一种基于 Web 访问模型的网络隐蔽通道①

廖晓锋[1,2], 邱桂华[3]

[1](南昌大学 信息工程学院, 南昌 330029)
[2](中国科学院软件研究所 基础软件中心, 北京 100190)
[3](江西科技学院 信息学院, 南昌 330029)

摘 要: 网络隐蔽信道是将窃取的机密信息隐藏在正常的网络传输协议中的一种通信方法. 由于网络时间隐蔽信道不修改网络数据包的内容, 因此更加难以检测和限制, 从而具有更大的威胁. 提出一种新的基于 Web 访问模型的网络时间隐蔽信道, 恶意用户通过规律性的访问 Web 服务器实现机密信息传输; 实现了该网络隐蔽信道原型, 并给出了信道的性能分析结果.

关键词: 隐蔽信道; web 访问模型; 容量编码机制; 同步机制

A New Network Covert Channel Based on Web Access Model

LIAO Xiao-Feng[1,2], QIU Gui-Hua[3]

[1](Information Engineering School, Nanchang University, Nanchang 330029, China)
[2](National Fundamental Software Center, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)
[3](Computer Science Department, JiangXi University of Technology, Nangchang 330098, China)

Abstract: Network covert channel is a transmission scheme which hides the confidential information to normal network channel. Network covert timing channel does not modify the network packets, therefore it is more difficult to detect and more dangerous. This paper presents a new network covert timing channel based on Web access model. Malicious users transfer the confidential information by regularly access the Web server in this channel. We implement the prototype of the covert channel, and analyze the channel performance.

Key words: covert channel; web access model; capacity encoding; synchronization

1973 年 Lampson 在程序限制问题中首次提出隐蔽信道的概念, 并指出恶意用户可以利用隐蔽信道窃取安全操作系统中的机密信息[1]. 从此, 隐蔽信道成为信息安全领域的热点问题, 其研究范畴从操作系统延伸到数据库系统、网络系统和最新的云计算系统[2]. 其中, 网络隐蔽信道能够带来系统间的信息泄漏, 危害最为严重, 因此国内外的安全标准都将隐蔽信道分析列为必须执行的硬性指标.

网络隐蔽信道是将窃取的机密信息隐藏在正常的网络传输协议中的一种通信方法. 根据借助的传输媒介的不同, 网络隐蔽信道可以分为网络存储隐蔽信道和网络时间隐蔽信道. 网络存储隐蔽信道通常使用一些特殊的报文字段, 如预留字段、或篡改一些不影响报文网络传输的字段, 作为携带机密信息的媒介[3,4]; 网络时间隐蔽信道通常使用网络数据包的发送时间特性、到达时间特性以及时间间隔特性来作为携带机密信息的媒介[5]. 由于网络时间隐蔽信道不修改网络数据包的内容, 因此更加难以检测和限制, 从而具有更大的威胁[6,7].

本文提出一种新的基于 Web 访问模型的网络时间隐蔽信道, 恶意用户通过规律性地访问 Web 服务器实现机密信息传输; 本文实现了隐蔽信道原型, 并给出了该信道的性能分析结果.

1 相关工作

1.1 隐蔽信道通信模型

Simmons 通过研究监狱中两个囚犯秘密协商逃跑计划的例子, 提出用监狱模型(The Prisoner Problem)来建模隐蔽信道[8]. 如图 1 所示, 囚犯 Alice 和 Bob 在

① 收稿时间:2012-06-20;收到修改稿时间:2012-09-07

看守 Wendy 的监视下密谋越狱. Wendy 允许 Alice 和 Bob 之间进行通信, 但是通信内容必须经过其审查. 一旦 Wendy 从通信内容中发现可疑的信息, 立即把 Alice 和 Bob 完全隔离开, 使其不可能越狱. 在这种情况下, Alice 和 Bob 只能将协商越狱的机密信息隐藏到不会引起 Wendy 注意的公开传递的信息之中.

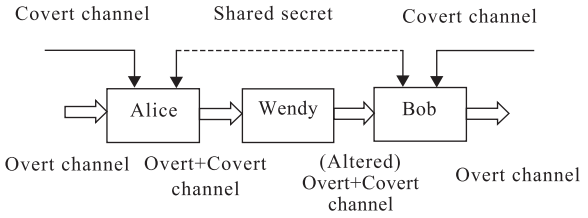

图 1 用监狱模型建模隐蔽信道

囚犯模型准确地描述了隐蔽信道的四个基本特征, 包括机密信息、编解码机制、公开信道、同步机制. 其中, 机密信息是 Alice 和 Bob 协商越狱的内容, 是被隐藏的内容; Alice 和 Bob 采用预先约定的编解码机制将机密信息隐藏到公开信道中通过 Wendy 传递; Wendy 审查后忠实地传递信息, 实现了信道的自同步机制.

