

一种基于 Xen 的 TPM 访问控制改进方法

季 涛, 李永忠

(江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003)

摘 要: 针对可信平台模块(TPM)访问受重放攻击和替换攻击威胁的问题,提出一种改进的 TPM 访问控制方法. 首先建立 TPM 长期访问控制功能,通过创建额外授权数据以保证进程结束后继续授权会话,同时将 TPM 地址与 Domain U 的身份标识号相关联,以保护其地址免受替换攻击. 其次,建立 TPM 所有权共享功能,允许多个 Domain U 使用相同的 TPM 地址并防止死锁,因共享地址不能被重写,可保护敏感数据免受攻击. 最后基于 Xen 实现了该方法并评估了其性能. 实验结果证明了该方法的可行性和有效性,TPM 访问性能开销在可接受的范围内.

关键词: 可信平台模块;访问控制;授权;虚拟机;Xen

中图分类号: TP393.08

文献标识码: A

文章编号: 1000-7180(2012)11-0152-05

An Improved Method of TPM Access Control Based on Xen

Ji Tao, Li Yong-zhong

(College of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: Aimed at solving the problem that TPM access was threatened by attacks such as replay attack and substitution attack, an improved method of TPM access control was proposed. First, TPM long term access control function was established. Extra authorization data was created to continue authorization session after finishing the process. Meanwhile, TPM address was related with an ID of Domain U to protect TPM address from substitution attack. Then, TPM shared ownership function was established, allowing plural Domain Us to use the same TPM address and preventing from the deadlock. Sensitive data was protected against attack because shared address was not rewritable. Finally, this method was implemented based on Xen and its performance was evaluated. Experiments results verified the feasibility and the effectiveness of this method and the overhead of TPM access was within the acceptable range.

Key words: trusted platform module; access control; authorization; virtual machine; Xen

1 引言

虚拟化技术能在一台物理机上提供多元化服务,提高计算资源的使用率. 可信平台模块(TPM)采用了加密方法和杂凑算法,可保护内核数据结构及其计算不被主机系统或系统管理员所破坏^[1]. 此外,访问控制功能可保证从 TPM 外部很难查询到非易失性内存中的数据. 目前 TPM 非易失性内存存在两个问题. 首先,当进程正在处理时,其他进程

不能在相同的非易失性内存中查询数据(密钥数据). 其次,攻击者在进程对 TPM 的使用结束后可访问非易失性内存中的数据. 文献[2]使用了加密芯片组从 TPM 内存中专门读取加密数据. 当虚拟机迁移到其他主机操作系统时,使用 TPM 能确保系统的安全性和完整性. 文献[3]提出了一种共享内存的方法以及调度功能. 文献[4]则实现了虚拟机间的通信并有效执行了内部计算. 本文旨在解决怎样有效使用 TPM 以及在此类虚拟机中如何能有效共享

收稿日期: 2012-05-08; 修回日期: 2012-06-18

基金项目: 江苏省高校自然科学基金项目(05KJD52006);江苏省研究生培养创新工程(CXZZ12_0722);江苏科技大学科研项目(2005DX006J)

相同 TPM 地址的问题. 鉴于此, 提出一种改进的访问控制方法, 通过建立 TPM 长期访问控制功能, 可更安全地保护用户, 同时通过建立 TPM 所有权共享功能, 可在同一时间访问 TPM. 此外, 基于开源 Xen 实现了这些功能并对其做了性能评估.

2 基于 Xen 的 TPM 访问控制方法

2.1 Xen

Xen 是一种由剑桥大学计算机实验室开发的运行在 Unix/Linux 环境下并同时使用半虚拟化和完全虚拟化的开源虚拟机软件^[5]. 该虚拟机具有进程空间、用户账户以及网络, 并且这些都独立于主机操作系统. 在 Xen 中, 主机操作系统叫做 Domain 0, 虚拟机叫做 Domain U. Domain 0 在硬件使用方面具有特权. 采用半虚拟化时, Domain U 指的是 Domain 0 的设备驱动. 此外, Domain 0 也叫做虚拟机监控器, 它提供为 Domain U 模拟设备驱动的功能.

