浙江大学

课程名称：	**信息安全原理**
报告题目：	**信息流和隐蔽信道**
指导老师：	**蔡亮**
学　　院：	**计算机科学与技术学院**
组　　号：	**第一组**

目录

信息流和隐蔽信道

1. 隐蔽信道的发展历程

我国的《计算机信息系统安全保护等级划分准则》(GB17859-1999)对隐蔽信道做出了如下定义：隐蔽信道是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。在维基百科中，其定义则更加直观明了：隐蔽信道是一种在原本不允许通信的进程之间传输信息的方法和能力。毋庸置疑,隐蔽信道是对计算机安全的一种重大威胁,会造成大量的信息非法泄密。美国的《可信计算机系统评估准则》(TCSEC)中明确提出安全要求 B2 级以上的系统必须进行隐蔽信道的分析，在识别隐蔽信道的基础上，对隐蔽信道进行评估和处理。

图灵奖得主 Butler Lampson 1973 年在《Communications of the ACM》期刊上发表了开创性的论文《A Note on the Confinement Problem》，首次提出隐蔽信道的概念。他对隐蔽信道的定义为“不是被设计或本意不是用来传输信息的通信信道”，该定义点明了隐蔽信道的本质和其被创造出来的重要原因。

TCSEC 标准使用 TCB(trusted computing base,可信计算基)表示计算机系统中所有保护机制的总和,负责执行计算机安全策略。运用 TCB 三元组表示隐蔽信道为：$\langle variable, PA_h, PV_i \rangle$。其中 *variable* 是系统中的变量；$PA_h$ 是修改这个变量的 TCB 原语且具有较高的安全级；PV_i 是感知、观察这个变量的 TCB 原语且安全级较低。在强制访问控制模型下，隐蔽通道能够实现从高安全级主体向低安全级主体的通信，破坏力极大。[1]

目前在隐蔽通信的研究中，通常将其分为存储隐蔽信道和时间隐蔽信道两大类。两种隐蔽通信在本质上没有太大的差别，只是对于 *variable* 有不同的选择而已。例如，在存储隐蔽信道中，*variable* 一般为双方能够修改和感知的共享资源；而在时间隐蔽通道中 *variable* 可能表示 CPU 时间或者响应时间等属性。其中，显然的一点是时间隐蔽信道不能长时间地存储信息，更类似于一种信息流。接收者必须及时接收发送者发送的信息，具有较高的时效性。一般认为，时间隐蔽信道更加复杂，更难以侦测。TCSEC 标准要求 B2 级安全系统进行存储隐蔽信道分析，而更高级别的 B3 级和 A1 级安全系统才要求同时进行存储和时间隐蔽信道分析。

2. 隐蔽信道的原理与攻击

隐蔽信道的概念形成于上世纪 70 年代初期，隐蔽信道是一个能绕过系统制定好的安全机制的通信通道，以违法信息安全策略的方式传输信息，发送、接收双方利用合法操作实现隐蔽传送信息的目的，具有抗截获、抗检测等特点。

隐蔽信道产生的原因主要是可信计算基存有安全缺陷及系统资源的共享,任何底层的设计漏洞，都会被带到高层中，出现每个阶段的安全漏洞都可能导致最终实现的系统中出现隐蔽信道，隐蔽信道分析应当在信息系统的各个层次上进行。进行隐蔽信道分析需要标识通道、计算通道带宽。按照 TCSEC 要求，起始安全级为 B2 以上需要进行隐蔽信道分析，并计算隐蔽信道带宽。

以嵌入隐蔽信息的方式进行划分，隐蔽信道分为存储隐蔽信道和时间隐蔽信道。

2.1 存储隐蔽信道

存储隐蔽信道是指发送方将信息直接或间接地写入某些存储位置(内存单元、外存空间、网络数据包等)，而接收方通过读取此存储位置的信息，按照双方约定好的规则还原出来双方发送的信息。时间隐蔽通道是指发送方将信息嵌入到与时间有关的参数中，在信息交互中并不改变信息的内容，接收方通过预先定义好的规则将顺序、间隔、周期变化等与时间有关的参数来还原信息，达到隐蔽传输信息的目的。

