并行分类信任链传递模型

林基艳,吴振强
LIN Ji-yan,WU Zhen-qiang

陕西师范大学 计算机科学学院,西安 710062
School of Computer Science,Shanxi Normal University,Xi'an 710062,China
E-mail:linjiyan@stu.snnu.edu.cn

LIN Ji-yan,WU Zhen-qiang.Parallel sorting model of chain of trust.Computer Engineering and Applications,2009,45(31):98-101.

Abstract: At present,chain model adopts serial integrity measurement means to solve the problem of the transfer on application layer which can increase the overhead time and affect the efficiency of the system.This paper brings forward a new chain of trust model which uses virtualization technology to advance a method called parallel sorting integrity measurement which can reduce the overhead time of the integrity measurement and decide which program can run.In the end,formal verification is given to the presented model which shows the new model can meet the requirement of the trust transfer.

Key words: trusted computing;parallel sorting integrity measurement;model of chain of trust;formal verification

摘 要:目前的信任链传递模型在解决应用层的信任传递问题上采用串行完整性度量方案,增加了系统的时间开销,影响系统效率。论文提出了一种新的信任链传递模型,该模型利用虚拟化技术提出并行分类完整性度量方法,降低了应用层可信传递过程中的完整性度量时间开销,同时用户也可以根据自己的需求决定是否允许应用程序运行;通过对所提模型的形式化验证,表明新模型满足可信传递需求。

关键词:可信计算;并行分类完整性度量;信任链传递模型;形式化验证
DOI:10.3778/j.issn.1002-8331.2009.31.029 **文章编号**:1002-8331(2009)31-0098-04 **文献标识码**:A **中图分类号**:TP309

1 引言

可信计算的基本思想是在计算机系统中首先建立一个信任根,再建立一条信任链,一级度量认证一级,一级信任一级,把信任关系扩大到整个计算机系统,从而确保计算机系统的可信[1-3]。但是这样也给系统带来了额外的时间开销,特别是当信任链传递到应用层时,为了保证应用层的可信,不仅要对所有待运行的应用程序进行完整性检查,还要对应用程序加载的动态共享库、二进制文件等所有可能影响应用程序可信性的组件进行度量[4],由此带来的时间开销降低了系统运行效率,因此设计一种更加高效安全的信任链传递模型具有重要的意义。

Demetrios Lambrou[5]提出了基于 LSM 的完整性度量方案,该方案在信任传递到应用层后, 利用 LSM 的这些钩子函数来验证内核模块、可执行文件是否可信;IMA[6](Integrity Measure Architecture) 是 IBM 的 Reiner Sailer 等人开发的基于 TCG 的可扩展的完整性度量结构。IMA 在 Linux 系统上实现,它在信任传递到应用层后,通过对系统中的可执行文件、动态加载器、内核模块以及可执行脚本进行度量来保证系统运行时的完整性。但是这两种方案都采用了串行度量的方式,完整性度量时间开销大。

该文利用虚拟化技术提出了一种并行分类信任链传递模型,模型利用虚拟化技术提出了应用层并行分类完整性度量方法,即在宿主操作系统上构建三个虚拟操作系统(Guest OS),利用这些虚拟的操作系统来度量影响应用程序完整性的相关组件(动态共享库,配置文件,解释型对象——可执行脚本,宏等),在提高了硬件利用率的同时,实现了对组件的并行分类度量,提高了完整性度量的效率;同时用户也可以根据自己的需求决定是否允许应用程序运行;最后对提出的模型进行了形式化验证。

2 基于虚拟化技术的并行分类信任链传递模型

2.1 并行分类完整性度量

多核技术是目前整个 IT 产业关注的一个关键词, 随着硬件产品在多核技术上的逐渐就绪,利用虚拟化技术能进一步发掘应用的时间和空间的并行性,全面释放多核处理器的性能潜