2.2 可信平台模块

可信平台模块(TPM)是一个与主板相连的含有密码运算部件和存储部件的小型片上系统, 可用来管理密钥、证明平台配置, 还可用密码技术将机密数据与特定系统配置绑定. TPM 具有时间戳、对称密钥加密芯片组、非易失性内存、非对称密钥加密芯片组、随机数生成器等组件.

TPM 非易失性内存空间存储的密钥和敏感数据及进程相关, 其他进程不能同时访问 TPM 非易失性内存同样的地址. 这样, 使用 TPM 的虚拟机就能独立运行在主机操作系统上. 在进程利用 TPM 完成一些操作后, 密钥数据和哈希数据这样的敏感数据就驻留在 TPM 非易失性内存地址中. 在此情况下, 其他进程就能读取并重写数据. 此外, 可能会发生死锁, 这是由于一些虚拟机不能同时访问 TPM 非易失性内存中相同的地址造成的.

2.3 虚拟可信平台模块

Xen 提供了虚拟可信平台模块(vTPM)软件将 TPM 指令与虚拟机相关联. 每个虚拟机(Domain U)将 vTPM 作为 TPM 的设备驱动来进行使用. 在 Domain 0 中, vTPM 执行以下两个守护进程:

TPM 管理: TPM 管理是一个访问控制守护进程, 它能允许虚拟机专有地使用 TPM 并将虚拟 TPM 地址映射到实际 TPM 地址上去.

TPM 迁移: TPM 迁移是一个迁移守护进程, 它可将 TPM 中的数据迁移到其他主机操作系统中.

2.4 TPM 访问授权

在对 TPM 的连接中没有泄露授权数据的前提下, TPM 提供两种协议来授权对 TPM 实体的访问和使用, 这两种协议分别是独立对象授权协议(Object-Independent Authorization Protocol, OIAP)和特定对象授权协议(Object-Specific Authorization Protocol, OSAP)^[6]. 他们能开启访问 TPM 功能的授权会话. OIAP 协议有不同的会话而 OSAP 协议可利用同一个会话进行多次授权认证.

OIAP: OIAP 是将 TPM 地址与 TPM 进程相关联的一种协议. OIAP 由以下步骤实现:

(1) 用户开始进程开启授权会话.

(2) TPM 将 TPM 地址与 P_1 相关联, 返回 Even 随机数.

(3) P_1 创建 odd 随机数并计算授权数据(Auth), Auth 是由 Even 随机数和 odd 随机数组成的的一种基于哈希密钥的数据. P_1 将 Auth 和 odd 随机数发送给 P_2 .

(4) P_2 用同样的方法计算 Auth. 如果 P_2 的 Auth 等于 P_1 的 Auth, 那么 P_2 允许 P_1 使用选定的 TPM 指令.

OSAP: OSAP 是一种用于保护相对重要数据的协议. OSAP 在基于哈希密钥的加密功能里增加了两个额外的数. 这两个数在上述步骤(2)和步骤(3)中被创建. 随后 OSAP 使用这两个额外的数来进行 TPM 指令的调用. OIAP 和 OSAP 协议的工作过程类似, 通过调用 TPM 指令来更新数. 然而在进程对 TPM 的使用结束后, TPM 会丢弃授权数据.

2.5 TPM 访问存在的问题

由于管理员能获取授权会话的数, TPM 访问易受重放攻击和替换攻击^[3]的威胁. 虚拟机会交换很多授权消息, 假如主机操作系统采用了虚拟机技术, 便可注意到其他建立授权会话的虚拟机. 此外, 主机操作系统能重写 TPM 的易失性内存. 鉴于此, 攻击者就能用其私钥假扮主机操作系统或虚拟机来进行攻击. 因此, 提出一种改进的访问控制功能. 该功能可保证虚拟机仅仅只用 TPM 地址而其他虚拟机不能使用同样的地址.

另一方面, 在处理 TPM 地址时, 同一时间内任何进程都不能访问该地址. 在此情况下, 可能会发生死锁. 这样会影响到一些需要进行实时执行的服务. 否则, TPM 地址空间就会被浪费于大量密钥的副本. 本文, 虚拟机通过主机操作系统可在同一时间内

使用 TPM 地址.

