安胜安全操作系统的隐蔽通道分析*

卿斯汉[1,2,3+], 朱继锋[1,2,3]

[1](中国科学院 信息安全技术工程研究中心,北京 100080)

[2](中国科学院 软件研究所,北京 100080)

[3](中国科学院 研究生院,北京 100039)

Covet Channel Analysis on ANSHENG Secure Operating System

QING Si-Han[1,2,3+], ZHU Ji-Feng[1,2,3]

[1](Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing 100080, China)

[2](Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[3](Graduate School, The Chinese Academy of Sciences, Beijing 100039, China)

+ Corresponding author: Phn: +86-10-62635150, E-mail: qsihan@ercist.iscas.ac.cn, http://www.iscas.ac.cn

Received 2004-05-17; Accepted 2004-07-16

Qing SH, Zhu JF. Covet channel analysis on ANSHENG secure operating system. *Journal of Software*, 2004, 15(9):1385~1392.

http://www.jos.org.cn/1000-9825/15/1385.htm

Abstract: ANSHENG operating system is a secure operating system with high security level based on Linux, which is independently developed with the security kernel, security framework and security models. In this paper, the approaches to analyzing covert channels in the underlying system are summarized, and it is the first time that the covert-channel-analysis results on a secure operating system based on Linux kernel have ever been reported. Some new covert channels have been found by the novel backward tracking approach. For the identified covert channels, accurate bandwidth computation and appropriate covert-channel handling have been performed.

Key words: ANSHENG secure operating system; covert channel analysis; backward tracking approach; storage channel; information flow

摘 要: 安胜安全操作系统是自主研制的基于 Linux 的高安全等级安全操作系统,包括安全内核,安全架构与安全模型.总结了对该系统进行的隐蔽通道分析方法,首次报道基于Linux内核开发的安全操作系统的隐蔽通道分析结果.应用新型的"回溯方法"发现了某些新的隐蔽通道.对被标识的隐蔽通道,准确地计算了它们的带宽,并进行了适当的隐蔽通道处理.

关键词: 安胜安全操作系统;隐蔽通道分析;回溯方法;存储通道;信息流

中图法分类号: TP309 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant No.60083007 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035810 (国家重点基础研究发展规划(973))

作者简介: **卿斯汉**(1939－),男,湖南隆回人,研究员,博士生导师,主要研究领域为信息系统安全理论和技术;**朱继锋**(1971－),男,博士生,主要研究领域为信息系统安全理论和技术.

1 概 述

1.1 隐蔽通道

自从 1973 年 Lampson[1]首先提出隐蔽通道的概念之后,对隐蔽通道的研究逐渐深入.目前,普遍认可 1990 年 Tsai,Gligor 和 Chandersekaran[2]给出的定义"给定一个强制安全策略模型 M 和它在一个操作系统中的解释 $I(M)$,$I(M)$中两个主体 $I(S_i)$和 $I(S_j)$之间的任何潜在通信都是隐蔽的,当且仅当模型 M 中的相应主体 S_i 和 S_j 之间的任何通信在 M 中都是非法的".由此可知,隐蔽通道分析与强制安全策略模型有着密切的关系,与自主安全策略模型无关.

美国国防部颁布的桔皮书 TCSEC[3]规定,在评估 B2 级以上的安全操作系统时,必须分析隐蔽通道.我国国家标准 GB17859-1999"计算机信息系统安全保护等级划分准则"[4],以及其他相关的国际国内标准,都有类似的规定.同时,安全级别越高,对隐蔽通道分析的要求也越严格.

隐蔽通道主要有两种类型:存储通道和定时通道.如果一个进程直接或间接地写一个存储单元,另一个进程直接或间接地读该存储单元,则称这种通道为隐蔽存储通道.如果一个进程通过调节它对系统资源的使用,影响另外一个进程观察到的真实响应时间,实现一个进程向另一个进程传递信息,则称这种通道为隐蔽定时通道.特别地,隐蔽存储通道还可以分为两种类型:如果发送方进程通过选择是否耗尽共享资源作为 0 和 1 的编码发送消息,则称为资源耗尽型隐蔽通道;如果发送方进程并不耗尽共享资源,而是通过修改共享资源的状态发送信息,则称为事件计数型隐蔽通道.另外,在隐蔽通道中,如果对发送进程传送的任意比特,接收进程都能以概率 1 正确地解码,则称该通道为无噪通道.相反地,在噪音通道中,对发送进程传送的任意比特,接收进程能够正确解码的概率小于 1.

