针对权限滥用的安全增强研究

姚立红, 訾小超, 茅 兵, 谢 立
(南京大学计算机软件新技术国家重点实验室, 计算机科学与技术系, 江苏南京 210093)

摘 要: 要限制权限滥用, 操作系统必须清楚哪些是应用程序正常所需权限. 本文分析了应用程序所需权限的决定因素, 提出以应用程序相关的用户知识信息为基础, 结合系统结构、程序实现和系统安全等方面的专业知识推导程序所需权限的权限控制方案. 该方案特点在于兼顾到控制准确性和易用性, 测试和试用表明其实用性较好.

关键词: 信息安全; 权限滥用; 不可信程序
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2003) 11-1747-03

Research on Security Enhancement to Privilege Abuse

YAO Li-hong, ZI Xiao-chao, MAO Bing, XIE Li
(State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu 210093, China)

Abstract: To prohibit privilege abuse, operating system must clearly know the necessary privileges needed by a program. The paper researched the key factors about the privileges needed by special applications, and brought forward a security framework. Based on user. s knowledge about an application, this framework deduced the privileges the application needed by using professional knowledge about system architecture etc., thus restricted the privilege abuse. Our framework balances accuracy and facility about privilege abuse, and is practical.

Key words: information security; privilege abuse; untrusted application

1 引言

目前流行的操作系统(Windows, Linux 等)主要依据进程的启动用户判断进程的操作权限, 这意味着进程可以享有启动用户的所有权限, 给应用程序篡改用户意图滥用权限提供了条件. 近年来应用程序滥用权限的问题逐渐引起人们的注意, 相关研究日益受到重视. 从本质上看, 操作系统对应用程序滥用权限实施控制并不困难, 但操作系统必须知道程序正常运行必需的权限, 才能判定出应用程序是否在滥用权限, 如何获得程序正常所需的权限成为控制不可信程序滥用权限的关键. 对此, 很多学者进行了大量的研究, 提出了多种解决方案, 根据所依赖信息的来源不同可大致分为:

(1) 用户直接提供操作系统层次上的程序权限需求[1,2]. 这种方案的特点在于容易实现, 权限控制比较准确(若权限配置正确), 但要求用户详细给出目标程序所需的系统级权限, 超出了大多数用户的能力范围, 难以被普遍接受;

(2) 通过预运行获取程序所需权限信息[3,4]. 通过多次预运行目标程序, 监测其相应的权限需求, 把这些需求作为目标程序正常的权限需求. 这种预运行方案对用户来讲使用比较麻烦, 也很难保证预运行的可靠性(即是否也存在权限滥用), 因而该方案并没有被普遍接受, 很少得到实际应用;

(3) 应用软件商提供权限需求信息[5]. 该方案要求软件厂商在发布应用软件时, 附带该应用程序所需权限的脚本, 操作系统据此进行权限控制. 这种思想在 Java 虚拟机对移动代码进行权限控制时得到应用[6], 但用作操作系统的权限控制并不理想, 因为很多程序的权限需求并不是由软件者唯一确定, 如编辑器需要打开哪些文件进行编辑由用户输入确定;

(4) 各种模糊控制. 依据相关要素进行一定的逻辑推理, 计算出特定权限是应用程序正常所需权限的概率及可能带来的安全威胁等, 从而进行权限控制, 如基于操作危险级别的权限控制方案[7]. 模糊控制使用比较方便, 很少需要用户干预, 其缺点在于权限控制的正确性难以得到保证.

总的来讲, 方案(1)、(2)主要依赖用户的信息和能力, 易用性问题突出; 方案(3)、(4)主要依赖相关专业信息, 难以实现权限控制的准确性. 鉴于此, 本文引入用户知识和相关专业知识相结合来获取程序所需权限的技术方案, 提出基于知识结合的权限验证和控制框架, 并把该框架运用于我们主持开发的安全增强系统.

2 系统级权限获取原则与方法

2.1 用户知识基础作用

理想情况下, 程序使用的权限是否用于完成用户功能需求可被看作判断程序是否滥用权限的基本依据, 与实现用户需求无关的权限需求从理论上都可被判定为权限滥用. 鉴于用户功能需求对确定程序所需权限的基础作用, 控制权限滥用应充分利用用户知识.

2.2 专业知识的辅助作用

尽管用户知识非常重要且可靠, 但要直接以此进行权限

收稿日期: 2002-08-29; 修回日期: 2003-07-10
基金项目: 国家 863 高科技基金(No. 2001AA144010)

控制并不现实: 操作系统需要系统级的权限控制信息, 用户知识大多表现为用户层次上的功能需求信息. 以专业研究人员拥有的知识为基础, 构造相应的自动化推理系统能有效地实现用户需求到系统级权限需求的转化, 以用于操作系统的权限控制.

213 基于关键字的知识结合

鉴于系统易用性考虑, 不可能要求用户以复杂的形式化语言详尽地描述自己的知识, 应该尽可能实现交互接口的简单化和交互内容的最少化. 为此, 本文引入了以关键词选择为主的用户知识获取接口, 具体过程包含三步骤: (1) 根据专业知识确定出供用户选择的关键词集合; (2) 用户在关键词集合中选择出与应用程序相关的关键词; (3) 推理系统根据专业知识和用户选定的关键词确定出该应用程序所需的权限范围.

