第 32 卷 第 8 期 计 算 机 工 程 2006 年 4 月
Vol.32 № 8 Computer Engineering April 2006
·发展趋势/热点技术· 文章编号：1000—3428(2006)08—0001—03 文献标识码：A 中图分类号：TP309

隐蔽通道标识与处理

刘文清[1,2,3]，陈 喆[1]，韩乃平[2]

(1. 解放军信息工程大学电子技术学院，郑州 450004；2. 上海中标软件有限公司，上海 200233；
3. 中国科学院软件研究所信息安全技术工程研究中心，北京 100080)

摘 要：标识与处理隐蔽通道是美国橘皮书 TCSEC 对 B2 及以上级别安全产品的关键评估要求，也是国际标准 CC 评估 EAL5 及以上系统的关键指标。目前，隐蔽通道分析是阻碍我国开发高安全等级信息系统的主要瓶颈。该文介绍了迄今为止国际上标识与处理隐蔽通道的主流方法，探讨了使用这些方法对安全信息系统进行隐蔽通道分析的可行性。

关键词：隐蔽通道；安全操作系统设计；标识与处理

Identifying and Dealing with Covert Channel

LIU Wenqing[1,2,3], CHEN Zhe[1], HAN Naiping[2]

(1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004; 2. China Standard Software Co.,Ltd., Shanghai 200233; 3. Engineering Research Center for Information Security Technology, Software Institute of Chinese Academy of Sciences, Beijing 100080)

【Abstract】 Recognition and dealing with covert channel is the key requirement of TCSEC to evaluate the B2 or above products, as well as the key requirement of international standard CC to evaluate the EAL5 or above products. Now covert channel analysis is the main bottleneck of development of security information system with high security level in our country. This paper describes the mainstream methods to identificate and deal with covert channel up to date and discusses the availability of these methods in the security-product-developing.

【Key words】 Covert channel; Design for security OS; Recognition and dealing with

1990 年，Tsai、Gligor 和 Chandersekaran[1]把隐蔽通道定义为“给定强制安全模型 M 及其在操作系统中的解释 I(M)，两个主体 I(Si)和 I(Sj)任何潜在通信是隐蔽的，当且仅当模型 M 相应主体 Si 和 Sj 之间的任何通信在 M 中是非法的”。Tsai 等人的这个定义既考虑到隐蔽通道与多级安全系统的强制存取策略的关系，也考虑到隐蔽通道与该策略在系统内的实现的关系，成为美国国家安全局推荐使用的定义，进而成为国际公认的隐蔽通道定义[2]。我国的 GB17859-1999 把隐蔽通道定义为“允许进程以危害系统安全策略的方式传输信息的通信信道”。该定义应与 Tsai 的定义保持一致。

1975 年，Lipner 根据通信双方传递信息所用媒介的不同，把隐蔽信道分为隐蔽存储通道（Covert Storage Channel）和隐蔽定时通道（Covert Timing Channel）两类。隐蔽存储通道包括所有如下的通道：允许一个进程直接或间接地写一个存储位置，而另一个进程可以直接或间接地读这个存储位置。隐蔽定时通道则指的是如下的通道：允许一个进程通过调节自己对系统资源的使用向另一个进程发送信息，后者通过观察响应时间的改变而获得信息。目前已知的隐蔽通道绝大多数是存储通道。存储通道和定时通道的划分不是绝对的，有些通道兼备二者的特征。对任何系统而言，只要处理器是共享的，就至少存在一个共享属性：CPU 的响应时间。收方进程通过监视时钟就能侦查到响应时间的变化，定时通道比存储通道更难以避免。

研究高安全性的可信计算机系统必须进行隐蔽通道分析，尽可能地搜索出所有的隐蔽通道，测量或估计它们的带宽并给予适当的处理，以控制隐蔽通道对系统的破坏。隐蔽通道分析是阻碍我国开发高安全等级信息系统的主要瓶颈。

1 隐蔽通道分析

隐蔽通道分析包括标识通道、计算通道带宽、处理通道和测试通道等 4 个阶段。

标识通道是隐蔽通道分析过程中最关键的阶段。系统内部的非法信息流是产生隐蔽通道的根源，标识隐蔽通道就是从系统描述中抽象出潜在的信息流。

计算通道带宽就是理论估计和工程测量隐蔽通道的最大可达带宽。带宽是隐蔽通道最重要的参数，描述利用该信道每秒钟可以传递多少比特信息（bps）。带宽越大，隐通道潜在的安全威胁也越大。Tsai 曾提出了一种用 Markov 模型计算带宽的方法[4]，并得到了一个可用于工程估计带宽的公式 $B=b*(Tr+Ts+2Tcs)^{-1}$。其中 b 是信息编码因子，Tr 是平均设置一次读环境并读一个 0 或 1 所用时间，Ts 是平均设置一个 0 或 1 所用时间，Tcs 上下文切换时间。J.Millen 则基于信息论的方法用有限状态图描述通信的过程，能从理论上求解最大带宽[3]，计算公式为 $B=\log_2 X$。其中 X 是差分方程 $1-X^{-d1}-X^{-d2}=0$ 的解，d1 和 d2 是发送一个 0 和 1 所用平均时间。

