可信计算平台中 Linux加密文件系统的设计与实现

罗 芳,徐 宁,孟 璟,刘雪峰

(信息工程大学 电子技术学院,河南 郑州 450004)

摘要:加密文件系统是保护用户敏感数据的一种有效手段,传统的加密文件系统对用户数据的保护并不是非常可靠且存在被篡改的可能性。文章基于可信计算平台和 Linux2.6内核设计,实现了一个堆栈式加密文件系统 TEFS(Trusted Computing-based Encrypted file system),该加密文件系统利用可信计算技术为其提供底层安全支撑,并结合可信计算平台中安全增强的操作系统定制、实施安全策略,进一步提高了其安全性。

关键词:可信计算平台;可信平台模块;加密文件系统;安全增强的 Linux操作系统

中图分类号:TP309 **文献标识码**:A **文章编号**:1671-0673(2008)02-0225-04

Design and Implementation of an Encrypted Filesystem for Linux in Trusted Computing Platform

LUO Fang, XU Ning, MENG Jing, LIU Xue-feng

(Institute of Electronic Technology, Information Engineering University, Zhengzhou 450004, China)

Abstract: Encrypted filesystem is an effective method to protect sensitive data. Protecting sensitive data through traditional encrypted filesystem isn't reliable and can be tampered, so based on TCP and the core of Linux2.6, a stackable encrypted filesystem TEFS is designed and implemented which enhances its security by the Trusted computing technology and improves its security by implementing policy through SELinux in TCP.

Key words: TCP; TPM; encrypted filesystem; SELinux

加密文件系统作为访问控制的有力补充手段,能够为计算机系统中存储的大量机密数据提供一种简便、有效的数据加密存储服务。传统的加密文件系统对用户数据的保护基本上是以软件为基础,并附以密钥技术来完成的,事实证明这种保护并不是非常可靠而且存在着被篡改的可能性。

可信计算平台(TCP)基于可信平台模块(TPM),以密码技术为支持、安全操作系统为核心[1],通过将加、解密和认证等基本安全功能写入 TPM,确保了其中的信息不能在外部通过软件随意获取。因此,对于数据安全要求较高的加密文件系统而言,可信计算平台能够为其实现提供基于硬件的保护和底层的安全支撑。

1 对已有加密文件系统的分析

传统的 Linux加密文件系统可分为:块加密文件系统、磁盘加密文件系统、网络加密文件系统和堆栈式加密文件系统。这些传统加密文件系统中,多数(如 EncFS和 CFS等)都是将加密元数据直接存储在挂载点的特殊文件内,因此解密时需要将元数据和加密文件一起拷贝,增加了系统开销。此外,用户的根密钥多是根据用户口令由单向函数变换生成,容易遭到攻击,事实上,将可信计算技术引入加密文件系统后,可将解密的唯一入口——根密钥保存在 TPM内部的屏蔽区域,独立于计算机的

收稿日期:2008-01-15;修回日期:2008-03-11

基金项目:国家 863计划资助项目(2006AA01Z433)

作者简介:罗 芳(1983-),女,硕士生,主要研究方向为信息安全和可信计算。

存储系统(如硬盘),这样传统的攻击方法就难以窃取密钥,提高了系统的安全性。正是由于可信计算技术对于加密文件系统的开发具有上述优势,使得其在加密文件系统中的应用而回益受到重视。微软和兆日公司都推出了其基于可信计算平台开发的加密文件系统。

但是,目前多数基于可信计算平台的加密文件系统大多属于块加密文件系统,其加密粒度过粗,实际上,在多数情况下,文件系统中只有某些文件需要加密,系统库和可执行文件则不必加密,因此,将密钥映射到单个文件,而不是整个块设备和整个目录中,只对那些需要的文件加密可以有效节省系统资源。

2 TEFS的设计

2.1 设计目标

在对以上加密文件系统分析的基础上,本文基于 Linux2.6内核、TPM_Emulator的仿真环境和Trousers设计,实现了一个堆栈式加密文件系统TEFS,以期为用户提供一种基于文件粒度的,透明的加、解密服务,并与可信计算技术相结合为用户提供合理的密钥管理机制,支持文件的共享,同时,依托可信计算平台对其进行安全增强,在保证安全的前提下,兼顾系统性能,尽可能地降低系统负载。