3 改进的 TPM 访问控制方法

3.1 TPM 长期访问控制功能

由于进程结束对 TPM 的使用后,攻击者能访问 TPM 中的敏感数据,所以 TPM 受中间人攻击和旁路攻击的威胁.因此,有必要对改进的访问控制功能进行实现以保证 TPM 进程完成后继续授权会话.符号说明如表 1 所示.

表 1 符号说明

符号	说明
DomID_i	Domain U_i 的 ID
K_{pub}	Domain 0 的公钥
K_{pri}	Domain 0 的私钥
r_i	由 Domain U_i 创建的随机数
R_i	由 Domain U_i 创建的随机数(使用到签名)
TPM_{mgr}	TPM 管理守护进程
ACCESS	访问控制标志(0:不使用,1:使用)
SHARE	TPM 所有权共享标示(0:不使用,1:使用)
USING	重用标志
INIT	初始化标示
$\text{NONVLO}_{\text{addr}}^i$	Domain U_i 的非易失性内存地址
COUNTER	采用 TPM 所有权共享功能的用户数
OWNER	使用 TPM 所有权共享功能的 Domain U

TPM 长期访问控制功能创建了一个额外的授权数据 Secret 来继续授权. Secret 通过 TPM 地址计算得出,随机数和 Domain ID 都不能被任何 Domain U 调用.

该功能由以下步骤来进行工作:

(1) Domain U_i 创建初始化信息 I,将 TPM 地址 $\text{NONVLO}_{\text{addr}}^i$ 以及标示设置为“INIT=1, USING=1, ACCESS=1”,随后 Domain U_i 将 I 发送给 Domain 0 中的 TPM_{mgr} .

(2) Domain 0 中的 TPM_{mgr} 创建额外的授权数据 Secret, Secret 由随机数 r_i 、TPM 地址 $\text{NONVLO}_{\text{addr}}^i$ 、 DomID_i 计算得出,并且 TPM_{mgr} 将 Secret 发送回 Domain U_i (若 $\text{NONVLO}_{\text{addr}}^i$ 为空的话).

(3) Domain U_i 发送信息 M,将 Secret 以及标示设置为“INIT=0, USING=1, ACCESS=1”.

(4) Domain 0 中的 TPM_{mgr} 将 DomID_i 与 NONV-

$\text{LO}_{\text{addr}}^i$ 相关联.

由于任何 Domain U 都不能获得其 Domain ID, Domain 0 便能保护敏感数据,防止通过 Domain U 对其进行恶意访问.因此,即使攻击者获取了其他 Domain U 与 TPM 管理守护进程之间的授权会话,攻击者也不能从 Domain U 访问 TPM 地址.此外,如果 Domain 0 实现了完全虚拟化的话,攻击者便不能访问 Domain 0.在半虚拟化中,如果 Domain 0 停止了 telnet、ssh、http 等服务的话,Domain 0 仍能保护 TPM 地址.此外,Domain 0 可迁移授权数据 Secret 到其他 Domain 上.

3.2 TPM 所有权共享功能

TPM 实现了两种授权协议来保护 TPM 指令.然而,当有进程在访问 TPM 地址时,任何其他 TPM 进程都不能访问该相同的 TPM 地址.在此情况下会发生死锁.鉴于此,TPM 需要将敏感数据拷贝到其他地址上去,但此时 TPM 的非易失性内存被限制了.

TPM 所有权共享功能通过 Domain 0 来允许 TPM 进程访问相同的地址.然而,共享地址不能被重写. TPM 所有权共享功能可进行以下工作. REG 是一种使用 TPM 地址或首次使用 TPM 地址的进程. REF 则是一种使用相同的 TPM 地址的进程.即便如此,如果 REG 被终止或完成后,REF 也不能保留 TPM 会话.鉴于此,REF 就成为了 REG 以保护 TPM 会话.

REG 工作步骤如下:

(1) Domain U_i 创建信息,将标示设置为“SHARE=1, INIT=1”,并加密数据 E, $E = r_i * \text{Data}_i$, Data_i 包括访问信息.随后 Domain U_i 开始访问 TPM 地址 $\text{NONVLO}_{\text{addr}}^i$.

(2) 若 $\text{NONVLO}_{\text{addr}}^i$ 为空时, Domain 0 中的 TPM_{mgr} 对 E 签名,并将签名数据 Sign 发送给 Domain U_i .