处理隐蔽通道的手段包括清除隐蔽通道、降低带宽和审计使用等。处理隐蔽通道要考虑到所处的安全环境、要保护的数据本身的特性和系统本身的特征。尽管隐蔽通道的存在

基金项目：国家“863”计划基金资助项目(2002AA1Z2101，2004AA1Z2020)；中科院知识创新工程基金资助项目(YC2K5609)
作者简介：刘文清(1967—)，男，博士、副研究员，主研方向：信息系统安全理论与技术；陈 喆，博士生；韩乃平，高工
收稿日期：2005-06-15 **E-mail：**wenqing.liu@cs2c.com.cn

使系统面临严峻的安全威胁，但是有些存储和定时通道是系统正常操作的必要组成部分，处理不慎会导致系统崩溃或者性能大大降低。通常采用的处理策略为：在具体环境允许的情况下，应当尽可能使用消除法对非法信息流进行控制。不得已的情况下采用降低带宽法，即设法降低隐蔽通道的信息传输速率，从某种意义上说就是噪音化，使得操作结果难以预测。

隐蔽通道处理的下限是必须采用审计法对非法信息流进行监督改进审计粒度、设置时间延迟降低带宽和消除内核变量共享性来处理隐蔽通道，这必然影响系统的执行效率。因此系统安全设计存在一个安全和效率的权衡问题，以往的实践已经证明，完全清除隐蔽通道是不可能的[3]。

按照美国橘皮书 TCSEC 的要求，在系统隐蔽通道分析工作的最后，还必须完成测试文档，清晰地说明隐蔽通道标识、计算和处理的过程。目前，美国通过橘皮书 B2 及 B2 以上级别认证的安全操作系统都具备专门的隐蔽通道分析报告。

一个信息系统，从提出最初的设计要求到最终完整的代码实现，要经过建立模型、编写规范说明、编写代码和测试等多个阶段。出现每个阶段的安全漏洞都可能会被带入下一阶段，并导致最终实现的系统中出现隐蔽通道。因此，隐蔽通道分析应当在系统实现的各个层次上进行，该结论已在实践中得到证明[3]。隐蔽通道分析其常用的资源包括：

(1)描述 TCB 原语、TCB 参数、CPU 和 I/O 进程指示的机器代码的系统参考手册；

(2)描述性顶层规范(DTLS)、半形式化顶层规范(SFTLS)和形式化顶层规范（FTLS)；

(3)系统的源代码（Source Code)。

原则上，由于系统内部的一致性，高层的设计漏洞肯定会被带到低层中（例如规范中的漏洞会被带到代码实现中)，因此设计漏洞发现得越早，纠正错误所需的代价就越小。如果假定模型和机器代码中不存在最终导致代码中隐蔽通道的设计漏洞，那么顶层规范和代码就成为分析的重点环节。显然，在顶层规范中标识设计漏洞并加以修改的代价，要比反复修改调试源代码的代价要低得多。

然而在实践中，无论是 DTLS 还是 FTLS，它们对系统的描述相对最终的代码实现而言总归是较粗略的，不可能包括数据结构等实现细节，加上无法确保规范与代码具有严格的一致性，单纯地分析规范并不能保证发现所有的隐蔽信道。要完全彻底地找出系统中所有隐蔽信道，必须对全部源代码进行搜索和分析。因此，对安全系统隐蔽信道分析，一般在描述性顶层规范、半形式化顶层规范和源代码层等 3 个层次上进行。

2 隐蔽通道标识

自 Lampson 以来，美国学者先后设计了多种隐蔽通道标识的方案，较有代表性的有信息流法（Information Flow)、无干扰法（Noninterference)、共享资源矩阵法（SRM)、语义信息流法（Semantic Information Flow Analysis)、隐蔽流树法（CFT 法)、改进 SRM 法（Extended SRM）等，这些方法从本质上看都是信息流法，其差异只是着眼点不同。现在的分析隐蔽通道的主流观点是：只有采用信息流分析技术，才能发现隐蔽通道。因此，对隐蔽通道的分析本质上就是分析系统中的非法信息流。