2.2 工作原理

TEFS是采用堰栈式文件系统工作原理实现的一种加密文件系统。它通过将加、解密功能加载到原有文件系统之上以实现递增式开发,即对文件的加、解密操作都由堆栈式文件系统来提供,而基本不涉及底层具体文件系统。因此,它具有良好的可扩展性,能够有效降低内核复杂度,并且避免了在内核空间和用户空间来回切换导致的性能损耗,如图1所示。

TEFS工作于Linux虚拟文件系统VFS(Virtual Filesystem Switch)和底层具体文件系统之间。对于VFS,TEFS是一个普通的文件系统,它接收VFS发送过来的函数调用和数据,之后,把用户传来的数据加密后传递给底层的具体文件系统存储;对于底层文件系统返回的数据,加密文件系统TEFS对其解密,然后返回给VFS,VFS再返回给用户,从而为用户提供一种透明的加、解密服务。因此,TEFS是叠加在一个底层文件系统之上的加密文件系统,它为用户提供了安全和透明的数据加密存储服务。

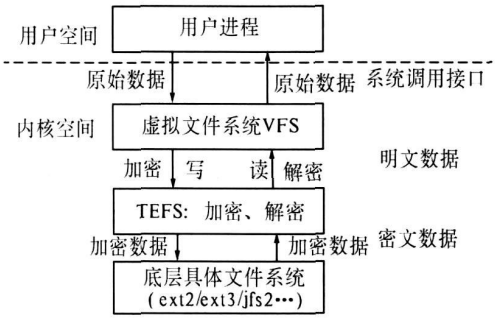

图1 TEFS工作原理图

2.3 TEFS体系结构

TEFS运行于可信计算平台环境中,与内核的Crypto API,keyring,用户空间的TEFS-daemon,底层具体文件系统(ext2/ext3/jfs..)以及可信软件堆栈TSS(Trusted Software Stack)交互来完成加、解密操作,其体系结构如图2所示。在使用TEFS之前,用户应首先向TPM中植入平台拥有者的授权数据来取得平台的拥有权,并向内核插入TEFS模块,启动TEFS-daemon守护进程,随后挂载TEFS,在挂载时需要指明用户加密密钥所对应的唯一标识符(UUID),在完成上述操作后,用户将文件写入加密文件目录,则可完成对文件的自动加密。

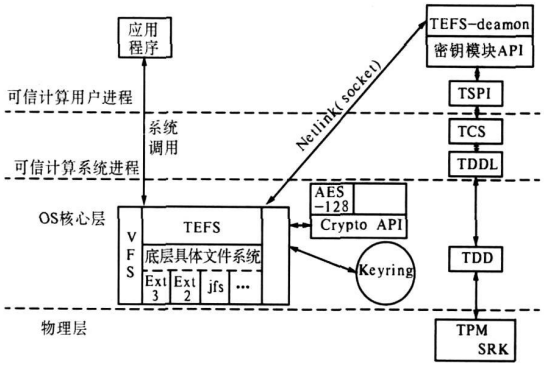

图2 TEFS体系结构图

TEFS的体系结构主要分成两部分,分别是内核可加载模块TEFS和用户空间的TEFS-daemon,两者之间通过Netlink(socket)进行通信。其中具体的加、解密以及挂载(包括对挂载参数的解析)等操作由TEFS内核可加载模块完成,该模块在需要的时候通过modprobe命令动态加载到内核中。在内核空间进行加、解密操作,能够避免用户和内核之间频繁的上下文切换,从而有效提高TEFS的运行效率。

运行于用户空间的守护进程TEFS-daemon接

收来自内核模块的请求, 通过 TSS与 TPM交互来完成密钥管理功能, 由于 Linux正向微内核发展, 因此, 将密钥管理功能置于用户空间可以降低内核空间运行代码的复杂度。

TEFS还采用了 Linux2. 6内核中新增的密钥保留服务 keyring来提供密钥缓存机制, 提高其密钥搜索速度。在 TEFS中, 利用 keyring存储授权令牌, inode的加密上下文和密钥等信息。当对文件加密时, 内核 keyring向用户空间的守护进程发起 request_key请求, TEFS-daemon在接到请求后与 TPM交互, 并将生成的密钥存储于用户的会话 keyring中, TEFS从中取得密钥, 利用 Linux内核提供的密码算法和 Crypto API对文件内容进行加、解密操作。可见, keyring为用户应用程序提供了一个空间使得其密钥能在稍后被 TEFS内核模块使用。