(3) Domain U_i 根据 R_i 和 Sign 计算 Sync,随后 Domain U_i 发送信息开始 TPM 所有权共享会话,将标志设置为“SHARE=1, INIT=0”.

(4) Domain 0 中的 TPM_{mgr} 访问 TPM 并用 TPM 与 Domain U_i 进行通信.随后, TPM_{mgr} 不允许 Domain U 对 TPM 地址进行重写.此外, TPM_{mgr} 将 DomID_i 设置为 OWNER 并增加 COUNTER.

REF 工作步骤如下:

(1) Domain U_j 和(1)进行类似的工作。

(2) TPM_{mgr} 从 Domain U_i 中拾取 REG,并允许 Domain U_j 与 Domain U_i 进行通信。

(3) Domain U_i 用 D-H 密钥交换方法为 Domain U_j 翻译 Sync。

(4) Domain U_j 解密 Sync,获得验证数据 $Verify = Data_i$,随后,若 Domain U_j 通过 $Data_i$ 验证的话,将 Sync 与 Domain U_j 共享。

(5) 若 TPM_{mgr} 接收到信息可使用共享 TPM 地址的话, TPM_{mgr} 建立 Domain U_j 并将其作为 REG。

如果 Domain U 对其内存空间进行了加密,那么 Domain U 利用 TPM 所有权共享功能可保护通信中的敏感数据,防止攻击者进行攻击。

表 2 执行 RSA 加密和 TPM 访问的实验结果

	执行 TPM 指令		用 TPM 长期访问控制功能执行 TPM 指令			用 TPM 所有权共享功能执行 TPM 指令		
	RSA 加密	TPM 访问	RSA 加密	TPM 访问 (第一次)	TPM 访问 (第二次)	RSA 加密	TPM 访问 (REG)	TPM 访问 (REF)
CPU/%	8.1	0.2	8.2	8.5	7.9	8.2	8.2	9.1
内存/MB	23.8	0.32	24.2	35.2	32.7	24.6	34.3	38.6
执行时间/ms	332	33	343	637	391	396	397	446

4.2 TPM 长期访问控制功能实验结果及分析

该功能只验证了会话,因此,TPM 访问可能会比用通常的 TPM 管理守护进程进行 TPM 访问慢。此外,RSA 加密可能也会慢一些,但从实验结果上看,与之前系统的计算能力相差不大。随后对使用 RSA 加密的时间和 TPM 访问时间进行了性能评估。实验结果如表 2 所示。TPM 访问花费了较多时间,但系统并不会经常更新密钥、开始或结束 TPM 访问。此外,RSA 加密执行时间比之前稍长。

4.3 TPM 所有权共享功能实验结果及分析

REG 需计算签名,而 REF 需验证 Sync 和交换密钥。因此,REF 比 REG 稍慢一些。此外,REG 和 REF 通过 Domain 0 加密 Domain U,这样会使 REG 和 REF 增加各自的执行时间。表 2 显示了 REG 和 REF 使用资源及执行时间的增加情况。值得注意的是,REF 的 TPM 访问时间由于密钥交换有所增加。但 TPM 所有权共享功能并不需要在 TPM 进程结束后访问 TPM 管理守护进程。此外,RSA 加密时间增加了约 20%。然而,完成 TPM 访问的方法和之前 TPM 管理守护进程类似。因此,该功能不能加密大量数据但足够完成对授权数据的验证。

4 实验结果

4.1 实验环境

实验环境为: Intel Pentium 4 2.5GHz 的 CPU、512MB RAM 的内存、基于 Linux-2.6.39 内核的 Debian GUN Linux 5.0.1 操作系统、Xen 版本为 4.0.0、Domain U 内存为 128MB RAM 且 Domain U 操作系统为 Debian GUN Linux 5.0.1。我们使用上述计算机和虚拟机配置评估了 RSA 加密并计算了 TPM 访问时间。TPM 访问时间其实就是从开始到完成对 TPM 地址使用的执行时间。实验过程中内核经过重新编译。当虚拟机用户更新密钥数据、开始对 TPM 的访问后,就会执行 TPM 访问。