如磁盘隐蔽通道、打印机连接隐蔽信道、目录结构隐蔽信道等都是储存隐蔽信道，磁盘隐蔽通道通过发送进程修改磁臂的方向,低级安全进程可通过自己的运行状态和周围环境来感知高安全级进程传递的这个信息，接收到后解码成信息，避开安全检查机制，发送进程使打印机处于忙碌或空闲状态让接收进程观察到,发送进程创建或删除目录让接收者观察目录是否存在，总之，以上各例都是通过发送进程使一个状态变量发生变化，接收进程对其观察来得到结果。[2]

2.2 时间隐蔽通道

时间隐蔽通道的做法就是创建一个准确的时间基准,它使得接收进程能够计算两个相继事件的相对时间，例如通过间隔一定的时间发送进程使用 CPU，让接收者能够观察到 CPU 的使用情况，如果接收程序没法访问时钟来判断时间间隔，还可以通过其他方式，例如发送方可以以每秒十个字符的速度输入字符串，那进程就可以得到以 0.1 秒为间隔的时间计时器，每受到一个字符时间就过去了 0.1 秒。此外，在多处理器系统中，一个程序可以通过循环程序的循环次数来为了一个进程确定时间间隔。[2]

2.3具体攻击手段

2.3.1 利用网络协议

现在由于网络技术的大幅度发展，整个因特网都可以看作是一个巨大的计算机系统，网络中的各种协议以及基础设施就是共享资源,因此在网络安全中隐蔽信道就成为了其中最常见的有效攻击手段。

IP 中的隐蔽信道既有存储隐蔽信道又有时间隐蔽信道，IP 的存储隐蔽信道主要利用网络协议漏洞，利用 IP 数据包报头中暂未被网络协议所使用的的各字段隐藏信息，将信息嵌入其中,因某些选项段字域空闲不用,且长度可变,安全机制对此不做检测来形成隐蔽信道，但后来的研究表明该类隐蔽信道可以被防火墙的流量正规化技术所消除，即将进出 IP 数据包的冗余位强制使用相同的格式改写，有效地限制了 IP 隐蔽信道的使用。基于网络的包间时延 IP 时间隐蔽信道则不修改数据包本身，将隐蔽传输的信息调制成与时间相关，发送方通过改变网络包的间隔、速率、顺序等方式将信息嵌入，接收方按照相同规则检查、度量这些网络包的时间属性进行解析获得信息，由于包交换网络的时延因素较为复杂，且时间间隔复杂多变，使得这种隐蔽信道更不容易被人发现，成为 IP 时间隐蔽信道研究的主流。

IP 时间隐蔽信道中有代表性的就是包间时延隐蔽信道，它是指发送方将秘密信息调制到数据包时间间隔，接收方记录网络包到达时间，通过时间间隔还原隐蔽信息，可根据时间参数不同划分为三类：IP 网络包隐蔽信道 一定时间间隔内，发送数据包为 1，不发送为 0；击键间隔隐蔽信道，Telnet 等应用每次击键都发送一个网络数据包，当网络包时间间隔大

于设定时间间隔是为 1，反之为 0；重放时间隐蔽信道 预先收集大量正常网络发送行为的包间隔，按间隔长短分为两个集合（较小的归入集合 1，大的归入集合 2），发送方发送 0 时从 1 中选取一个时间间隔，发送 1 时从 2 中选取 1 个时间间隔。[3]

2.3.2 利用操作管理系统

操作系统、数据库系统、网络、分布式数据库管理系统中都可能存在隐蔽信道，操作系统处于计算机系统的最底层，它的安全是系统中一切安全的基础，也就是说操作系统的安全问题是一切安全的问题中最核心的问题。而隐蔽信道是一种通信通道，但它不是系统设计者设计的打算用来通信的信道，因而它能绕过系统强制安全机制的检查，使得进程能以违反系统安全策略的方式传递信息，从而对计算机系统的安全造成很大的威胁。[4]

攻击的主要类型：

Hardware	Storage	Value
		Transition
	Timing	Value
		Transition
Operating System	Storage	Value
		Transition
	Timing	Value
		Transition
Network	Storage	Value
		Transition
	Timing	Value
		Transition