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60633020);西安电子科技大学计算机网络与信息安全教育部重点实验室开放课题(No.2007CNIS-06)。

作者简介:林基艳(1986-),女,硕士研究生,主要研究兴趣为可信计算、无线网络安全;吴振强(1968-),男,博士,副教授,主要研究方向为可信计算、匿名通信技术、自适应安全。

收稿日期:2009-05-26 **修回日期**:2009-07-13

力。因此引入了虚拟化技术,提出了并行分类完整性度量的方法。利用虚拟机软件如 VMWare 在宿主操作系统 Linux 上构建三个虚拟操作系统(Guest OS),各系统间相互独立运行,不干扰,用这些 Guest OS 来度量影响应用程序完整性的组件——动态共享库的度量在 Guest OS1 中进行,配置文件的度量在 Guest OS2 中进行,解释型对象的度量在 Guest OS3 中进行;可执行代码及内核模块的度量在宿主操作系统中进行,Guest OS 度量完相应的组件后,要将被度量的组件标记为不可修改,只有当组件被使用后才能把标记改回可修改状态。

用户可以根据自己的需要来决定是否允许程序运行——如果用户只关心其中一个组件如动态共享库的完整性,那么只需要根据 Guest OS1 的度量结果来判定是否允许程序运行,依此类推;如果用户关心其中两个组件的完整性,如动态共享库和配置文件,那么只要这两个组件的完整性没被破坏,用户就可以运行程序运行;而如果用户对这三个组件的完整性都关心,那么只能三个组件都完整的情况下,程序才可以运行,否则禁止运行。在并行分类完整性度量方案中,如果度量动态共享库的时间为 t_d,度量配置文件的时间为 t_c,度量解释型对象的时间为 t_s,那么应用程序完整性度量所需的时间为三者中的最大者,即 $t=\max(t_d,t_c,t_s)$;而文献[5-6]所需的时间 $t=t_d+t_c+t_s$,由此可见,该方案降低了完整性度量所需的时间。系统的体系结构如图 1 所示。

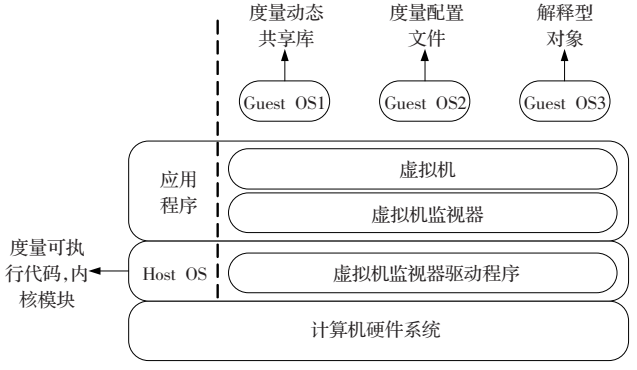

图 1　系统体系结构

2.2　基于虚拟化技术的并行分类信任链传递模型

由 1.1 节可以看出,TCG 规范中的信任链传递模型[7]已经不适用了,因此需要重新构造信任链。基于虚拟化技术的并行分类信任链模型如图 2 所示。

该模型在终端 PC 上通过完整性度量进行可信传递,传递包括两个阶段:第一阶段是从终端加电到操作系统装载(系统引导)。该阶段是一个顺序固定的单一链式过程,而且 BIOS、主引导记录 MBR、操作系统装载器 GRUB 以及操作系统内核一般相对稳定,在此可信传递过程的完整性度量相对容易[8];第二阶段是从宿主操作系统到宿主应用层以及宿主操作系统到虚拟操作系统的传递过程,下面给出上两个阶段的设计方案。

(1)系统引导阶段

虚拟机运行在宿主机内,由虚拟机管理工具提供服务支持。因此若其宿主机不可信,则在其中运行的虚拟机一定也不可信,即可信的宿主机是虚拟机可信验证的前提条件,而宿主机的可信则由系统引导阶段来保证,系统引导具体过程如图 3 所示。