Handel 将囚犯模型扩展到网络场景中, 描述了网络隐蔽信道模型[9]. 在网络中, Alice 和 Bob 表示能够相互通信的两台主机, Wendy 是网络中的安全监控设备, 如防火墙系统, 入侵检测系统等. Alice 和 Bob 将要传输的机密信息编码为网络数据包的特征(包括特殊字段、数据包时间特征等), 通过正常的网络通信附带传输出去, 从而实现网络隐蔽信道.

1.2 网络隐蔽信道实例

网络隐蔽信道危害严重, 能够逃避网络监控系统的限制, 即使一个数据包泄漏 1bit 信息, 一个大型站点每年都会泄漏 26GB 的信息. 网络隐蔽信道的实现方式多样, 其中最经典的是 Cabuk 提出的网络时间隐蔽信道(IP Covert Timing Channel)[10].

如图 2 所示, 在 Cabuk 提出的网络隐蔽信道中, 机密信息的发送方和接收方约定固定的时间间隔(Timing Interval)作为传输周期, 构成了自同步机制. 在每个时间段内, 发送方向接收方发送一个数据包, 或者不发送数据包, 这两种动作分别代表机密信息编码后的“0”和“1”, 从而构成二元隐蔽信道. Cabuk 的信道依赖数据包的时间特征, 因此属于时间隐蔽信道. 图 2 中, 发送方在 8 个时间间隔内发送了 4 个数据包, 传递了 8bit 的机密信息; 接收方按照约定解码就能够得到该机密信息.

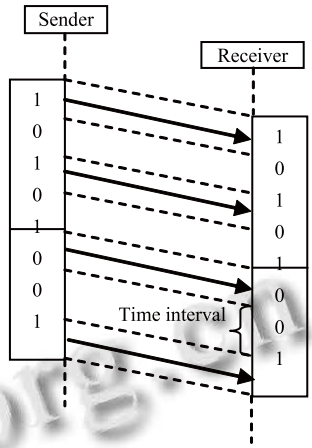

图 2 Cabuk 提出的网络时间隐蔽信道

2 基于Web访问模型的隐蔽信道设计

2.1 Web 访问模型

Web 服务基于应用层的 HTTP 协议提供网络信息浏览功能, 是目前最流行的网络服务[11]. HTTP 协议使用 POST 和 GET 等命令来实现网络浏览. POST 命令被设计用来向服务器上传数据, GET 命令被设计用来从服务器下载 URL 地址. 通常的 Web 服务采用客户机/服务器的工作模式, 具体的工作流程如下:

(1) 在客户端, 用户使用浏览器或其他程序与服务器建立连接, 并发送访问请求;

(2) Web 服务器接收到请求后, 将响应信息发回客户端;

(3) 一次请求通信完成, 关闭连接.

基于 Web 访问模型, 本文提出一种全新的隐蔽时间信道, 其架构如图 3 所示. 其中, Web 服务器位于开放网络环境中, 发送方能够通过公开信道访问 Web 服务器资源, 且网络公开信道中存在防火墙等安全防护设备, 能够防御一般的入侵行为. 接收方位于公开信道中, 能够监控发送方的网络数据包行为, 但是无法获得数据包的具体内容.

在图 3 的基于 Web 访问模型的隐蔽信道中, 收发双方预先约定编解码机制和同步机制. 利用编解码机制, 发送方将要发送的信息编码成不同的时间段, 插入到访问 Web 服务器的时间间隔中. 接收方观察发送方的访问模型, 计算同步周期开始之后的数据包时间特征, 通过得到的数据包到达时间计算数据包间隔, 再使用解码算法逆向解出传输的机密信息.

与传统的网络存储隐蔽信道不同, 该信道在机密信息的传输过程中, 不修改数据包的内容, 只影响数据包的时间特征. 正常用户的 Web 访问操作时间序列符合随机化模型, 而由于机密信息编码后的时间段的嵌入, 该信道的访问模型随机性减弱. 在海量的 Web 访问记录中, 该信道访问模型的微小变化难以察觉和检测; 同时, 到目前为止, 还没有一款专用的网络设备能够根据访问模型检测隐蔽信道. 因此, 基于 Web 访问模型的隐蔽信道能够在公开信道中传输机密信息, 具有更大的安全威胁.