1） 硬件存储值/基于转换的攻击

在本类攻击当中，各方的进程都可以接触到在特定硬件当中存储的数值，随后一个进程可以存储一些能表现需要传输的信息的数据，而另一个进程则可以读取这些数据并且进行翻译，以此实现通信。若接收方无法直接接触到储存器当中的数值，则可以采用 transition based attack，如判断寄存器当中是否存有数据，若有则判断为 1，否则判断为 0。

2） 硬件定时基于值的攻击

在这种类型的信道中，发送方能够随时调用接收方，但是其是基于观察到的硬件的反应时间或其他参数。如果表示硬件性能的参数数值可预测地增加，则可以将其除以一个预设的值，以便以传输速率为代价来降低噪声。

3）操作系统存储基于转换的攻击

我们假设两个进程在不同的安全级别运行，并且发送方可以对接收方接收的数据进行一定的改变。这是一个比任意修改（比如直接储存数值）更弱的假设，因此需要更少的特权。由于发送方无法将状态更改为任意值，因此用于传输信息的状态的更改。此攻击的一种手法会使用文件系统。例如，发送方可以更改或不更改文件的属性以传输 1 或 0。在这种情况下，接收方不需要访问文件本身。它需要一种机制来读取该特定属性；例如，通过目录列表进行查询。

4） 操作系统定时基于值的攻击

该攻击方法要求发送方有能力发出一个可以调用或者是传送信号的进程，在此攻击中，发送方和接收方都不需要“共享资源”的“写入”访问权限。相反，发送方监视操作系统内的特定数据结构，并且每当它检测到适当的值时，它就调用（invoke）也可以读取该共享数据结构的接收方进程。请注意，在许多安全模型中，允许更高安全级别的进程执行较低安全级别的进程。

为了实现此攻击，我们使用系统计时器作为共享资源。它经常变化，许多进程都可以读取它。为了隐蔽地发送一些数据，发送器使用奇数定时器值来发送“1”和偶数定时器值以发送“0”。我们还包括通过将时间除以固定值（INT）来调整定时器频率的功能。发送器的算法如下所示。该信道有噪声，因为定时器的值可能会在接收器读取值之前发生变化，从而导致不正确的比特值。[5]

```
1.  for each current_bit in data
2.  {
3.    while (true)
4.    {
5.       milliseconds := current_time();
6.       if (current_bit == (milliseconds / INT) mod 2)
7.       {
8.          invoke_receiver();
9.          break;
10.      }
11.    }
12. }
```

3 隐蔽信道的防治

隐蔽信道的存在是安全操作系统所面临的一个重要威胁。因此,高安全等级的操作系统都要求对隐蔽信道进行分析。隐蔽信道的分析包括 3 个步骤:

1) 搜索所有潜在的隐蔽信道;
2) 根据搜索的结果,计算和测量隐蔽信道的带宽;
3) 根据计算或测量的结果, 对隐蔽信道进行相应的处理。

隐蔽信道的分析可以在 3 个层次上进行: 描述性顶层规范(DTLS)、形式化顶层规范(FTLS)和源代码。 在隐蔽信道的分析中,最复杂、最困难的是隐蔽信道的标识。 在理论上,缺乏有效的方法; 在实践中,工作量大,缺乏有效的自动分析工具。

目前主流的隐蔽信道标识方法有下面几种:语法信息流法、语义信息流法、回溯搜索法、共享资源矩阵法、无干扰分析法等。 下面将依次对这些方法进行介绍。

3.1 共享资源矩阵法及其拓展

1) 共享资源矩阵法

这其中最成功的是共享资源矩阵法,它可以同时应用于 DTLS ,FTLS 和源代码,并且已经有 多个成功应用的例子。隐蔽流树法是共享资源矩阵法的一种拓展它采用了树数据结构模拟共享资源属性之间的信息流,通过遍历树可以构造隐蔽信道 的实际应用场景。

Kemmerer 提出了共享资源矩阵法, 又称为 SRM （Shared Resource Matrix）方法。 它的分析步骤是:

1)分析所有的 TCB 原语操作,确认通过 TCB 接口用户可见/可修改的属性;