3 基于知识推导的系统安全框架

基于上面的分析, 本文提出以知识结合为基础的系统安全框架 KBSF (Knowledge. Based Security Framework), 用于限制应用程序滥用权限, 如图 1 所示, 该框架包括:

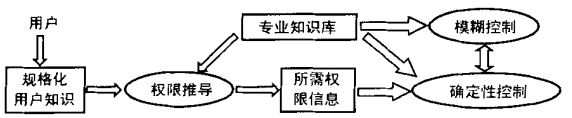

图 1 基于知识结合的系统安全框架示意图

311 用户知识的规格化描述

根据以往的相关研究工作[7, 8] 和 2 节中的讨论, 本文提出如下的用户知识需求及其规格化描述:

(1) 关键词列表: 由用户从候选关键词集合中选出与目标程序相关的多个关键词, KBSF 主要包括三类候选关键词:

- 程序类别关键词: 指明程序的类别, 如系统管理工具类和网络客户端类等;
- 管理功能关键词: 目标程序在系统管理方面的功能, 如用户管理等;
- 特殊功能关键词: 有些系统特殊权限 (如 Ptrace 调用) 只在完成特殊功能 (如程序跟踪和调试) 时才需要使用;

(2) 资源需求列表: 给出用户所知的与目标程序有关的资源使用信息, 具体包括两方面:

- 用户数据资源: 用户运行目标程序所涉及到的用户数据文件和目录列表, 如要编辑的文件等;
- 程序自带资源: 包含可执行文件和配置资源 (如翻译工具带的字典文件) 等.

(3) 程序宏观描述: 用于表示用户提供的目标程序的其他相关信息, 希望从中推导出一定的信息, 以实现更加合理的权限控制. 具体包括:

- 程序可信度: 表示应用程序的可靠程度 (Trusted, Self2Trusted, Untrusted), 用户可以通过该程序的来源大致确定;
- 程序重要性: 表示该程序对用户的重要程度 (Important, Unimportant).

KBSF 框架以用户知识文件保存针对特定程序 (或功能相近的多个程序) 的用户知识, 它在运行该目标程序前通过与用户交互生成, 形式如 (/usr/sbin/usermanage. des):

```
ExecuteFile: /usr/sbin/adduser, /usr/sbin/deluser, /usr/sbin
        /usermod;
Key Word: UserManage;
MacroParmeter: Trusted; Important;
```

312 专业知识与权限推导

从 311 用户给出的用户知识可看出, 资源需求信息比较具体, 可以直接用于实施权限控制, 而关键词对应的权限需要结合专业知识才能确定. KBSF 框架的专业知识库通过如下形式表示关键词对应的操作权限:

```
UserManage:
  Read, Write: /etc/passwd, /etc/group, /etc/shadow;
  CreatFile: /home/*;
  Read: /etc/login. defs, /etc/skel/* #*, /etc/default/useradd;
```

权限推导模块的主要功能就是根据专业知识库的内容, 把规格化的用户知识转化为相应的权限控制脚本: 特殊系统调用列表和所涉及到的文件及设备等资源列表. 此外程序宏观描述也复制到权限控制脚本中, 形式如 (/usr/sbin/usermanage. spe):

```
ExecuteFile: /usr/sbin/adduser, /usr/sbin/deluser, /usr/sbin
        /usermod;
SpecialSyscall:
Resource:
  Read, Write: /etc/passwd, /etc/group, /etc/shadow;
  CreatFile: /home/*;
  Read: /etc/login. defs, /etc/skel/*. *, /etc/default/useradd;
  MacroParmeter: Trusted; Important;
```

313 权限验证与控制

根据目标程序的权限控制脚本对该进程提出的权限请求进行验证, 根据验证的结果进行权限控制: 许可目标进程的操作请求, 或者调用模糊控制模块进一步处理.

314 模糊权限控制

为尽可能减少对程序正常运行的影响, KBSF 框架对不能通过验证的权限请求并不立即拒绝, 而是依据专业知识综合相关要素进行模糊控制, 主要考虑要素包括:

权限近似性: 即该权限是否与权限脚本中的某权限存在近似关系;

权限滥用危害性: 即如果目标程序滥用该权限, 对系统安全造成危害的严重程度;

程序可信度: 从用户给出的宏观描述中获得, 可信度越高则权限请求越容易通过;

程序重要性: 从用户给出的宏观描述中获得, 不重要目标程序的权限请求容易被拒绝.

鉴于篇幅, 本文不再给出具体的模糊控制算法.

4 系统实现和测试

411 系统实现

本文提出的 KBSF 框架已在南京大学主持开发的基于 Linux 的安全增强操作系统中得到实现, 如图 2 所示:

- 用户知识获取模块: 通过图形化交互界面从用户处获取目标程序的规格化描述;
- 权限推导模块: 利用专业知识库的相关内容, 把用户给

出的程序规格化描述转化为具体的权限信息, 保存在相应的权限脚本中;

u权限验证与控制模块: 根据相应的权限脚本对目标程序提出的请求进行验证.

