



信息安全概论

网络威胁

刘亚维

主要内容

The slide features decorative circles in the top header area. On the left, there is a solid light purple circle and an empty light purple circle outline. On the right, there are three circles: a solid light purple circle, an empty light purple circle outline, and another solid light purple circle.

- 6.1 概述
- 6.2 计算机病毒
- 6.3 网络入侵
- 6.4 诱骗类威胁

6.1 概述

- 威胁：用威力逼迫恫吓使人屈服。
- 网络威胁：是网络安全受到威胁、存在着危险。
- 随着互联网的不断发展，网络安全威胁也呈现了一种新的趋势，
 - 最初的病毒，比如“CIH”、“大麻”等传统病毒
 - 逐渐发展为包括特洛伊木马、后门程序、流氓软件、间谍软件、广告软件、网络钓鱼、垃圾邮件等等，
 - 目前的网络威胁往往是集多种特征于一体的混合型威胁。

网络威胁的三个阶段

- 第一阶段（1998年以前）
 - 网络威胁主要来源于传统的计算机病毒，其特征是通过媒介复制进行传染，以攻击破坏个人电脑为目的；
- 第二阶段（大致在1998年以后）
 - 网络威胁主要以蠕虫病毒和黑客攻击为主，其表现为蠕虫病毒通过网络大面积爆发及黑客攻击一些服务网站；
- 第三阶段（2005年以来）
 - 网络威胁多样化，多数以偷窃资料、控制利用主机等手段谋取经济利益为目的。

网络威胁分类

- 从攻击发起者的角度来看，
 - 主动攻击型威胁，如网络监听和黑客攻击等，这些威胁都是对方人为通过网络通信连接进行的；
 - 被动型威胁，一般是用户通过某种途径访问了不当的信息而受到的攻击。
- 依据攻击手段及破坏方式进行分类
 - 第一类
 - 传统病毒、蠕虫、木马等为代表的计算机病毒；
 - 第二类
 - 黑客攻击为代表的网络入侵；
 - 第三类
 - 间谍软件、广告软件、网络钓鱼软件为代表的欺骗类威胁。

6.2 计算机病毒

- 电脑病毒，或称计算机病毒。
 - 是一种在人为或非人为的情况下产生的、在用户不知情或未批准下，能自我复制或运行的电脑程序；
 - 电脑病毒往往会影响受感染电脑的正常运作，或是被控制而不自知，也有电脑正常运作仅盗窃数据等用户非自发引导的行为。

关于病毒作者

- 一公司的分析报告称：目前全世界现有**200万**有能力写较成熟电脑病毒的[程序员](#)。^[1]
- 有不少病毒制作者及[黑客](#)们被逮捕并予以起诉，判决的轻重各国都有所不同
 - [罗马尼亚西欧班尼](#)花费**15分钟**写的MSBlast.F变种大约只感染了**1000台**计算机，按他们国家的法律他就有可能最高会被判**15年**[有期徒刑](#)。
 - **1998年**[台湾](#)病毒作者[陈盈豪](#)写的[CIH](#)病毒据猜测造成全球**6000万台**计算机瘫痪，但他因为在被逮捕后无人起诉而免于法律制裁，在**2001年**有人以CIH受害者的身份起诉陈盈豪，才使他再次被逮捕，按照台湾当时的法律，他被判[损毁罪](#)面临最高**3年**以下的有期徒刑。
 - [中国](#)大陆的木马程序“[证券大盗](#)”作者[张勇](#)因使用其木马程序截获股民账户密码，盗卖[股票](#)价值**1141.9万元**，非法获利**38.6万元**[人民币](#)，被逮捕后以[盗窃罪](#)与[金融犯罪](#)起诉，最终的判决结果是[无期徒刑](#)。

历史

- 学术工作

- 1949年由约翰·冯·诺伊曼 "Theory of self-reproducing automata"。
 - 冯·诺伊曼在他的论文中描述一个计算机程序如何复制其自身。
- 1972年，Veith Risak "Self-reproducing automata with minimal information exchange"。
 - 该文描述一个以西门子4004/35计算机系统为目标，用汇编语言编写，具有完整功能的计算机病毒。
- 1980年，Jürgen Kraus 的学位论文"Self-reproduction of programs"。
 - 在他的论文中，他假设计算机程序可以表现出如同病毒般的行为。
- 1984年弗雷德·科恩（Fred Cohen）的论文《电脑病毒实验》。
 - “病毒”一词最早用来表达此意

历史

● 科幻小说

- 病毒一词广为人知是得力于[科幻小说](#)。
- 1970年代中期大卫·杰洛德（David Gerrold）的《When H.A.R.L.I.E. was One》
 - 描述了一个叫“病毒”的[程序](#)和与之对战的叫“[抗体](#)”的程序；
- 1975年约翰·布鲁勒尔（John Brunner）的小说《震荡波骑士（ShakewaveRider）》
 - 描述了一个叫做“磁带蠕虫”、在[网络](#)上删除数据的程序。[\[2\]](#)

历史

● 病毒程序

- 1960年代初，[美国麻省理工学院](#)的一些青年研究人员，在做完工作后，利用业余时间玩一种他们自己创造的[计算机游戏](#)。
- 做法是某个人编制一段小程序，然后输入到计算机中运行，并销毁对方的游戏程序。
- 这可能就是计算机病毒的雏形。