2)构造共享资源矩阵。 矩阵的行对应于 TCB 原语, 列对应于可见/可修改的属性。 如果一个原语可以读一个属性,则将相应的矩阵项标记为 R; 如果一 个原语可以修改一个属性,则将相应的矩阵项标记为 M;

3)对共享资源矩阵完成传递闭包操作。

最后,如果某个共享资源属性所在的行中同时 包含有 R 和M ,则该属性可能被用作隐蔽信道。

根据对每个隐蔽信道发送者和接收者的分析,可以得出以 下 4 种可能的结论: 1)该通道是无用的,即在发送者和接收者之间 存在另外的合法的通道; 2)该通道不能向接收者提供任何有用的信息; 3)该通道要求发送者和接收者相同; 4)该通道可以被用于非法信息传递。[6]

SRM 方法会产生大量潜在的隐蔽信道。由于没有自动分析工具可以应用,用户必须手工对每个隐通 道进行分析,判断它是否是一个真实的隐蔽信道。 手工分析工作量大,并且容易出错。它在效率以及准确性方面仍然存在较大的问题, 此外,SRM 方法不能帮助用户构造隐蔽信道实时应用的场景。

2) 隐蔽流树法

隐蔽流树法 Porras 和 Kemmerer 提出了隐蔽流树法,又称为 CFT 方法。 它可以看作是对 SRM 方法的补充 和发展。 构造隐蔽流树所需的信息与构造共享资源 矩阵所需的信息基本相同,每个操作包含一个引用列表、一个修改列表和一个返回列表。CFT 方法的分析步骤为 1)构造一棵隐蔽流树。 树的根结点为待识别的隐蔽信道,左子树为发送者的修改操作,右子树为接收者的识别操作序列,包括直接识别和间接识别; 2)对隐蔽流树进行遍历,产生支持通信的操作 序列列表; 3)分析这些操作序列,并构造支持隐蔽通信的应用场景。

CFT 方法采用树生长的方式表示隐蔽信道的形成过程,具有较低的误报率, 它能发现其他方法检测不到的隐蔽信道, 并且易于直观地描述隐蔽信道利用的情况。

它的缺点是隐蔽流树生长的非常快,一个简单的文件系统其所对应的隐蔽流树已经超过了 104 个结点。构造和遍历这样一棵庞大的隐蔽流树,不但需要消耗 大量的 CPU 和内存资源,而且也不利于隐蔽信道的图形化表示。因而, 此外,用户也需要手工对算法产生的隐蔽信道操作序列进行分析。

3.2 语法信息流法与语义信息流法

语法信息流法是较早用于系统进行隐蔽信道分析的方法之一。语法信息流法进行隐蔽信道检测的基础就是信息流，它根据赋值语句、条件语句等语句的特点， 对应生成信息流，并且定义信息流安全策略。根据信息流及安全策略，针对形式化顶级规范或源代码生成信息流公式，并通过证明公式的正确性来判断一条 信息流是非法信息流还是伪信息流，进而用于判断该条信息流是否会产生真实隐通道。语法信息流法能够涵盖所有可能产生隐蔽信道的非法信息流，但是通过对每条信息流显式或隐式地赋予特定的安全等级会存在风险，最终导致产生大量的伪非法信息流，还需要进一步借助语义分析来手工消除伪隐蔽信道路径。

语义信息流法是对语法信息流法的改进，具体改进体现在前者增加了语义分析，弥补了语法信息流的一个重要缺陷。语义信息流法首先会选择源代码中需要进行隐蔽信道分析的内核原语；然后通过语义分析，来确定共享变量的可见性 和可修改性，并借助于信息流分析，确定内核变量的间接可见性与可修改性，消除局部变量；最后分析共享变量，检测源代码中可能存在的存储隐蔽信道。语义信息流法能够应用于源代码级的隐蔽信道分析，并能正确利用系统中定义的安全策略 来进行隐蔽信道路径的判断。但是语义信息流法需要借助于构造函数调用依赖集合 来进行语义分析，工作量很大且容易造成状态爆炸，而且该方法误报率很高，实际应用意义不大。[7]