1.2 隐蔽通道分析

隐蔽通道共有 3 个分析步骤:(1) 搜索隐蔽通道;(2) 根据搜索的结果,计算/测量隐蔽通道的带宽;(3) 根据计算/测量的结果,处理隐蔽通道.隐蔽通道分析可以在以下 3 个层次进行:(a) 描述性顶层规范 DTLS(detailed top-level specification);(b) 形式化顶层规范 FTLS;(c) 源代码.

通过隐蔽通道标识方法静态分析顶层规范或源代码时所产生的隐蔽通道,称为潜在隐蔽通道.在系统动态执行中,由于某些条件不满足,这些通道可能不会产生.因此,只有能够构造出隐蔽通道真实应用的场景,潜在隐蔽通道才是真实隐蔽通道.

1.3 安胜安全操作系统

安胜安全操作系统(版本 4.0),以下简称安胜 OS v4.0,是我们自主研制的基于 Linux 的高安全等级安全操作系统,包括安全内核、安全架构与安全模型等.共分析了 Linux 内核源程序 38 万行,改造和开发了 4.5 万多行 C 语言程序.对系统原有的 200 多个系统调用,逐个进行了安全性分析.未做任何修改的 49 个,加入审计机制的 130 个,加入存取控制机制的 95 个,新开发系统调用 33 个.新开发用户 shell 命令 69 个,修改用户 shell 命令 16 个.建立了新的支持多策略的安全架构与 3 个独创的安全模型:新的机密性模型、新的完整性模型与新的权限控制模型.对于相当于 B2 级的操作系统,国内外各类信息安全产品评估标准只要求分析隐蔽存储通道.安胜 OS v4.0 的设计目标为 B2 级(相当于国标 GB17859[4]第 4 级),因此本文只涉及隐蔽存储通道的分析.关于安胜 OS v4.0 的安全模型与安全架构更详细的介绍,请参看文献[5].

下面,我们简单介绍安胜 OS v4.0 与隐蔽通道分析有关的部分.安胜 OS v4.0 与隐蔽通道分析有关的安全策略是:

(1) 一个信息流路径是否构成隐蔽通道,取决于该通道是否违背系统强制安全策略.

(2) 对文件/目录/设备等的访问受 DTE,MAC,ACL 和特权的共同控制,而对 IPC 等客体的访问则只受 MAC,ACL 和特权的控制.

(i) ACL 是读、写、执行(搜索)权限位的匹配.

(ii) DTE 权限分为读、写、执行、连接、创建和目录搜索,也是针对不同的域和型来进行匹配.

(iii) MAC 根据客体的不同,分为

(a) 单级属性的客体(文件、目录、IPC 等):等写下读.亦即,如果主体安全级等于客体安全级,则主体可以写客体;如果主体安全级支配客体安全级,则主体可以读客体.

(b) 多级属性的客体(文件系统、设备等):如果主体安全标记支配客体的最小安全级,则主体可以读客体;如果主体安全级在客体安全级范围之内,主体可以写客体.

安胜 OS v4.0 基于 Linux-2.4.18 内核,Linux-2.4.18 比早期的 Linux-2.2.x 等版本有许多重要改进,例如,新版本应用文件锁表链表取代了文件锁表数组,因而消除了耗尽文件锁列表可能导致的隐蔽通道.

2 安胜安全操作系统的隐蔽通道标识方法

2.1 已有的隐蔽通道标识方法总结

在隐蔽通道分析中,最困难的是隐蔽通道的标识.在理论上,缺乏有效的方法;在工程上,工作量大,缺乏有效的自动工具.主流的隐蔽通道标识方法共有 4 种:(1) 语法信息流法;(2) 无干扰法;(3) 共享资源矩阵法;(4) 语义信息流法.目前,应用最为广泛的是方法(3)和方法(4).下面,我们对它们进行简短的回顾.