5 结束语

本文改进了 TPM 访问控制方法。如果 Domain U 实现了完全虚拟化,这将使得任何人都不能访问 Domain 0,Domain U 也不能注意到其 Domain ID。在半虚拟化中,Domain 0 采用了可信启动,这能将保护敏感数据不受攻击。因为 Domain 0 在 TPM 中有其完整性数据,当 Domain 0 正在对其进行处理时,任何 Domain U 都不能访问该数据。此外,实验结果表明 TPM 访问的时间和性能开销都在可被接受的范围内。TPM 所有权共享功能的执行时间由于 Domain U 而增加。因此,下一步工作会将加密密钥指派给加密存储器以减少执行时间,同时 TPM 长期访问控制功能需尽可能实现时间戳以便重用 TPM 地址。

参考文献:

- [1] Shen Changxiang, Zhang Huanguo, Wang Huaimin, et al. Research on trusted computing and its development[J]. Science China(Information Sciences), 2010, 53(3):405-433.
- [2] Chen Haibo, Chen Jieyun, Mao Wenbo, et al. Daoni-

- ty-Grid security from two levels of virtualization[J]. Information Security Technical Report. 2007,12(3): 123-138.
- [3] Carl A W. Memory resource management in VMware ESX server[C]//Proceedings of the 5th symposium on Operating systems design and implementation. New York: ACM Press, 2002:181-194.
- [4] Zhang Xiaolan, Suzanne M, Pankaj R, et al. Xen-Socket: A high-throughput interdomain transport for VMs[C]//Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware. New York: Springer-Verlag, 2007:184-203.
- [5] Todd D, Zachary S, Jeanna N M, et al. Quantitative comparison of Xen and KVM[C]//In Xen Summit. Boston: USENIX Association, 2008: 1-2.
- [6] 顾韵华, 李丹, 陈玮. 基于认证可信度的角色访问控制模型[J]. 微电子学与计算机, 2009,26(7):5-8.
- [7] 秦宇, 兰海波. TPM 虚拟域安全模型[J]. 中国科学院研究生院学报, 2011,28(5):648-658.
- [8] England P, Loeser J. Para-virtualized TPM sharing [C]//Proceedings of the 1st International Conference on Trusted Computing and Trust in Information Technologies: Trusted Computing-Challenges and Applications. Berlin: Springer-Verla, 2009: 119-132.
- [9] Matthias B, Michael S, Georg B, et al. Non-intrusive virtualization management using libvirt[C]//Proceedings of the Conference on Design, Automation and Test in Europe. Belgium: ACM Press, 2010: 574-579.
- [10] Shane B, Eimear G, Chris J M, et al. Challenges for trusted computing[J]. IEEE Security and Privacy, 2008,6(6):60-66.

作者简介:

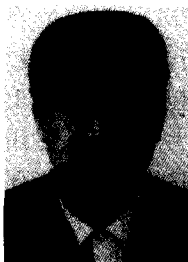
季涛男,(1985—),硕士研究生.研究方向为网络与信息安全.

(上接第 151 页)

- [2] Gu P, Sosale modularization for life cycle engineering [J]. Robotics and Computer Integrated Manufacturing 1999,15(5):387-401.
- [3] Hiroshi Sako, Takafumi Miyatake. Image recognition technologies towards advanced automated teller machines[C]//Proceedings of the 17th International Conference on Pattern Recognition. 2004:1051-1054.
- [4] Rafael C Gonzalez, Richard E Woods. Digital Image Processing[M]. 北京:电子工业出版社,2007:118-127.
- [5] 张凯,金瓯,贺建飏.一种基于 μ Clinux 的嵌入式纸币识别系统设计[J]. 计算机测量与控制. 2008(04): 552-555.
- [6] 徐琳琳,苑玮琦.基于线阵图像传感器点钞机纸币号码识别系统[J]. 计算机工程与设计. 2008(19):4954-4956.
- [7] 崔天横,高晶敏,刘振慧.经典边缘检测的快速算法[J]. 微计算机应用. 2007(06):579-675.
- [8] 郭艳平,侯凤贞.纸币面值识别系统图像分割技术的算

法[J]. 重庆工学院学报:自然科学版. 2008(11):124-126.

作者简介:



赵喜清男,(1970—),副教授,硕士.研究方向为无线网络,空间数据库.



张利明男,(1974—),讲师,硕士.研究方向为图像处理,空间数据库