部分可以分析和加载的 elf32 格式内核文件等功能,内核态的部分通过 reboot 系统调用来实现系统切换。文中实现的信任度量在 kexec 的用户态完成。

kexec 装载并启动新内核的步骤如下:

1)系统载入新的内核前,先分配相应的内存,其起始位置为驻留地址,地址空间限定为计算机的字长,只能在无符号长整型的取值范围内的地址空间存放新内核,并且不能和操作系统的保留地址冲突;

2)将分配的内存清零,读取内核文件到驻留地址;

3)检查内核文件是否符合 elf32 格式,因为 kexec 仅支持 elf32 格式的内核文件;

4)检查内核文件是否为适合当前系统架构的内核文件;

5)通过 kexec_load 系统调用执行内核的装载,并通知 kexec 内核态,装载成功,可以进行内核切换;

6)新内核运行时,关闭计算机外设、禁止中断、清空寄存器,转而执行一段由原内核保留的汇编代码(control_code_buffer),将新内核从驻留地址搬移到该架构下的指定地址,最后启动新内核,并把系统控制权交给新内核。

文中的实现是在步骤 2)后,加入信任度量功能。具体做法是:

a)读取文件 hash. properties 生成一个二维数组,后续操作将在该数组中查询预存散列值;

b)通过 1.1.2 的方法获取本机的 BIOS flash 内容的二进制文件,并与数组中相应的预存散列值进行比较,匹配则进入下一步,否则返回-1,并且中止 kexec 的内核装载;

c)对内存中驻留地址的内核文件计算散列值,与预存值比较,不匹配返回-10 且中止 kexec 的内核装载;

d)从主文件系统中读取和该内核文件匹配的 initrd 文件,计算散列值并与预存值比较,不匹配返回-100 且中止 kexec 的内核装载;

e)以上步骤通过,则返回 0。

若在 b)、c)、d)步骤中止,主系统启动脚本根据返回值提示用户是否进行相应的恢复操作;若在步骤 e)返回 0,则继续执行后续内核装载操作,信任度量完成。

1.3 对主存储区的安全增强

预引导系统制作完成后,原 Linux 系统的 bootloader 仍可引导系统,所以可以根据用户需求删除主系统 bootloader,或对整个引导分区进行加密,只有在预引导信任度量通过后才能在该环境中解密主系统分区。

1.4 主系统为 Windows 的引导

若主系统为 Windows 操作系统,引导的流程以及信任度量方式与 Linux 有较大差异。需要度量的系统文件是 NTLDR,并度量 MBR(Master Boot Record, 主引导记录),而在 Windows 的引导序列中,NTLDR 对后续文件的信任度量不在文中讨论的范围中。由于 kexec 仅支持装载 elf32 格式的内核文件,而 Windows 不支持该格式的内核文件,所以需要借助第三方 bootloader,文中借助的第三方工具是 GRUB4DOS[12]。GRUB4DOS 是一个以 GNU GRUB 为基础的 bootloader,可以在 DOS 和 Linux 下运行,也可以通过其他 bootloader 加载后运行。使用预引导系统启动 Windows 的系统架构如图 3 所示,预引导系统在完整性校验后,使用 kexec 启动 GRUB4DOS,再引导 NTLDR(或者 MBR),进而引导

Windows 系统。

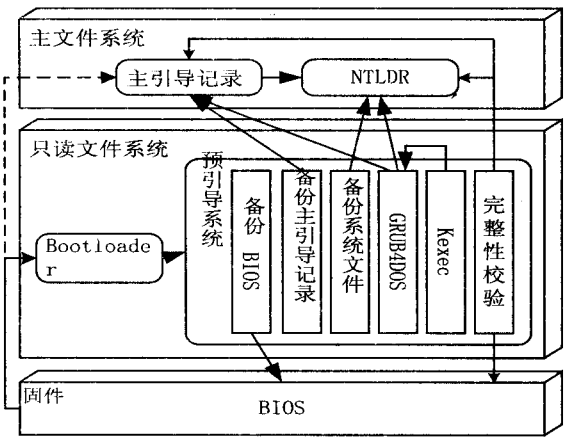

图 3 引导 Windows 的系统架构图

如果采用安装在 USB 存储设备上的预引导系统启动 Windows，USB 设备可能被识别成第一磁盘(hd0)，原第一硬盘将被识别成第二磁盘(hd1)，由于 Windows 限制引导非安装在第一磁盘的系统，所以在配置 GRUB4DOS 时，需要对磁盘进行映射，使用的 GRUB4DOS 命令序列如下：

```
map(hd0)(hd1)
map(hd1)(hd0)
rootnoverify(hd1,0)
makeactive
```