2.1.1 共享资源矩阵法

Kemmerer[6]提出的共享资源矩阵法,又称为 SRM(shared resource matrix)方法,是目前应用较多的一种隐蔽通道标识方法.该方法的分析步骤是:

1) 分析全部 TCB 原语操作,确定用户通过 TCB 接口可见/可修改的共享资源属性;

2) 构造共享资源矩阵,该矩阵的行对应 TCB 原语,列对应可见/可修改的共享资源属性.如果一个原语可以读一个变量,则将该矩阵项标记为 R.如果一个原语可以修改一个变量,则将该矩阵项标记为 M;

3) 对共享资源矩阵完成传递闭包操作[6];

4) 分析每个矩阵行,找出同时包含 R 和 M 的行,并删去其他矩阵行.当一个进程可以读一个变量且另一个进程可以写该变量时,如果写进程的安全级支配读进程的安全级,就可能产生潜在的隐蔽通道.通过以上分析,我们将通道分为以下类型:

① 合法通道标记为“L”;

② 无法获得有用信息的通道标记为“N”;

③ 发送进程与接收进程相同的通道标记为“S”;

④ 潜在的隐蔽通道标记为“P”.

5) 将可以构造出实际应用场景的潜在隐蔽通道标识为真实隐蔽通道.

SRM 方法的主要优点是:简单直观,实际应用广泛,有多个成功应用的例子;可以同时应用于 DTLS,FTLS 和源代码;可以同时适用于隐蔽存储通道和隐蔽定时通道.SRM 方法的主要缺点是:通过源代码构造共享资源矩阵工作量很大,且没有有效的构造工具;不能证明单个的 TCB 原语或原语对是安全隔离的,不利于增量分析新的 TCB 原语;SRM 方法是一种“保守”方法,亦即它所标识的潜在隐蔽通道往往不是真实隐蔽通道.

对 SRM 方法有下述具有代表性的推广:(1) Porras 和 Kemmerer[7]提出的隐蔽流树方法 CFT(covert flow trees);(2) McHugh[8]提出的改进共享资源矩阵法;(3) Kemmerer 和 Taylor[9]提出的模块化共享资源矩阵法.

2.1.2 语义信息流法

Tsai 等人[2]提出的语义信息流法的基础是:分析编程语言的语义、代码和内核中使用的数据结构,发现变量的可见性/可修改性;解决内核变量的别名问题,确定内核变量的间接可修改性;对源代码进行信息流分析,确定内核变量的间接可见性.他们的方法,以下简称 Tsai 方法,曾经成功地应用于 Secure Xenix 系统.

该方法的分析步骤是:(1) 选择内核原语.(2) 确定内核变量的可见性/可修改性:(i) 确定内核变量的直接可见性/可修改性;(ii) 对每个原语生成一个“函数调用依赖关系”集合 FCD;(iii) 确定内核变量的间接可见性;(iv) 解决变量别名问题;(v) 标识在原语间共享的可见/可修改变量,消除局部变量.(3) 分析共享变量,并标识隐蔽通道.

语义信息流法的主要优点是:适用于对源代码的形式化分析,搜索彻底,并能确定强制安全规则的实现是否正确;可以发现大量伪非法流;可以帮助确定安置审计代码和时间延迟变量的位置.该方法的主要缺点是:从原语出发构造函数依赖关系集合容易产生状态爆炸,在构造过程中没有退出机制,作了很多无效劳动;没有自动工具很难进行手工分析,工作量大,对分析人员要求很高;对不同的编程语言需要开发不同的词法分析器和流生成器.

2.2 隐蔽存储通道的"回溯搜索方法"

2.2.1 "回溯搜索方法"的特点

在安胜 OS v4.0 的隐蔽通道标识中,我们应用了自主研制的"回溯搜索方法".它本质上是一种语义信息流法,其分析步骤如下:

(1) 搜索内核代码中的共享变量.