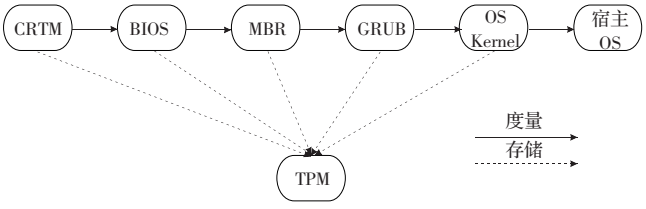

图 3　可信引导过程

①用户在可信硬件提供的 USB 接口中插入开机身份卡;

②硬件平台加电,TPM 首先复位并进行初始化;

③TPM 对用户的开机身份卡进行认证,如果通过开机身份认证,则进入可信引导过程(否则停机或进入普通引导过程);

④CRTM 验证支持可信计算的 BIOS 的完整性,如果验证通过,把执行权交给 BIOS;

⑤在 TPM 芯片的帮助下 BIOS 验证 MBR(主引导扇区)的完整性,通过之后把执行权交给 MBR;

⑥MBR 验证 GRUB 的完整性,验证通过则把执行权交给 GRUB;GRUB 验证 OS Kernel 的完整性,验证通过后把执行权交给 OS Kernel。

(2)应用可信传递阶段

应用可信传递阶段用来保证从宿主操作系统到宿主应用层的传递过程以及从宿主操作系统到虚拟机监视器 VMM,到虚拟操作系统传递过程的可信。该阶段具有多样性和无序性等特点,如何对影响应用层可信性的组件进行度量是个难点,论文在文献[4-6]的基础上,基于 Linux 安全模块 LSM(Linux Security Modules)来建立 Linux 平台下的应用可信传递模型。在 Linux 启动后,可能改变系统可信状态的有内核及其模块,

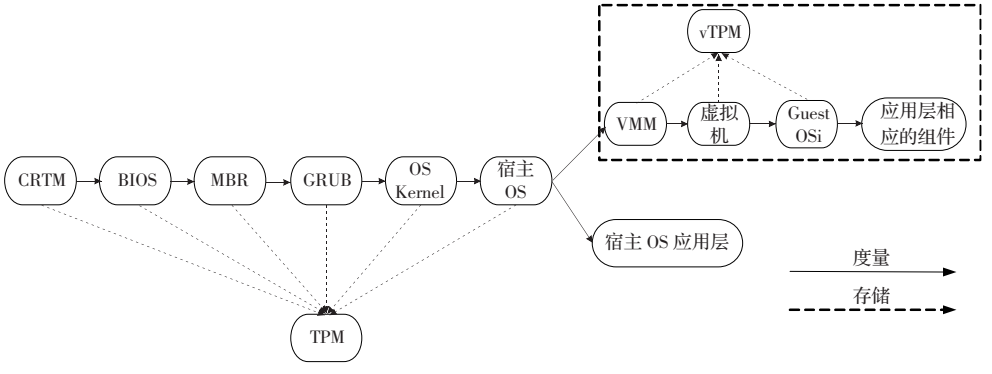

图 2　基于虚拟化技术的并行分类信任链模型

虚拟机监视器驱动程序及相关组件、虚拟机监视器、虚拟机进行检查、Guest OS 等可执行文件,动态共享库,配置文件,解释型对象，完整性度量模块必须对上述所有的组件都进行度量。LSM 是一个轻量级的通用的访问控制架构。在内核很多地方设置了钩子函数，在打开文件装载程序等操作之前需要检验权限。根据 IMA 可以创建基于 LSM 的模块 keeper 利用 LSM 的这些钩子函数来验证内核模块、可执行文件、解释型对象的完整性[6]。度量方案如图 4 所示。