2.1 Denning 的信息流法

D.Denning 给出的信息流法[5]，详细阐述了信息流的概念和并提出了信息流的格模型。在源代码中，信息流描述了两个变量之间的因果关系。在任一变量 b 或变量 a 的函数中，如果能通过观察新状态下 b 的值推断出旧状态下 a 值的信息，就存在从 a 到 b 的信息流。信息流分析过程包括找出信息流并检验它是否违反信息流规则。

分析的过程是：每次分析一个函数，直到分析完函数中的每个表达式，并把每一对变量之间的信息流写成一个流语句（例如：若有语句“ a:=b ”，则信息从变量 b 流向变量 a，记为 a←b)，这样，就从一个给定的函数产生很多流语句；接着根据信息流格模型规则“ 若信息从 b 流向 a，则 a 的安全级必然支配 b 的安全级 ”；然后对这些流加以检验，找出非法流；如果非法流是真实存在的，就标记为隐蔽通道。

信息流分析法理论上可以找出系统中所有的信息流安全缺陷，而不会漏过一条非法流，能彻底地搜索出所有的隐蔽通道。尽管该方法虽然从未真正被用于分析操作系统实际上，却具有重大的理论意义，后来出现的实用分析方法则都借鉴信息流分析法。

2.2 Kemmerer 的 SRM 法

与 Denning 的信息流法不同，Kemmerer 的共享资源矩阵（SRM）方法[6]一经提出就立刻被用于工程实践中，并在好几个安全操作系统的开发中（例如 UCLA 数据安全 Unix 和 DG/UXTM 等）发挥了重要作用。SRM 方法可以用于描述性顶层规范、形式化顶层规范和源代码。其基本思想是：既然产生隐蔽信道是因为系统中存在共享资源，那么如果找出所有用于读写的系统资源和操作，就能找到所有隐蔽信道。采用 SRM 方法进行隐蔽信道分析步骤如下：

(1)列举一个主体可以访问或修改的所有共享资源，检查每个资源，确定它是否可能被用于在主体间隐蔽地传递信息；

(2)构造矩阵并且根据规范或代码确定操作原语对该变量的读写关系；

(3)生成该矩阵的传递闭包，从而得到所有的间接读写操作；

(4)分析该矩阵，得到形如<Pa，Var，Pv>的三元组，其中 Pa 和 Pv 是执行修改和访问动作的原语，Var 是二者共享的全局变量。

SRM 方法的不足之处是：(1)不能证明单独一个原语是否安全；(2)在源代码级构造共享资源矩阵工作量大，手工构造效率低，且容易出错。该方法优点不仅可以用于代码分析，还可以用于规范分析甚至模型和机器代码分析，它具有广泛的适用性。

2.3 Tsai 的信息流法

Tsai 等人提出的改进信息流法[1]秉承了 Denning 信息流法和 SRM 法的某些特点，从某种意义上说兼备了二者之长。该方法的分析过程类似 SRM 法：

(1)选择系统中用户可见的原语；

(2)进行原语可读/可写变量的分析，分析出每个原语可读访问的所有变量和每个原语可写访问的所有变量，包括原语直接可读/可写变量和间接可读/可写变量（其中，原语间接可读/可写变量分析又包括函数间调用关系分析、信息流分析、变量换名分析和函数间信息流分析)，根据分析结果，找到原语间可读/可写的共享变量，并且将它们表示成原语流的形式；再对原语流进行分析，过滤掉不可能产生隐蔽信道的原语流，以缩小用于隐蔽信道分析的原语流数量；

(3)对这些原语流进行隐蔽信道分析，找出系统中的隐蔽

信道。

与 Denning 的信息流法相比，Tasi 信息流法在确定共享变量的安全级之后（从流内客体的安全级中确定）才使用信息流规则，从而避免了后者向变量硬加安全级的弱点，因此不会出现大量伪非法流。

与 SRM 法相比，Tasi 信息流法分析原语和变量之间读写关系的步骤更细致、准确，特别是用信息流关系取代“生成矩阵传递闭包”保证了搜索工作的彻底性。此外，Tasi 信息流法能准确找出内核代码中共享变量被查看/变更的位置，从而能准确确定安置审计代码和时间延迟变量的位置以及实施强制策略的存取检查的位置。