(2) 使用信息流分析规则,分析包含这些变量的所有函数,确定既可读又可写的变量.

(3) 使用信息流分析规则,回溯直接或者间接调用该函数的系统调用入口函数.

(4) 将读、写该变量的系统调用与该变量共同构成五元组.应用系统强制安全规则分析两个系统调用对该变量的读写路径,判断强制安全策略的实现是否正确.

(5) 为每个潜在隐蔽通道构造场景,找出真实隐蔽通道.

我们的方法继承了 Tsai 方法的诸多优点,但同时对 Tsai 方法作了重大改进,主要包括:

1) 扩大了系统调用的选择范围.Tsai 方法认为,只有特权用户可以使用特权系统调用,且系统应当无条件信任特权用户.因此,Secure Xenix 内核中的特权系统调用如 acct(),audit(),lock(),mount(),umount(),setflbl(),shutdn(),stime()等都不必进行隐蔽通道分析.但 Linux 与 Secure Xenix 不同,上述结论不能成立.虽然 Linux 中的某些系统调用,例如 sys_acct(),确实在分析的第 1 步就需要进行capable(CAP_SYS_PACCT) 的特权检查,不具备这种特权的用户就不能执行这个系统调用.但是像安胜 OS v4.0 的 sys_mount()等系统调用,并不是一开始就进行特权检查,因此执行这些系统调用的函数就可能在特权检查之前读写共享变量.

2) 扩大了要分析的代码范围.对安胜 OS v4.0 核心代码的分析涉及到内联汇编代码,而 Tsai 方法则预先假定汇编代码是可靠的.在安胜 OS v4.0 源代码的分析中发现,有些函数,例如 spin_lock()对变量 files_lock 的修改是在内联汇编代码中实现的,因此,必须要对这些代码进行分析.

3) 相对 Tsai 方法而言,我们的方法是一种回溯的方法,因此叫作"回溯搜索法".Tsai 方法从系统调用的入口函数着手,分析该系统调用能够读写的共享变量.回溯搜索法则是从共享变量着手,找出能够读写该共享变量的所有的系统调用.

我们从共享变量出发,首先分析直接访问共享变量的函数:内核函数或系统调用,并将直接访问该共享变量的系统调用列入读写该变量的系统调用列表之中.其次,无论系统中有无系统调用,直接读写该共享变量,都要分析访问该变量的其他内核函数,判断这些函数是否读写了该共享变量.如果没有函数读或写该变量,分析到此为止,转向分析下一个共享变量.否则,找出这些函数读写这个共享变量的条件,并找出调用这些函数的函数,直到这些函数是系统调用时为止.系统调用对共享变量的读写,归根结蒂都是通过调用这些中间函数实现的.读写的区别在于,一个变量是可读的,必须能够通过系统调用入口函数返回该变量的状态;一个变量是可写的,只需要系统调用的函数能够修改它.如此,在这个阶段的分析之后,我们得到读写该共享变量的全部系统调用列表.

在分析过程中,必须记录函数调用变量和函数调用的条件,将这些条件能否成立作为判断整个信息流路径能否成立的重要判据.因此,需要构造以共享变量为中心的五元组$\langle Pa,Ca,Var,Cv,Pv\rangle$.其中 Pa 和 Pv 是完成写与读的系统调用,Ca 和 Cv 是完成写与读的条件,Var 是共享变量.潜在的隐蔽通道集合,必然是系统中所有满足这种五元组的集合的子集.

4) 在对已经标识的隐蔽通道采用审计、噪音或者延时,处理时加入处理程序的位置很清楚.我们的方法有利于选择控制共享变量的最佳点,用最有效的方式控制读写该共享变量的每个系统调用.

5) 我们的方法与 Tsai 方法最重要的区别在于,在 Tsai 方法的第 3 步,才从得到的流路径中找到涉及共享变

量的信息流.使用 FCD 分析信息流的过程缺乏退出机制,作了许多“无效”工作,且容易产生状态爆炸.而回溯搜索法只分析涉及到共享变量的函数,因为系统调用读写共享变量必然是通过这些中间函数实现的.由于采用了退出机制,回溯搜索法有效地避免了状态爆炸.