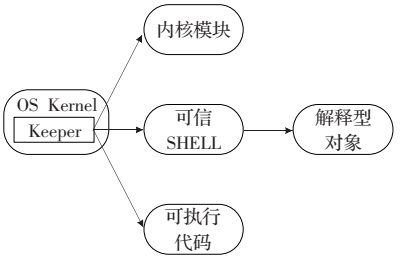

图 4 应用层度量方案

该方案需要维护五张表:可信模块列表——内核模块路径及模块及签名散列值的二元组的集合;可执行文件列表——可执行文件路径及其签名散列值，配置文件路径和动态共享库名；可信配置文件列表——可信配置文件路径及其签名散列值；动态共享库列表——动态共享库路径及其签名散列值;解释型对象列表——解释型对象路径及其签名散列值。这些表需要在一个被隔离的可信系统上获得,以确保签名散列值所对应的组件的确是可信的,当系统启动的时候这些列表会被装载到相应的操作系统中[4]。

①内核模块随时可以被插入或卸载,这样可信状态随时可能被破坏,因此需要在插入或卸载内核模块时验证该操作是否会影响内核可信状态。当试图装载一个内核模块时 Keeper 模块事先注册好的 LSM 钩子函数会被调用，对该内核模块取散列值并与从可信列表中计算到的散列值进行比较,如果匹配就允许加载;如果不匹配,则根据预定义策略或拒绝加载;

②可执行文件是由相应的用户态的装载器来进行装载的。当试图装载可执行文件时,Keeper 模块事先注册好的 LSM 钩子函数会被调用,这个函数先对该可执行文件的装载器进行度量,然后在宿主操作系统中由装载器度量可执行代码;Guest OS1 和 Guest OS2 利用可信可执行文件列表中的动态共享库名与配置文件的路径信息同时对该可执行文件对应的配置文件与动态共享库进行验证;

③解释型对象是由 shell 调用相应解释器来解释执行,在 Guest OS3 中,当 Shell 调用脚本解释器时,先对解释型对象进行验证,然后再调用相应的解释器。

3 形式化验证

3.1 系统引导阶段

文献[9]给出一个组件可信的条件及以下相关定义。

条件 I 组件身份明确——组件的身份可验证。

条件 E 组件的行为可预期——一个组件总是以厂商设计的行为方式执行,除非该组件被非法篡改。即组件的完整性状态为真时,其确定的行为可以期望。

条件 M 组件完整性状态可被真实度量——组件的完整性状态,在任何运行时刻都能被度量和真实反映。

定义 1(可信谓词) Trusted(C),表示一个组件 C 是可信的;条件满足谓词:Meet(B,conditions)表示 B 满足条件 conditions。

在可信平台内,满足条件 I、E、M 的组件为可信组件,即公式(1)。

$$\text{Meet}(C,I)\wedge\text{Meet}(C,E)\wedge\text{Meet}(C,M)\rightarrow\text{Trusted}(C) \quad (1)$$

可信平台启动过程是信任链的建立过程,信任将沿信任链传递。平台启动时第一个运行的组件是信任传递的原点,即 CRTM。一般认为 CRTM 是可信、功能正确而且不需要外界保护的，它可以由专家来评估和确定是否符合可信的标准。则 CRTM 满足可信组件的三个条件 I、E、M[9]。即:

$$\text{Meet}(\text{CRTM},\text{I})\wedge\text{Meet}(\text{CRTM},\text{E})\wedge\text{Meet}(\text{CRTM},\text{M})\rightarrow \text{Trusted}(\text{CRTM}) \quad (2)$$

文献[10]给出了以下定理:

定理 1 一个系统 M 当满足下面 3 个条件时:

(1)M 从可信根开始运行;

(2)M 中的进程满足单步隔离性和输出隔离性;

(3)M 满足可信传递性质。

则 M 是可信系统。

推论 1 系统引导阶段的设计满足可信需求。

证明:

(1)式由(2)可知,CRTM 符合可信根的要求;

(2)从终端加电到操作系统装载是一个顺序固定的过程,前面的进程执行结束后才会执行下一个的进程,满足依次执行的单个进程之间满足单步隔离性质和输出隔离性质;

(3)前一个进程验证后一个进程,如果验证结果和预期的值相同,则表明后一个进程可信,也就是满足可信传递性质,此时才会把系统的控制权交给后面的进程;

即该系统引导阶段的设计满足可信的需求。

3.2 应用可信传递

文献[11]中给出应用可信的定义 2:应用可信的概念包括:应用的完整性可信和行为的可信。通过保障应用程序的完整性可信,能够不依赖于入侵型恶意代码的特征,对所有入侵型恶意代码都能有效防范，从而保障整个应用环境的完整性可信;通过保障应用的行为可信可以有效防止解释型恶意代码的传播,将其破坏程度降到最低。参考文献[11]给出以下定义:

定义 3 系统 M 是一个五元组<S,P,D,PE,R>;S:系统状态集合,包含一个初始状态 $s0\in S$。ST 为安全状态集合,且 $s0\in ST$;SF=S-ST 为不安全状态集合;P:系统可运行的应用程序集合;Pd:系统可加载的动态共享库集合;Ps:系统内解释型对象的集合;Pc:系统内配置文件的集合;R:完整性状态,包括{True,False}。约定使用 s,t,…代表系统状态集合中的元素,也即表示单个系统状态;使用小写 p,q,…代表系统运行的应用程序集合的元素,也即单个应用程序;使用 p_d,q_d 代表应用程序 p,q 需要加载的动态共享库的集合;p_s,q_s 分别代表应用程序 p,q 对应的解释型对象的集合;p_c,q_c 分别代表应用程序 p,q 使用的配置文件的集合;

定义 4 某一应用程序 p 是完整性可信的，当其满足如下

条件:

$int(p,p_d,p_c,p_s)$=True

定义 5 某一应用程序 p 可执行,当且仅当其完整性是可信的。

定义 6 若一应用程序 p 是完整性可信的,则系统 M 的状态 $s \in S_T$ 执行 p 后的状态也是完整性可信。

文献[11-12]给出以下定义:

定义 7 对系统允许运行的应用程序进行控制,控制所有应用程序的执行脚本、宏等,防止解释型恶意代码的扩散,可以保证应用的行为可信。

定义 8 对解释程序的输入进行检测,可以防止解释型病毒被触发。

定理 2 若系统 M 中执行的所有应用程序都完整性可信,则系统 M 是完整性可信的。

推论 2 若系统 M 完整性可信,则任何新允许运行的应用/服务都是完整性可信,即能够保证信任在应用环境中传递。

定理 3 若当前系统 M 完整性可信,必然可以防止入侵型恶意代码的感染和传播。

推论 3 应用可信传递阶段的设计能够保证应用的完整性可信和行为可信。

证明:

(1)在应用层可信传递阶段,首先对所有待运行应用程序的完整性进行检查,只有完整性校验值一致的应用程序才允许运行;其次对应用程序加载的动态库,配置文件,可执行脚本进行检查,判断其是否是系统允许加载的动态库,并对其完整性进行检查,符合条件的才允许加载或使用,否则不允许加载或使用;同时还通过完整性度量保证了虚拟机监视器驱动程序及相关组件、虚拟机监视器、虚拟机进行检查、Guest OS 等的完整性。根据定义 4,定理 2,推论 2 系统能够保证应用的完整可信;

(2)系统保障了应用程序的完整性可信,根据定理 3,系统能够防止入侵型恶意代码的感染和传播;

(3)在应用层可信传递阶段,对解释型对象的完整性进行检查,如果其完整性没有被破坏,则执行否则拒绝,根据定义 7,8 可知,系统能够防止解释型病毒被触发,保证应用行为可信;