较基于共享资源矩阵的检测技术，直接基于信息流的检测技术（语法分析法、语义分析法）能够方便地表示共享变量的可见性和可修改性，并且易于将安全策略与信息流相结合，有利于进行真实隐蔽信道路径与伪隐蔽信道路径的判断。

【补充 1：语法信息流法简单样例】

例如在函数 func 中，有以下情况：

```
1.  func()
2.  {
3.        /*……*/                  //其他代码
4.         Va=Vb;                 //第一次出现对 Va 的赋值（1）
5.        /*……*/
6.         Va=Vc;                 //第二次出现对 Va 的赋值（2）
7.  }
```

(2)处产生的信息流(Vc-> Va，func)要覆盖(1)处的信息流(Vb->Va，func)，实际产生（Vc->Va,func）的信息流。

【补充 2：语义信息流法如何区分真实隐蔽信道与伪隐蔽信道】

```
1.  func()
2.  {
3.      /*……*/          //其他代码
4.      if(Va>0)
5.         Vb = Vc;     //赋值语句(1)
6.      else
7.         Vd = Vb;     //赋值语句(2)
8.  }
```

单独来看，(1)处产生了(Vc->Vb,func)的信息流，(2) 处产生了 (Vb->Vd,func) 的信息流。如果忽略两组信息流的互斥性，就会产生(Vc->Vb->Vd,func)的信息流动，而这样的信息流动实际是不会产生真实隐蔽信道的。

4 隐蔽信道的案例

各种应用中的隐蔽信道非常多样，万物皆可成为隐蔽信道，在此仅列举一些比较典型的隐蔽信道。

隐蔽信道一般通信模型

在安全操作系统中，在公共信息流中利用公共信道中的合法信息流和未认证的信息流把未授权的秘密信息仍然可以由高级用户转向低级用户。[8]

阈下信道：一个在看守完全监控下两个囚犯如何协商逃跑计划的例子。Alice 和 Bob 两人因合伙从事犯罪活动二被捕，分贝关在两个相距很远的囚室里，他们企图逃跑，为制定粤语计划他们必须进行协商，但是他们唯一的通信方式是通过看守 Wendy 来传递信息。两个囚犯所传递的消息对 Wendy 来说是完全公开的，若他发现或察觉到两个囚犯传递了违法信息，Alice 和 Bob 将会被隔离，他们的越狱也就以实拍高中。而为了协商越狱计划，Alice 和 Bob 之间要传递合法消息，并通过某种方式包含了 Wendy 不能发现的隐蔽通道。(考试作弊暗号也是隐蔽信道的一种实现方式)

网络中的隐蔽信道

1) FTP

基于命令的 FTP 隐蔽信道：

利用 NOOP、ABOR 和 ALLO 三个命令一个作为起始结束标志，另外 2 个命令以各种方式代表数据，以此方式构筑隐蔽信道转递信息。此方式隐蔽性和鲁棒性很高。

基于目录编码的 FTP 隐蔽信道：

通过将文件目录编码，首先发送一个 ABOR 命令表示通信开始，接着将隐蔽信息分割，每次取 N 位对照编码表，找到对应的目录名称，发送该目录的 CWD 命令 (FTP 切换目录的指令)。此方式鲁棒性高，但在传递的数据量较大时隐蔽性降低，频繁的 CWD 指令会引起怀疑。