6) Tsai 方法需要更多的分析步骤与分析过程,因而对自动工具更为依赖.如果没有强有力的自动工具作为支撑,应用 Tsai 方法搜索隐蔽通道是不现实的.

2.2.2 应用“回溯搜索方法”的结果

采用回溯搜索方法,我们共发现隐蔽存储通道 18 个.根据隐蔽通道变量的分布情况,其中,进程子系统 1 个,文件子系统 12 个,IPC 子系统 4 个,网络子系统 1 个.如果根据编码特征分类,其中,事件计数型通道 9 个;资源耗尽型通道 9 个.如果根据噪音特征分类,其中,噪音通道 14 个;无噪通道 4 个.

在应用回溯搜索方法之前,我们应用改进的 SRM 方法对安胜 OS v4.0 的 DTLS 进行隐蔽通道标识,一共发现 10 个隐蔽通道.应用回溯搜索方法,再次发现了它们.

2.3 自主研制的辅助搜索工具

为方便隐蔽通道分析,增加分析的可靠性,我们研制了以下 5 种辅助工具:

(1) 共享资源矩阵传递闭包生成工具;

(2) 隐蔽流树生成器;

(3) 持久变量搜寻器;

(4) 隐蔽通道带宽计算器;

(5) 通信环境预设置和清理工具.

3 安胜安全操作系统的隐蔽通道带宽计算与测量

3.1 概 述

带宽是隐蔽通道传送数据的速度,单位是 bit/s.带宽的计算或工程测量之所以重要,是因为隐蔽通道的处理策略依赖于隐蔽通道带宽的确定.影响带宽计算的因素很多,其中最重要的有:噪音与延迟;编码与符号分布;TCB 原语的选择;测量与使用场景;系统配置与初始化;隐蔽通道的聚集.一般地,隐蔽通道的串行聚集和并行聚集都可以增加有效带宽.

目前,计算带宽的方法主要是 Millen[10]提出的形式化方法和 Tsai 与 Gligor[11]提出的非形式化方法,它们分别适用于不同的场合.一般认为,前者优于后者.

Tsai 与 Gligor 提出了一种计算隐蔽通道最大带宽的 Markov 模型,它基于以下基本假设:在隐蔽信息的传输过程中,0 与 1 的分布相同.在这种假设下,可以节省估计最大带宽的工作量.在只有发送进程和接收进程的情形下,计算无噪隐蔽通道最大带宽 $B(0)$的公式十分简单:

$$B(0)=b(T_R+T_S+2T_{CS})^{-1}.$$

其中,b 是编码因子,在实际应用中通常假设为 1.T_R 表示接收进程观察共享变量所需的时间以及接收进程建立隐蔽通信环境所需的时间.T_S 表示发送进程修改共享变量所需的时间以及发送进程建立隐蔽通信环境所需的时间.当为一个新进程分配 CPU 时,内核从当前进程向新进程执行一个“上下文切换”操作.T_{CS} 即表示进程切换或上下文切换所需的时间.

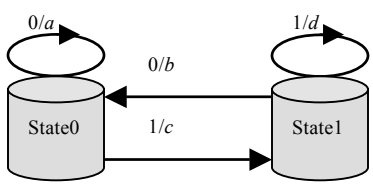

Fig.1 Two-State graph for a covert channel

图 1 隐蔽通道的两相状态图

在隐蔽信息的传输中,当 0 与 1 的分布明显不同时,可以应用 Millen 提出的基于信息论的形式化方法.该方法假设隐蔽通道是无噪通道,通道工作期间只有接收与发送两个进程,且收发同步的时间为 0.这种方法将隐蔽通道模拟为有限状态机(如图 1 所示),且这些图都是确定性的.例如,在两状态情况下,可以使用下面的两相图模拟通信过程:

图 1 中 a,b 分别是从状态 0 和状态 1 发送 1 个比特 0 所用的时间,而 c,d 分别是从状态 0 和状态 1 发送 1 个比特 1 所用的时间.