3　关键技术的实现

3. 1　TEFS的加、解密流程

在 TEFS中定义了 3类密钥, 分别是文件加密密钥 FEK(File Encryption Key)、用户加密密钥 UEK(User Encryption Key)和 TPM非易失性存储区中的存储根密钥 SRK(Storage Root Key)。TEFS中的每个文件都唯一对应一个文件加密密钥 FEK, FEK是一个对称密钥, 在文件创建时由 Linux内核的 get_random_bytes( )系统调用产生, 其长度由内核所采用的对称密码算法确定。用户加密密钥 UEK由用户调用 TSS接口函数: Tspi_key_createkey( )通过 TPM生成, 并由 TPM向用户返回 UEK所对应的 UUID。存储根密钥 SRK用于对 UEK进行密封存储操作, 密封后的数据块存储于用户永久性存储区 TSS_USER_PS_FILE(PS文件)中, UEK在用户永久性存储区中的存储结构如下所示。

```
Struct uek_mapper{
    TSS_UUID uuid;
    TSS_HKEY hkey;
    Struct key_mapper * next;
} * mapper= Null
```

TEFS通过 FEK和 Linux内核的 Crypto API, 采用密码分组链接 CBC(Cipher Block Chain)模式, 对文件数据块进行加、解密操作, 如图 3所示。

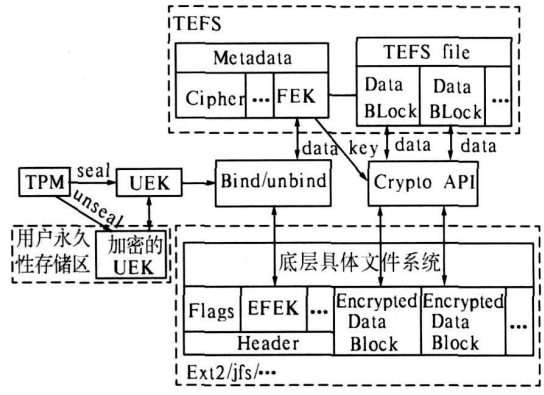

图 3　TEFS加、解密流程图

FEK则由 UEK进行绑定加密后生成 EFEK, 存储于 TEFS文件首部 Header中。这样, 用户访问加密文件所需的信息在文件首部就可直接获得, 有助于提高 TEFS的加、解密效率。TEFS文件首部 Header的格式如图 4所示。

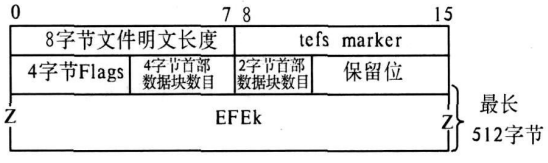

图 4　TEFS文件首部 Header格式

整个加、解密流程的最后一环 UEK的机密性是由 SRK通过密封存储操作来确保的, 可见层层加密的最后一环是 SRK, 并由 TPM对其提供硬件保护。

此外, 为了实现加密文件在一组可信计算平台之间的共享, TEFS应使可信核心服务模块 TCS(Trusted Core Service)的后台程序 TCS-daemon能够接收远程连接请求, 并建立连接。通过设置 TSS的配置文件 tcsd. conf中 remot_ops项来开启 TCS-daemon的远程连接服务, 从而在一组可信计算平台之间共享用户加密密钥 UEK, 一旦同组用户共享该密钥, 则可以以任何方式交换他们的文件。

3. 2　TEFS与 TCP的交互

TEFS在生成用户加密密钥 UEK以及进行密封存储、绑定操作时都需要通过 TSS与 TCP进行交互。其中密封存储提供了对密钥的硬件保护, 用户通过对 TPM进行配置, 即若机器不是以确定的配置方式启动, TPM将拒绝解密数据, 这样可以防止从不可信的媒体启动从而造成的攻击。

TEFS与 TCP的交互过程大致如下: 可信服务提供层 TSP(Trusted Service Provider)为 TEFS提供接口, 它把来自加密文件系统的参数打包传给可信核心服务模块 TCS, 由 TCS来提供具体的功能函数

(例如密钥管理); TCS把来自 TSP的参数进行分析和操作以后生成一个 TPM可以识别的字节流,通过可信设备驱动库 TDDL(Trusted Device Drive Library)传送给 TPM; TPM接收到字节流以后进行相应的操作,例如生成用户唯一标识符 UUID或产生与当前配置有关的加密数据块,并把结果以字节流的形式通过 TDDI返回给 TCS; TCS对字节流解析以后,把结果传给 TSP,由 TSP把正式的结果返回给 TEFS。