若将预引导系统存放在光盘中，不需要进行上述磁盘映射的操作。

GRUB4DOS 加载 Windows 有 2 种方式，一种是直接加载 ntldr 文件(注：ntldr 是 Windows NT 5.x 系列操作系统的引导文件，而 Windows NT 6.x 系列则使用文件 bootmgr)，另一种是先加载磁盘的 MBR，再由 MBR 来加载 ntldr 文件，在 GRUB4DOS 中分别使用命令 chainloader +1 与 chainloader /ntldr 实现上述 2 种加载方式。文中支持上述 2 种加载方式，以提高加载成功率。

2 结束语

文中设计的安全引导方式在虚拟机 VirtualBox 和实际的 PC 机上实现，其中 PC 机采用 USB-CD 启动，分析安全引导增加的系统启动时延。虚拟机的平均增加启动时延在 10 秒以内，在 PC 机上实验，平均时延为 18 秒(不含等待用户选择的时间)。实验表明文中的安全引导的方式对整个系统的启动时间的影响是可以接受的。

参考文献：

[1] Parno B. Bootstrapping trust in a "trusted" platform[C]// Proceedings of the 3rd conference on hot topics in security. Berkeley, CA, USA: USENIX Association, 2008.

[2] 陈书义，闻英友，赵 宏. 基于条件谓词逻辑的可信计算形式化分析[J]. 华南理工大学学报：自然科学版，2009，37(5)：106-110.

[3] TPM Main Specification Level 2 Version 1.2, Revision 116 [EB/OL]. (2011-03-01)[2011-11-14]. http://www.trustedcomputinggroup.org/developers/trusted_platform_module/specifications.

[4] 陈建勋，侯方勇，李 磊. 可信计算研究[J]. 计算机技术与发展，2010，20(9)：1-4.

[5] 张 颖，周长胜. EFI 下基于便携式 TPM 的可信计算平台研究[J]. 计算机技术与发展，2010，20(1)：167-171.

[6] Horman S. Kexec[EB/OL]. 2010[2011-11-14]. http://horms.net/projects/kexec/.

[7] Nellitheertha H. Reboot Linux faster using kexec[EB/OL]. 2004[2011-11-14]. http://www.ibm.com/developerworks/linux/library/l-kexec/index.html.

[8] Heasman J. Implementing and Detecting an ACPI BIOS Rootkit[C]//Blackhat Federal 2006. Washington, DC: [s.n.], 2006.

[9] Shingledecker R. Tiny Core Linux[EB/OL]. 2008[2011-11-14]. http://www.tinycorelinux.com.

[10] Advanced Computing Laboratory at Los Alamos National Laboratory (LANL). Coreboot[EB/OL]. (2011-06-24)[2011-11-14]. http://www.coreboot.org/.

[11] Eastlake D, Jones P. US Secure Hash Algorithm 1 (SHA1)[S]. Internet RFC 3174, 2001.

[12] GRUB4DOS and WINGRUB[EB/OL]. 2009[2011-11-14]. http://grub4dos.sourceforge.net/.

(上接第 138 页)

[11] de Rauglaudre D. Camlp5-Reference Manual[M]. Ville de Paris: Institut National de Recherche en Informatique et Automatique, 2008: 45-61.

[12] 刘 全，孙吉贵. 基于语义 tableau 的一阶逻辑自动定理证明[J]. 计算机工程与应用，2005(23)：22-24.

[13] 刘 全. 基于 tableau 的自动推理研究[D]. 长春：吉林大学，2004.

[14] 陆汝钤. 强有序输入消解原理[J]. 中国科学，1981(8)：1035-1042.

[15] 王元元. 计算机科学中的现代逻辑学[M]. 北京：科学出版社，2001：74-100.

[16] 张灵峰，夏战锋，彭志平. 基于 Tbox 和 Abox 的描述逻辑推理研究[J]. 计算机技术与发展，2010，20(11)：122-125.