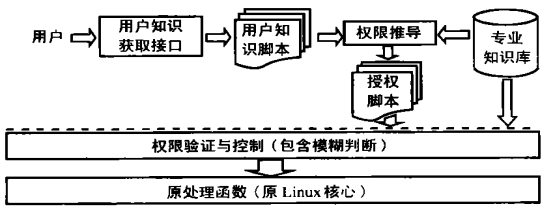

图 2　KBSF 的实现结构简图

4.2　权限控制效果测试

针对错误控制率的测试, 由非专业人员分别对 Linux 系统中常见的应用程序进行知识配置和实际运行, 共出现了三次因权限控制而影响了应用程序的正常运行, 经适当调整后, 消除了这三处错误控制. 针对漏控率的测试, 本文选择 Wuftp 和 Sendmail 等应用程序的漏洞版本以及恶意作为测试对象, KBSF 框架发现并控制了所有严重危害系统安全的权限滥用行为.

4.3　系统性能测试与分析

表 1 列出了原 Linux 和通过 KBSF 框架安全增强后的系统(下文称 KBSF_Linux)在主要系统操作的平均时间开销. 系统测试以 Intel P Ó 733 作为平台, 基准程序为 LMbench[9].

表 1　性能测试数据

Object System	Create (0K)	Create (10K)	Delete (0K)	Delete (10K)	Open	Fork	Exec
Linux	7510	16812	9144	2115	5180	144	7227
KBSF_Linux	7813 (+ 414%)	17115 (+ 210%)	9185 (+ 413%)	2210 (+ 213%)	6123 (+ 714%)	148 (+ 218%)	8679 (+ 2011%)

从表 1 可看出, 除 Exec 调用外, KBSF_Linux 处理的时间开销与原 Linux 核心相比增加幅度不大, KBSF_Linux 的 Exec 操作比原 Linux 核心有一定增加(达 2011%), 这主要因为 Exec 调用启动新应用程序, 需要把新应用程序对应的权限脚本一次性加载到核心中.

5　相关工作比较

同以往相关研究工作[见引言]相比, KBSF 框架具有如下特点:

u与依靠用户获取权限需求的控制方案相比, KBSF 不再要求用户提供详尽的系统级的权限需求信息, 用户交互接口简单, 易用性好;

u与依靠软件厂商提供权限需求的控制方案相比, 由于用户输入信息等相关知识的引入, KBSF 框架提高了权限控制的准确性;

u用户功能需求是判断权限滥用的重要依据(见 2.1 节), 与单纯利用安全专业知识进行模糊控制相比, 用户知识的引入使得 KBSF 权限控制更加准确和合理.

6　结束语

本文提出了针对权限滥用的安全增强框架, 该框架在我们开发的安全操作系统上得到应用, 大量测试和试用结果表明该框架使用方便, 权限控制比较准确, 且对系统性能影响不大, 具有较高的实用性.

参考文献:

[1] Massimo Bernaschi, et al. Operating System Enchancements to Prevent the Misuse of System Calls [DB/OL]. http://delivery.acm.org/10.1145/360000/352624/p174-bernaschi.pdf.

[2] Ian Goldberg, et al. A Secure Environment for Untrusted Helper Applications [DB/OL]. http://www.usenix.org/publications/library/proceedings/sec96/goldberg.html, 2002-05-01.

[3] G Ko, et al. Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring [DB/OL]. http://seclab.cs.ucdavis.edu/papers/pdfs/ck2gl2kl294.pdf, 2001-10-03.

[4] R Sekar, et al. A Fast Automaton2Based Method for Detecting Anomalous Program Behaviors [DB/OL]. http://www.computer.org/proceedings/sp/1046/10460144abs.htm.

[5] T Jaeger, et al. Building Systems That Flexibly Control Downloaded Executable Content [DB/OL]. http://www.usenix.org/publications/library/proceedings/sec96/jaeger.html, 2002-05-01.

[6] G Necula, P Lee. Research on Proof2Carrying Code for Untrusted2Code Security [DB/OL]. http://www.computer.org/proceedings/s&p/7828/78280204.pdf.

[7] Massimo Bernaschi, et al. Remus: A security2enhanced operating system [J]. ACM Transactions on Information and System Security, 2002, 5(1):36-61.

[8] Anurag Acharya. Mandar Raje. MAPbox: Using Parameterized Behavior Classes to Confine Untrusted Applications [DB/OL]. http://www.usenix.org/publications/library/proceedings/sec2000/acharya.htm, 2002-01-29.

[9] Larry Mcvoy, Carl Staelin. Lmbench: Portable tools for performance analysis [DB/OL]. http://www.usenix.org/publications/library/proceedings/sd96/full_papers/mcvoy.pdf, 2001-04-19.

作者简介:

姚立红　女, 1974 年 12 月生于江苏建湖, 博士生, 研究方向: 信息安全.

訾小超　男, 1974 年 4 月生于河南商水, 博士生, 研究方向: 系统安全.