由定义 2 可知,应用可信传递阶段的设计能够保证应用的完整性可信和行为可信。

4 结束语

提出的信任链传递模型能够保证终端从可信源头开始至系统启动整个过程的安全可信性,且模型利用虚拟化技术提出了并行分类完整性度量方法,降低了应用层可信传递过程中的完整性度量时间开销。未来将在现有研究基础上,继续完善信任链传递过程中的完整性度量机制,详细实现并行分类完整性度量方案。

参考文献:

[1] TCG.TCG Specification Architecture Overview [EB/OL].(2005-02). https://www.Trustedcomputinggroup.org.

[2] 张焕国,罗捷,金刚.可信计算研究进展[J].武汉大学学报:理学版,2006,52(5):513-518.

[3] 沈昌祥,张焕国,冯登国.信息安全综述[J].中国科学:E,37(2):129-150.

[4] 叶波,陈克非.可信 Linux 关键组件验证方案的研究[J].计算机工程,2006,32(22).

[5] Lambrou D.TCPA enabled open source platforms.

[6] Sailer R,Zhang Xiaolan,Jaeger T,et al.IBM Reaseach Report,Design and Implementation of a TCG-based Integrity Measurement Architecture[C]//13th Usenix Security Symposium,San Diego,California,August 2004.

[7] TCG.TCG Infrastructure Working Group Architecture Part II Integrity Management Specification Version 1.0[EB/OL].(2006).http://www.trustedcomputing.org.

[8] 谭良,周明天.一种并行可复原可信启动过程的设计与实现[J].计算机科学,2007,34(10).

[9] 李莉,曾国荪,陈波.基于时态逻辑的可信平台信任链建模[J].计算机学报,2008,35(4).

[10] 赵佳,沈昌祥,刘吉强,等.基于无干扰理论的可信链模型[J].计算机研究与发展,2008,45(6).

[11] 王飞,刘威鹏,沈昌祥.应用可信传递模型研究[J].计算机工程与应用,2007,43(29):1-3.

[12] 陈泽茂,沈昌祥.解释型病毒及其防御策略研究[J].计算机工程与应用,2004,40(19):29-30.

(上接 97 页)

[5] Verma D.Simplifying network administration using policy based management[J].IEEE Network Magazine,2002:20-26.

[6] Basile C,Lioy A.Towards an algebraic approach to solve policy conflicts[C]//FCS'04,Turku,2004:319-338.

[7] Dulay N,Lupu E,Sloman M,et al.A policy deployment model for the ponder language[C]//IEEE/IFIP International Symposium on Integrated Network Management,London,2001:529-543.

[8] van Lamsweerde A.Goal-oriented requirements Analysis with KAOS [C]//The First International Workshop on Policies for Distributed Systems and Networks (POLICY 1999),Bristol,United Kingdom,1999:31-37.

[9] Bandara A K,Lupu E C,Moffett J,et al.A goal-based approach to policy refinement[C]//The Fifth IEEE International Workshop on Policies for Distributed Systems and Networks,London,2004:229-239.

[10] Fontaine P J.Goal-oriented elaboration of security requirements[D]. Université Catholique de Louvain,Belgium,2001.

[11] He Qingfeng.Requirement-based access control analysis and policy specification[D].Department of Computer Science,North Carolina State University,2005.

[12] 张翼.信息和系统安全管理策略工程研究[D].上海:交通大学,2006.

[13] 王永亮.网络安全设备策略冲突检测与消解技术研究[D].郑州:解放军信息工程大学,2008.

[14] Wies R.Using a classification of management policies for policy specification and policy transformation[C]//Proc of the IEIP/IEEE International Symposium on Integrated Network Management,Santa Barbara,California,USA,1995:1-14.

[15] Sethi R.程序设计语言概念和结构[M].裘宗燕,译.2 版.北京:机械工业出版社,2002:25-35.