目录	编码
Demo	00
Movie	01
MP3	10
Pic	11

……

2) 数据库

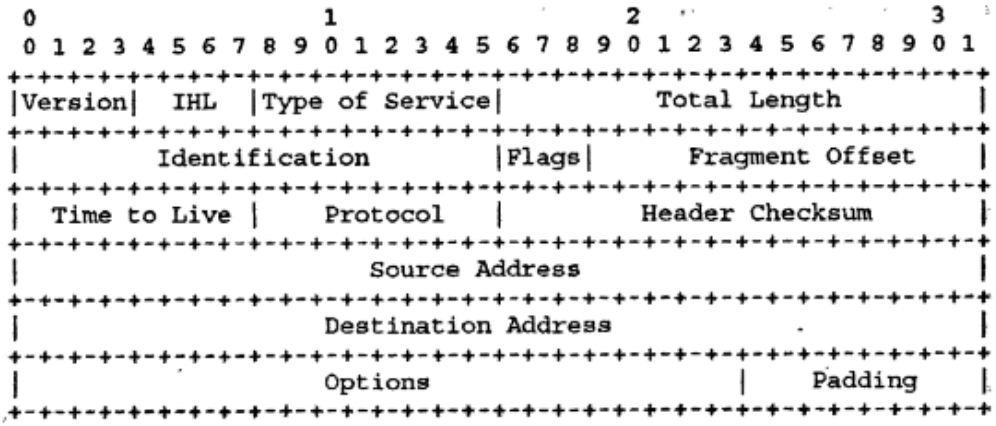
（1)利用不同安全级的并发冲突构造隐蔽信道,称作数据冲突隐蔽信道(data conflict covert channel，DC 信道）

（2）发送者修改数据或数据字典，接收者通过完整性约束等方式简介感知数据/数据字典的修改，从而获得信息。

……

3) 协议

利用 HTTP、TCP、ICMP、ipv4、ipv6 协议的头域、报头中的各个字段来构建隐蔽信道。例如下图 ipv4 协议的头域中，很多字段如 Identification、Type of Service 等等好几个字段都可以构建隐蔽信道。



基于 HTTP 协议有很多中隐蔽信道实现方法。HTTP 协议语法定义较为宽松，存在很多冗余信息，可以用了嵌入隐蔽信息，构建隐蔽信道，例如利用 Cookie、URL 等。

利用物理层中的冲突检测和重传机制构造隐蔽信道。物理层中当两个数据包发生冲突时，双方将各自收到一个冲突信号，之后将延迟将一段随机时间然后重新发送数据包。Alice 通过碰撞网络中的数据包,并根据要传输的隐蔽信息早于或晚于先前碰撞过的数据包来重发数据包，Bob 通过观察 Alice 重发的数据包是早于还是晚于之前碰撞过的数据包就能编码出相应的隐蔽信息。[9]

5. 总结

从诞生以来，隐蔽信道的研究一直持续推进，从程序到单机、从单机到网络，隐蔽信道在各个时期、各种领域都找到了自己的用武之地。目前随着互联网的快速发展，万物互联互通的大背景之下，网络隐蔽信道已经成为当前信息安全的热点研究领域。无论是从网络模型的理论分析，还是 TCP/IP 协议中的网络隐蔽信道实例，不可否认网络隐蔽信道已经开始威胁计算机系统的安全。而且,相比最初的单机隐蔽信道,网络隐蔽信道更容易受到网络延迟、网络抖动等噪音的影响，具有更高的复杂性和不易侦测性。

相信在未来的一段时期内，隐蔽信道的研究将会着眼于网络隐蔽信道的编码、传输和解码技术以及同步机制。毋庸置疑，随着互联网、计算机领域的迅速发展，隐蔽信道发展的速度也会飞快地增长，一路高歌猛进。

6. 参考文献

[1]钱玉文,赵邦信,孔建寿,王执铨.一种基于 Web 的可靠网络隐蔽时间信道的研究[J].计算机研究与发展,2011,48(03):423-431.

[2]王永吉,吴敬征,曾海涛,丁丽萍,廖晓锋.隐蔽信道研究[J].软件学报,2010,21(09):2262-2288.

[3]吴其祥. 网络中的隐蔽信道[D].西安电子科技大学,2009.

[4]翟江涛. 网络通信的信息隐藏技术研究[D].南京理工大学,2008.

[5]邹昕光.基于 FTP 协议的命令序列隐蔽信道[J].哈尔滨工业大学学报,2007(03):424-426.

[6]崔宾阁,刘大昕.基于信息流图的隐通道分析技术研究[D].哈尔滨工程大学.

[7]孔德兰..基于信息流的隐通道自动研究[D].西安电子科技大学.

[8]邹昕光,金海军,郝克成,孙圣和.基于 HTTP 协议的参数排序通信隐藏算法[J].计算机工程,2006(20):147-149.

[9]柏森. 基于信息隐藏的隐蔽通信技术研究[D].重庆大学,2002.