3.3 TEFS安全性增强研究

TEFS在运行过程中可能存在一些安全威胁,例如非法用户可能杀死 TEFS-daemon守护进程,或将直接存储于底层具体文件系统中的文件密文、PS文件或库文件删除、替换,这些恶意行为将使 TEFS不能提供有效的数据机密性保护。事实上,如果仅依赖应用层的安全措施与密钥管理功能,而不通过可信计算平台中安全增强的操作系统保护数据文件,不以可信计算平台为基础,数据加密就没有真正的安全性可言。

对于上述安全威胁,可信计算平台能够从可信度量根 CRTM开始,依次对硬件平台、操作系统内核以及上层应用系统进行完整性度量,确保系统的正常启动,并对 TPM芯片内部存储的密钥、数据以及数字证书提供保护,从而有效阻止非法程序对 TEFS的攻击和破坏,保证其可靠、稳定地运行。

此外,利用可信计算平台中安全增强的操作系统为 TEFS实施安全策略,也能够有效防止非法用户的安全威胁。例如,TEFS的 PS文件中存储了用户加密密钥 UEK的密文,需要对其完整性进行保护,策略配置为其定义 PS_类型,为需要更新此文件的程序(seal,unseal等)分别分配单独的域,且仅授权这些域有写 PS文件的访问权限,从而可以防止恶意代码对该文件的篡改。以下是具体策略:

```
Allow local_seal_tPS_t: file | read write;
Allow remote_seal_tPS_t: file | read write;
Allow unseal_tPS_t: file | read write;
...
```

在完成 TEFS的策略规则编写后,通过 SELinux的 m4宏处理语言生成配置文件 policy.conf,再经 checkpolicy命令将配置文件编译成二进制策略文件,以 LSM模块的形式载入内核空间,并将其作为系统策略的一部分执行,以增强 TEFS的安全性。

4 性能测试

本文对 TEFS的工作性能进行了测试,目的是将加密文件系统 TEFS与普通堆栈式加密文件系统 Wrapfs和 ext3文件系统进行比较,以确定 TEFS与 TPM交互所带来的性能损耗程度。测试环境为:奔腾Ⅳ3.06 GHz处理器/512M内存,操作系统为中标普华 Linux安全增强操作系统(内核版本号为 2.6.21),并采用 TPM仿真模块 TPM_Emulator0.5、TCG软件堆栈 Trousers0.3和 Linux内核提供的密码算法 AES-128。测试内容是关于 3种文件系统的数据读、写速度的测量,数据量为 16M,读写单位为 1M,使用测试程序 IOZONE对以上 3种文件系统的读、写速度测试结果如表 1所示。

表 1 TEFS文件读、写速度比较

	数据量	消耗时间(S)		速度(MB/S)	
	(MB)	读	写	读	写
Ext3	16×1MB	0.125	0.274	127.18	58.24
Wrapfs	16×1MB	0.350	3.58	45.71	4.47
TEFS	16×1MB	0.396	3.78	40.41	4.23

从表 1可知,与 ext3文件系统相比,TEFS的读、写速度分别下降了 69.01%和 92.73%;与堆栈式文件系统 Wrapfs相比,TEFS的读、写速度分别下降了 11.59%和 5.37%,可见,TEFS与 TPM的交互虽然会带来一定的性能损耗,但与同类文件系统相比,其性能损耗并不是十分显著。

5 结束语

本文设计、实现了一个基于可信计算平台的加密文件系统 TEFS,论述了其工作原理和关键技术,并对其工作效率进行了测试。与传统加密文件系统相比,TEFS应用可信计算技术为其提供硬件保护和底层安全支撑,通过密封存储等技术来防止从不可信媒体启动所造成的攻击,并结合可信计算平台中的 SELinux为其实施安全策略,进一步提高了其安全性。下一步拟对 TEFS的性能进行优化,进一步提高其加、解密速度。

参考文献:

[1] 沈昌祥. 可信计算平台与安全操作系统[J]. 网络安全, 2005(4): 8-9.

[2] 毛德操, 胡希明. Linux内核源代码情景分析[M]. 杭州: 浙江大学出版社, 2001.

[3] TCG. TCG Software Stack(TSS) Specification Version 1.2 Level1[DB/OL]. [2006-01-10]. http://www.trustedcomputing.org/downloads/TSS_1_2_Change_final.pdf

[4] Blaze M A. Cryptographic File System for Unix[C]// First ACM Conference on Communication and Computing Security. Fairfax VA, 1993: 158-165.