在信息论中,通道的最大传输速率定义为通道容量 C:

$$C=\lim_{t\to\infty}(\log_2 N_i(t))/t \tag{1}$$

这里,$N_i(t)$是从状态 i 开始在时间段 t 内所能传输的符号序列数,且满足下述差分方程组:

$$N_i(t)=\sum_j N_j(t-T_{ij}) \tag{2}$$

其中,T_{ij} 是由状态 i 迁移到 j 所需的时间.

为了确定通道的容量,只需求 $N_i(t)=A_ix^t$ 的渐近上界.将它代入方程组(2),即得到如下方程组:

$$A_ix^t=\sum_j A_jx^{t-T_{ij}} \tag{3}$$

上述方程组可以表示成矩阵形式:$(P-I)A=0$,其中,P 是由 x 的负幂组成的矩阵,I 是单位矩阵.$(P-I)$是奇异矩阵,所以它的行列式 $\mathrm{Det}(P-I)=0$.因此,由等式(1)可得:

$$C=\lim_{t\to\infty}(\log_2 A_ix^t)/t=\log_2 x\,.$$

一般地,x 可以有多个解.在计算最大可达带宽时,应当选取其中最大的解.

例如,在两状态图的情况下,从方程(3)可以得到

$$x^{-(a+d)}-x^{-a}-x^{-d}-x^{-(b+c)}+1=0.$$

通过实测 a,b,c 和 d 的值,使用第 2.3 节中提到的带宽计算器,可以求解该方程,计算出两状态情况下的带宽.

3.2 安胜安全操作系统的隐蔽通道带宽计算与测量

在计算安胜 OS v4.0 隐蔽通道的带宽时,我们实测了应用 Millen 公式和 Tsai 公式计算带宽时所需要的参数,根据这两个公式计算了已标识通道的带宽.结果表明,多数情况下使用两种方法得到的通道带宽比较接近.不过,有些通道发送 1 个 0 和 1 个 1 所用的时间相差较大,因此采用两种公式所得到的带宽差距较大.由于 Millen 公式得到的带宽较大,我们取 Millen 公式的计算结果作为最大可达带宽.

安胜 OS v4.0 的隐蔽通道分析表明,大多数可以构成隐蔽通道的共享变量都可以被不止一个系统调用读或写.为了使测得的带宽最大,我们实测了与每条通道相关的所有系统调用的运行时间,并且优化场景构造,用最快的系统调用、最优的工作模式构造通信场景.

实验表明,准确计算带宽的关键在于能否为隐蔽通道通信创造无噪音的通信环境和能够降低通信同步所用时间.噪音是在隐蔽通道的收、发进程工作期间,访问共享变量时产生的.因此,我们在测量最大可达带宽时,关闭所有用户进程以及不必要的后台进程,从而创造无噪音环境.同时,我们也努力改进同步技术,将同步时间降低到一个可以忽略的程度.

4 安胜安全操作系统的隐蔽通道处理方法

4.1 概 述

隐蔽通道共有 3 种处理方法:消除法、带宽限制法和审计法.虽然原则上应当尽量设法消除隐蔽通道,但实际上往往不可行.首先,有些隐蔽通道本质上无法消除;其次,消除隐蔽通道通常要求改变系统的设计与实现,代价太大,甚至不可能.

带宽限制法是事先设定可以接受的阈值,通过引入噪音或延时,将隐蔽通道的最大带宽或平均带宽降低到阈值以下.在实际应用中,应当在限制隐蔽通道带宽的同时,尽量设法不过多地降低系统的性能.

审计法通过监控系统中隐蔽通道的使用情况,对用户利用隐蔽通道传递敏感信息形成威慑.对审计机制的基本要求是保证审计机制不被旁路,不漏报也不误报.审计的难点在于:(1) 难以区分 TCB 原语的正常应用与隐蔽通道应用;(2) 难以区分隐蔽通道中的发送进程与接收进程.甚至,有些隐蔽通道是无法进行审计的.