图 3 基于 Web 访问模型的隐蔽信道结构图

2.2 隐蔽信道设计

2.2.1 编码机制设计

通常的网络时间隐蔽信道采用 ASCII 码(American Standard Code for Information Interchange)的编码方式, 将机密信息编码成二元序列, 构成二元隐蔽信道. 在时间间隔能够表示的情况下, 还可以采用更复杂的基于编码表的编码方法, 收发双方通过查表编解码机密信息, 提高隐蔽信道的性能.

本文设计的隐蔽信道采用 ASCII 的编码方式, 但是可以方便地扩展为多元编码. ASCII 是计算机系统中通用的单字节编码系统, 主要用于显示现代英语和其他西欧语言. ASCII 码中每个字符使用指定的 8 位二进制数组合来表示, 因此 ASCII 用 8bits 的二进制字符串表示一个字符(由符号“0”和“1”).

假设待传输的机密信息为 secret, 发送方使用 ASCII 编码得到 secret_ascii, 编码函数为 encode_ASCII, 该过程可以表示为 secret_ascii←encode_ASCII(secret); 接收方使用 ASCII 解码是编码的逆过程, 解码函数为 decode_ASCII, 该过程可以表示为 secret←decode_ ASCII (secret_ascii).

2.2.2 机密信息载体设计

在基于 Web 访问模型的隐蔽信道中, 机密信息的载体是 HTTP 协议数据包的时间间隔. 根据编码机制, 收发双方预先约定使用不同的时间间隔 T_0 和 T_1 表示编码后的符号“0”和“1”, 时间间隔 T_0 和 T_1 之间满足 $T_0 < T_1$ 关系.

假设正常的 Web 访问时间间隔为 normal_intervals, 表示为 normal_intervals= $\{t_1, t_2, \cdots, t_n\}$. 发送方根据编码后的机密信息 secret_ascii 计算得到的时间段为 send_intervals, 该过程表示为 send_intervals← secret_mapto_intervals(secret_ascii), 映射后的表示为 send_intervals= $\{\Delta t_1, \Delta t_2, \cdots, \Delta t_n\}$. 隐蔽信道通信过程中, 将 send_intervals 插入到 normal_intervals 的过程表示为 add_intervals ← insert_intervlas(send_intervlas, normal_ interlvals). 最终, 发送方按照 add_intervals 间隔访问 Web 服务器.

接收方观察发送方的 Web 访问模型, 截获到的时间间隔为 received_intervals, 表示为 received_intervals = $\{\Delta R_1, \Delta R_2, \cdots, \Delta R_n\}$. 由于 ΔR_i 和 ΔR_j 之间存在 $\Delta R_i > \Delta R_j$ 或 $\Delta R_i < \Delta R_j$ 的关系, 因此可以逆向映射为机密字符串 secret_ascii ← intervals _mapto_ secret (received_intervals).

2.2.3 同步机制设计

一个完整的信息传输周期包括发送者/接收者同步阶段、信息传输阶段和反馈阶段. 同步的目的是为了确定译码的边界, 保证通信有效地进行. 同步阶段中, 发送方通知接收方准备接收信息; 当信道传输结束后, 发送方通知接收方传输完毕[12].

本文提出一种简单易行的同步机制, 称为长延迟同步机制. 发送方在传输开始和结束都是用一个较长的延迟作为标志, 向接收方发送通知. 例如, 同步长延迟为 T_s, 且远大于 T_0 和 T_1, 表示为 $T_0 < T_1 < T_s$, 使用 syn_start(Ts)和 syn_stop(Ts)表示传输过程的开始和结束.

3 实验分析

为了验证本文提出的隐蔽信道, 实验采用 Python 编程语言实现了基于 Web 访问模型的网络时间隐蔽信道.

实验中使用四台机器, 其中机器 A 为信道发送方, 机器 B 为接收方, 机器 C 运行 Web 服务器, 机器 D 运行防火墙系统. 机器 B 和 C 处在同一局域网中, C 使用 tcpdump 采集机器 A 访问 Web 服务器的数据包时间特征, 具体的原型实现如下文所示.