该方法的不足在于它仅适用于源代码层，即只能在系统开发的后期发挥作用。

3 隐蔽通道处理

对隐蔽通道的常见处理技术包括消除法、带宽限制法和威慑法等，美国橘皮书 TCSEC 建议结合使用这 3 种方法。隐蔽通道处理的基本原则是：

(1)通道带宽低于某个预先设定值 b 的隐蔽通道是可以接收的。

(2)带宽高于 b 的隐蔽存储通道都应当可以审计。所有不能审计的存储通道的带宽要记入文档。这使得管理员可以察觉并从程序上采取纠正措施对付重大的威胁。

(3)带宽高于预先设定的上限 B(B>b)的隐蔽通道代表重大威胁，应当尽可能把消除或者把其带宽降低到 Bbps 以下。

根据橘皮书发布时的具体情况，还具体提出了处理策略：

(1)对于通道带宽大于 0.1bps 的通道，只要有可能，就要采用消除法；

(2)只要有可能，就采用带宽限制法每个无法消除的通道，使之最大带宽降低到到 1bps 以下；

(3)使用威慑法（即审计法）处理所有带宽大于 0.1bps 的通道；

(4)对于带宽低于 0.1bps 的通道，可以不作处理。

按照这一策略，允许存在不可审计的存储通道（策略(4)，并且允许存在带宽高于 B=1bps 的存储通道和定时通道（策略(2)）。建议的阈值 b=0.1bps 和 B=1bps 可以根据特定策略进行调整。确定上述阈值的依据是：隐蔽通道处理会影响系统性能，而对于多数系统环境而言，1bps 的带宽都是可以接受的。不过橘皮书认为带宽高于 100bps 的隐蔽通道不能接收，原因是在 20 世纪 80 年代初，老式计算机的终端的运行速度都是在这个数量级。对现在的系统设计者来说，B 和 b 值的设定要根据具体的安全需求和应用环境等情况。

3.1 消除法

消除隐蔽通道需要改变系统的设计和/或实现，这些改变包括：

(1)消除潜在隐蔽通信参与者的共享资源；

(2)清除导致隐蔽通道的接口和机制；

(3)增加存取控制策略等。

3.2 带宽限制法

带宽限制的策略是设法降低通道的最大或者平均带宽，使之降低到一个事先预定的可接受的程度。限制带宽的方法有：

(1)故意引入噪音，即用随机分配算法分配诸如共享表、磁盘区、PID 等共享资源的索引，或者引入额外的进程随机修改隐蔽通道的变量。

(2)故意引入延时。

3.3 威慑法

威慑法是假定用户都知道存在哪些通道，但是系统采用某种机制让恶意用户不敢使用这些通道。最重要的威慑手段是通道审计，使用有效的审计手段监视通道的使用情况，让隐蔽通道使用者知难而退。隐蔽通道审计要求审计数据记录足够的数据，确保能够：(1)标识每次隐蔽通道的使用事件；(2)识别通道的发送者和接收者，即通道的用户。

设计隐蔽通道审计机制的关键是：为了保证发现所有使用隐蔽通道的事件，要明确审计机制应该记录什么事件，和审计工具应该维护什么数据。由于标识隐蔽通道可以归结为确定<PAh, Var, PVi>三元组，因此记录所有包含<PAh, Var>和<PVi, Var>数据的事件就是隐蔽通道审计的充要条件。

4 讨论

标识与处理隐蔽通道是安全操作系统设计的关键技术，也是安全操作系统评测的重要指标。橘皮书提供的隐蔽通道带宽阈值 b(0.1bps)和 B(1bps)并不令人满意，原因是隐蔽通道造成的威胁与特定的应用环境有关，只有根据特定应用环境里的威胁情况才能评价阈值 b 和 B 是否恰当，所以在设计和评估中不宜指定阈值。最理想的情况是系统向系统安全管理员提供可调的隐蔽通道延迟值，由管理员决定阈值选取。

隐蔽通道造成的威胁还跟隐蔽通道本身的特点有关。例如：

(1)有些隐蔽通道的最大可达带宽可以很高，但系统正常负载情况的噪音和延迟使该通道的实际带宽低于可以接受的上限 B。

(2)使用场景比较简单的隐蔽通道更容易被使用。

(3)出现在系统低层的隐蔽通道无法审计。

(4)有的隐蔽通道可以集成起来使用以获得较高的带宽，也有的隐蔽通道不能集成。

参考文献

1 Tsai C R, Channdersekaran G V D. A Formal Method for the Identification of Covert Channels in Sorce Code[J]. IEEE Trans. on Software Engineering, 1990, 16(6): 1990.

2 International Standards Organization. ISO/IEC15408-99.Information Technology Security Techniques Evaluation Criteria for IT Security[Z]. URL: http://csrc.nist.gov/cc, 1999.

3 Millen J. 20 Years of Covert Channel Modeling and Analysis[C]. IEEE Computer Society Symp. on Security and Privacy, 1999.

4 Tsai C R, Gligor V D. A Bandwidth Computation Model for Covert Storage Channels and Its Applications[C]. Proc. of IEEE Symposum on Security and Privace, Oakland, 1988.

5 Denning D. A Lattice Model of Secure Information Flow[J]. Communication of the ACM, 1976, 19 (5): 236-250.

6 Kemmerer R A. Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels[J]. ACM Transactions on Computer Systems, 1983, 1(3): 256-277.