4.2 安胜安全操作系统的隐蔽通道处理方法

安胜OS v4.0处理隐蔽通道的原则是:(1) 确保不破坏安胜OS v4.0的安全策略;(2) 保证经过处理的系统调用仍然遵守POSIX标准;(3) 隐蔽通道处理尽可能不降低系统性能.

为此,我们针对不同情况,应用下述方法处理已被标识的隐蔽通道:

1) 加入噪音,亦即通过引入随机分配算法,使系统调用返回的共享变量状态值不再具有线性规律,破坏隐蔽通道的通信机制.

例如,对于进程标识符隐蔽通道 PID,我们在系统分配进程标识符算法中引入随机分配算法,降低该通道的带宽.经过随机化处理,系统从未占用的进程标识符中随机取出一个标识符分配给新创建的进程.实际上,通过在内核函数 get_pid()中加入随机分配算法,该通道的带宽降低到了可以忽略的程度(低于 0.1bit/s).不过,加入随机分配算法之后的系统调用 fork 的运行时间延长了近 15%.

2) 引入延时,亦即将延时植入到系统调用的出错返回路径上,降低隐蔽通道的带宽.这种方法对于资源耗尽型隐蔽存储通道非常有效,不但可以有效降低带宽,而且不会明显影响系统的性能.

例如,构成系统打开文件表通道的关键在于耗尽系统打开文件表.如果耗尽了系统打开文件表,系统就返回出错号 23.我们在安胜 OS v4.0 的 get_empty_filp()函数中引入一个延时函数,使得发生这种出错时,系统在出错点延迟若干时间后返回出错信息.系统管理员可以根据系统可以容忍的信息泄漏速率来设定延时函数的参数,参数取值越大,延时越长,则隐蔽通道的带宽就越小.同时,只要系统的动作不导致出错号 23,系统的性能就丝毫不受影响.

3) 扩充审计功能

安胜 OS v4.0 中需要审计的安全相关事件分成 3 类:注册事件、使用系统的事件及利用隐蔽通道的事件.第 1 类属于系统外部事件,即准备进入系统的用户产生的事件,后两类属于系统内部事件,即已经进入系统的用户产生的事件.为了确保对隐蔽通道事件进行追踪,必须保证隐蔽通道的审计数据有足够细的粒度.

隐蔽通道审计技术的关键是如何提取隐蔽通道的审计特征.由于前面已经标识出共享变量读写路径,因此,我们在读写路径上要加上隐蔽通道审计标记,以这些标记是否出现作为是否记录审计事件的依据.

通过提取隐蔽通道特征,并将这些特征列入审计事件,就能做到审计所有的隐蔽通道行为.系统安全管理员通过定期检查审计记录,可以发现使用隐蔽通道的通信行为,在攻击者下次入侵之前采取必要的准备.我们对不宜采用降低带宽处理的每个通道都进行了审计处理.安胜安全操作系统中的审计机制如图 2 所示.

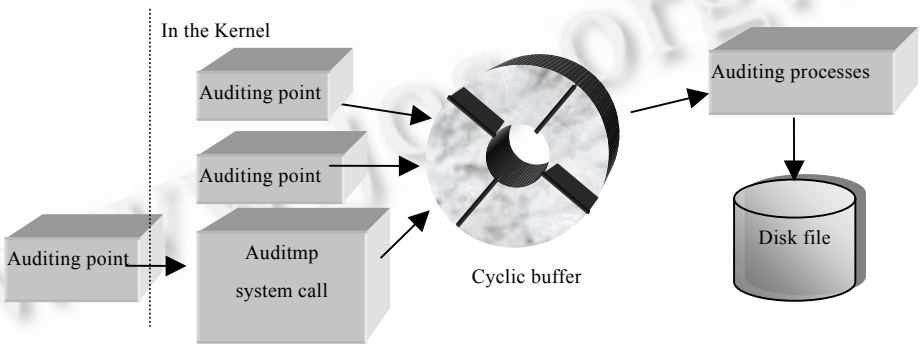

Fig.2 Auditing mechanism of covert channels in ANSHENG OS v4.0

图 2 安胜 OS v4.0 中的隐蔽通道审计机制

5 小 结

本文是首次关于安全 Linux 系统隐蔽通道分析的报告.我们通过一种搜索隐蔽存储通道的新方法——“回溯搜索方法”,在对 38 万行内核源代码、200 多个系统调用、365 个全局变量和 75 个可信进程进行分析之后,

成功地发现了 18 个真实隐蔽通道.其中,有些通道只能应用回溯分析法才能找到.