3.1 信道传输实现

发送方首先编码传输的机密信息，然后按照时间间隔访问 Web 服务器，实验的伪代码如下:

```
#预处理待发送的机密信息
secret_ascii ← encode_ASCII(secret)
send_intervals ← secret_mapto_intervals(secret_ascii)
add_intervals ← insert_intervlas(send_intervlas, normal_interlvals)
#开启同步周期
syn_start(Ts)
#打开 socket 连接，准备发起 Web 服务请求
socketserver.ThreadingTCPServer((C, 80), WebServer)
#根据修改后的时间间隔访问 Web 服务器
for each interval in add_intervals do
    sleep(interval)
    sendHttpRequest(C)
end do
#发送完成，停止传输周期
syn_stop(Ts)
#传输完毕，关闭 socket 连接
socketserver.close()
```

接收方监控发送方的数据包时间特征，并最终从该时间特征中解码出通过隐蔽信道传输的机密信息，实验的伪代码如下:

```
#使用 tcpdump 采集发送方访问 Web 服务器的时间特征
tcpdump A
#计算一个传输周期之内的数据包时间间隔
received_intervals ← getIntervals(syn_start, syn_stop)
#逆向映射时间间隔
secret_ascii←intervals_mapto_secret(received_intervals)
#接收方解码得到的机密信息
secret ← decode_ASCII(secret_ascii)
```

实验中，收发双方传输的机密信息字符串为“The quick brown fox jumps over the lazy dog”. 编码后的二进制字符串为“01010100 01101000 01100101 00100000 01110001 01110101 01101001 01100011 ……”，总长度为 344bit.

图 4 表示基于 Web 访问模型的网络时间隐蔽信道的完整的传输周期. 传输过程中开始时的 100 个数据包和结束的前 100 个数据包分别表示正常的网络传输，可知访问 Web 服务的网络数据间隔大约保持在 10～20ms. 第 100～110 和 455～465 个数据包是传输起始和结束同步阶段的数据包，产生了较大的时间间隔，大约在 40～60ms 之间. 第 111～454 个数据包是隐蔽信道使用的数据包，其时间间隔大于正常值，范围在 20～40ms 之间. 本文中，使用 20～30ms 范围的时间间隔表示传输编码后的符号“0”，使用 30～40ms 范围的时间间隔表示传输编码后的符号“1”.

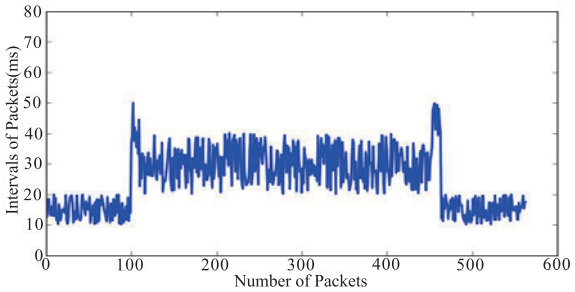

图 4 隐蔽信道传输过程中的数据包间隔分布情况

3.2 信道容量分析

通常使用信道容量指标分析隐蔽信道的性能. 信道容量是国内外安全标准普遍采用的隐蔽信道威胁度量指标，并且得到了广泛的实际应用[13,14].

信道容量是通过一个通信信道能够可靠传输的信息量的上限值，单位是 bit/s. 容量是信道的属性，其仅仅依赖于信道的传输特性. 信道容量是对信道最大传输能力的度量，信道实际传送的信息量必然不大于信道容量.

信道容量的计算方面，主要包括两种方法: 一种是理论分析，通过假设一个噪声模型来实现，其中最为著名的是 Millen 提出的形式化方法; 另一种是实验方法，即实现一个隐蔽信道而后计算其信息传输速率，其中最著名的是 Tsai 等人提出的非形式化方法[15].

本文采用实验验证的方法，采用如下的隐蔽信道容量计算公式:

$$C=\frac{N(t)}{t}bits/s$$

其中 $N(t)$表示 t 时间内传输的机密信息总比特数.

为了取得隐蔽信道容量的平均性能，实验将隐蔽信道传输过程重复执行 100 次. 图 5 中上部的蓝线表示这 100 次传输过程的平均容量指标.

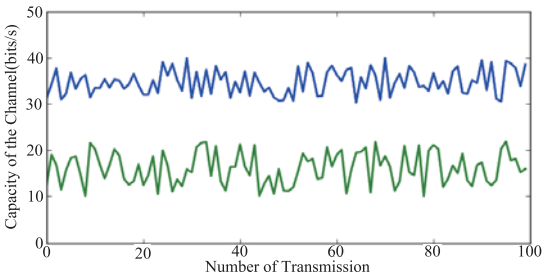

图 5 隐蔽信道传输过程中的容量指标

从中可见，由于网络环境的影响，每次隐蔽信道执行过程中的容量值都不相同，但是信道容量基本保持在

30～40bits/s 的范围内. 与 Cabuk 提出的隐蔽信道相比, 如图中绿线所示, 其容量只有 16.67bits/s, 本文提出的基于 Web 访问模型的隐蔽信道容量高于 Cabuk 的信道, 因此具有更好的性能.