另外,在我们发现的 18 条隐蔽存储通道中,ND,SD,AT,INO,IPC 等 5 个真实隐蔽通道还未见于其他文献.此外,对被标识的隐蔽通道,我们准确地计算了它们的带宽,并根据具体情况进行了适当的隐蔽通道处理.国标 GB17859-1999[4]对于第 4 级(相当于 TCSEC 的 B2 级,CC 标准的 EAL5 级)安全操作系统的要求是:“系统开发者应彻底搜索隐蔽存储通道,并根据实际测量或工程估算确定每一个被标识通道的最大带宽”.综上所述,我们所作的隐蔽通道分析达到了国标的要求,也符合各类国际、国内的相关标准.

因篇幅所限,关于隐蔽通道原理以及安胜 OS v4.0 隐蔽通道分析的更详细的论述,可以参看文献[12,13].

References:

[1] Lampson BW. A note on the confinement problem. CACM, 1973,16(10):.613~615.

[2] Tsai CR, Gligor VD, Chandersekaran CS. A formal method for the identification of covert storage channels in source code. IEEE Trans. on Software Engineering, 1990,16(6):569~580.

[3] U.S. Department of Defense. Trusted Computer System Evaluation Criteria. DoD 5200.28-STD, 1985.

[4] General Administration of Quality Supervision, Inspection and Quarantine of the People’s Republic of China. Classfied criteria for security protection of computer information system. GB18859-1999, 1999 (in Chinese).

[5] Qing SH, Ji QG. Formal model design for secure operating systems. In: ITI 1st Int’l Conf. on Information and Communications Technology. 2003.

[6] Kemmerer RA. Shared resource matrix methodology: An approach to identifying storage and timing channels. ACM Trans. on Computer Systems, 1983,1(3):256~277.

[7] Porras PA, Kemmerer RA. Covert flow trees: A technique for identifying and analyzing covert storage channels. In: Proc. of the 1991 IEEE Computer Society Symp. on Research in Security and Privacy. 1991. 36~51.

[8] McHugh J. Covert channel analysis: A chapter of the handbook for the computer security certification of trusted system. NRL Technical Memorandum 5540:062A, 1995.

[9] Kemmerer RA, Taylor T. A modular covert channel analysis methodology for trusted DG/UX. In: Proc. of the 12th Annual Computer Security Applications Conf. Washington: IEEE Computer Society, 1996. 224~235.

[10] Millen JK. Finite-State noiseless covert channels. In: Proc. of the Computer Security Foundations Workshop. Franconia: IEEE Computer Society, 1989. 81~85.

[11] Tsai CR, Gligor VD. A bandwidth computation model for covert storage channels and its applications. In: Proc. of the IEEE Symp. on Security and Privacy. OakLand: IEEE Computer Society, 1988. 108~121.

[12] Qing SH, Liu WQ, Weng HZ, Liu HF. Operation System Security. Beijing: Tsinghua University Press, 2004 (in Chinese).

[13] Zhu JF. Covert channels analysis for the ANSHENG operating system. Technical Report, Beijing: Engineering Research Center for Information Security Technology, the Chinese Academy of Sciences, 2003 (in Chinese).

附中文参考文献:

[4] 中国国家质量技术监督局.中华人民共和国国家标准:计算机信息系统安全保护等级划分准则.GB17859-1999,1999.

[12] 卿斯汉,刘文清,温红子,刘海峰.操作系统安全.北京:清华大学出版社,2004.

[13] 朱继锋.《结构化保护级》安全操作系统隐蔽通道分析处理报告.北京:中国科学院信息安全技术工程研究中心,2003.