3.3 信息传输错误率分析

除了使用安全标准中要求的容量指标分析信道的性能, 本实验也采用了 Cabuk 提出的错误率指标分析信道的性能. 隐蔽信道中传输的是机密信息, 如果传输错误率过高, 可能导致接收方无法解码机密信息, 导致传输过程无效. Cabuk 使用编辑距离(Edit Distance)计算出错的字符串比特数, 从而计算传输错误率. 其中, 编辑距离是指两个字串之间, 由一个转成另一个所需的最少编辑操作次数. 许可的编辑操作包括将一个字符替换成另一个字符, 插入一个字符, 删除一个字符.

图 6 表示了隐蔽信道执行 100 次的传输错误率. 受网络环境的影响, 每次隐蔽信道执行过程中的错误率略有差异, 但是总体低于 6%. 对于传输的 344bit 信息而言, 在 6%的错误率下, 接收方能够解码出传输的机密信息. 从而保证了隐蔽信道传输的有效性.

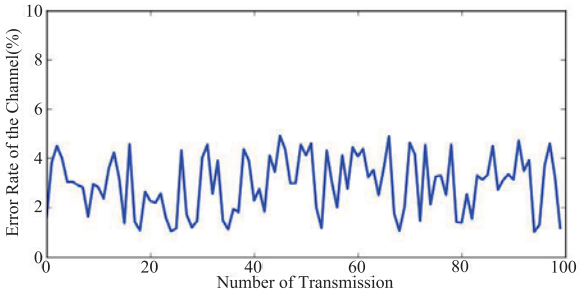

图 6 隐蔽信道传输过程中的错误率指标

4 结语

目前, 隐蔽通道是网络安全研究热点, 尤其是云计算平台的出现, 进一步促进了隐蔽信道的研究. 网络隐蔽信道由于其严重的危害性, 吸引了更多的研究人员的注意. 本文提出的网络时间隐蔽信道基于 Web 访问模型, 设计了隐蔽信道的核心机制, 包括编解码机制、信道载体以及同步机制. 与其他类型的隐蔽信道相比, 具有高效、难以检测的特点. 今后的研究将从安全防护的角度入手, 研究网络隐蔽通道的检测和限制方法.

参考文献

1 Lampson BW. A note on the confinement problem . Commun ACM, 1973,16(10):613–615.

2 吴敬征,丁丽萍,王永吉.云计算环境下隐蔽信道关键问题研究.通信学报,2011,32(9A):184–203.

3 邹昕光,孙圣和.基于 TCP 选项域的信息隐藏算法研究.计算机工程与设计,2006,27(12):2111–2145.

4 胡静,谢俊元.IPSec 协议中潜在的隐蔽信道问题研究.计算机工程与设计,2007,28(17):4116–4125.

5 Zander S, Armitage G, Branch P. A survey of covert channels and countermeasures in computer network protocols. Communications Surveys & Tutorials, IEEE, 2007, 9(3):44–57.

6 王湘渝,江文,唐俊.一种基于 NDIS 驱动程序实现隐蔽通道的方法.计算机应用与软件,2011,28(6):299–301.

7 邹昕光,金海军,郝克成,等.基于HTTP协议的参数排序通信隐藏算法.计算机工程,2006,32(20):147–149.

8 Simmons G. The prisoners' problem and the subliminal channel. Proc. of CRYPTO83-Advances in Cryptology, F, 1984.

9 Handel T, Sandford M. Hiding data in the OSI network model. Information Hiding.1996:23–38.

10 Cabuk S, Brodley C E, Shields C. IP covert timing channels:design and detection. Proc. of the 11th ACM Conference on Computer and Communications Security. Washington DC: ACM, 2004:178–187.

11 钱玉文,赵邦信,孔建寿,等.一种基于 Web 的可靠网络隐蔽时间信道的研究.计算机研究与发展,2011,48(3):423–431.

12 王永吉,吴敬征,曾海涛,等.隐蔽信道研究.软件学报,2010,21(9):2262–2288.

13 卿斯汉.高安全等级安全操作系统的隐蔽通道分析.软件学报,2004,15(12):1837–1849.

14 王宜菲,杨亚磊,饶孟良.基于 C4.5 的 HTTP 隧道检测技术研究.计算机工程与设计,2012,33(2):493–497.

15 卿斯汉,沈昌祥.高等级安全操作系统的设计.中国科学(E辑:信息科学),2007,37(2):238–253.