知名病毒及蠕虫的历史年表

年份	日期	事件
1980年		Jürgen Kraus撰写硕士论文： 自我复制的程序 （Selbstreproduktion bei programmen）。
1982年		一个运行于 Apple II 系统，称为 Elk Cloner 病毒的电脑病毒出现，据信是第一个非实验室制造的电脑病毒。
		Joe Dellinger在A+M大学撰写了一个Apple II电脑病毒。
1986年	1月	(c)Brain 开机扇区病毒出现，可认为是第一个个人电脑病毒。此病毒也称为拉合尔（Lahore）、巴基斯坦（Pakistani）或巴基斯坦大脑，因为它是在巴基斯坦的拉合尔诞生的。
1987年	10月	耶路撒冷病毒 在 耶路撒冷 市被发现，它是一个项目在十三号星期五引导并毁灭所有可执行文件的病毒。
	11月	以 Amiga 为对象，称为 SCA病毒 的 开机扇区 病毒出现，立刻造成病毒作者间的风暴。而不久之后瑞士鬼客联合（SCA）马上发布另一个被认为更具破坏力的病毒： 比特强盗 。
1988年	6月	Festering Hate病毒 与 ProDOS病毒 从私有 BBS 散布到主要网络上。
	11月2日	由 罗伯特·泰潘·莫里斯 创造的 莫里斯蠕虫 （或称I.Worm），可感染 DEC VAX 与 SUN 以 BSD UNIX 运行的 网络 机器，成为了第一只荒野诞生（非实验室制造）的蠕虫，且是第一个利用 缓存溢出 漏洞的恶意程序。而 乒乓病毒 是本时代第一个知名的开机扇区感染病毒。
1989年	10月	第一只 跨领域病毒 ： 鬼球病毒 被 Fridrik Skulason 发现。

知名病毒及蠕虫的历史年表

年份	日期	事件
1992年	3月6日	米开朗基罗病毒 制造了一次数字大灾难。
1995年		第一个 宏病毒Word concept病毒 诞生。
1998年	4月26日	第一版 CIH病毒 v1.0问世。
1999年	3月26日	Melissa蠕虫 出现，攻击 Microsoft Word 的全局模板Normal.dot，所有新创建的文件都被感染，瞄准了使用 Microsoft Word 与 Outlook 的系统，病毒感染 Microsoft Outlook 通讯录上的前50个用户，并造成大量网络流量以及拥塞，本日全球许多大企业的邮件服务器因而停止运作一天。
	4月26日	CIH病毒 在全世界各地爆发。
	6月6日	可摧毁 Microsoft Office 文件的 ExploreZip蠕虫 首次被发现。

年份	日期	事件
2000年	5月	ILOVEYOU 病毒，或称VBS/Loveletter现身。并造成如同2004年一般可怕的商业损失，大约有100亿美元。
2001年	1月	一个类似 Morris蠕虫 ，称做 Ramen蠕虫 袭击安装 Red Hat Linux 6.2或7版的机器，使用了三个在 wu-ftpd 、 rpc-statd 与 lpd 的漏洞。
	5月8日	悲伤心情蠕虫 借由一个在 SUN 上 Solaris （ 信息安全告示区00191 ）与 Microsoft 上 IIS （ MS00-078 ）的漏洞而传播。
	7月	Sircam蠕虫 发布，借由电子邮件以及未保护的 网络分享 传播。
	7月13日	红色警戒蠕虫 发布，此蠕虫攻击Microsoft IIS的Index Server ISAPI Extension的漏洞（描述在 MS01-033 ），且造成非常严重的商业损失。
	8月4日	一个完全重新撰写的 红色警戒蠕虫 ，称为 红色警戒蠕虫II 被恶意发布，来源据说在中国。
	9月18日	Nimda蠕虫 被发现，并借由许多方式传播，传播方式叙述于 MS01-044 ，且利用了 红色警戒蠕虫II 与 悲伤心情蠕虫 留下的后门。
	10月26日	Klez蠕虫 第一次被发现且辨别。
2003年	1月24日	SQL slammer蠕虫 ，也称为 蓝宝石蠕虫 ，攻击了 Microsoft SQL Server 与 MSDE 的漏洞，详情公布于 MSDE described in MS02-039 与 MS02-061 ，造成了互联网上的广泛的灾情。
	8月12日	冲击波蠕虫 ，又称为 Lovesan蠕虫 ，借由 Microsoft Windows 在 MS03-026 第一次与稍后于 MS03-039 描述的漏洞进行攻击。
	8月18日	Welchia蠕虫 （假好心蠕虫，或 Nachi蠕虫 ）出现，此蠕虫试图移除冲击波蠕虫并修复Windows。
	8月19日	太上蠕虫 （ Sobig蠕虫 ）借由电子邮件与网络分享快速传播。
	10月24日	清醒蠕虫 （ Sober蠕虫 ）第一次出现，并且以多种变种维持到2005年。
本年冲击波与太上蠕虫的同时攻击造成非常大的伤害。		
2004年	1月下旬	世界末日蠕虫 （ MyDoom蠕虫 ）现身，并创下最快散播速度的邮件病毒记录。
	3月19日	Witty蠕虫 是一只打破多项纪录的蠕虫。它感染 ISS 产品的多项漏洞。它是最快散布的蠕虫，也是第一个携带毁灭性代码的蠕虫
	5月1日	震荡波蠕虫 借由一个叙述于 MS04-011 的 LSASS 漏洞现身，造成 互联网 瘫痪，甚至影响商业活动。
	12月	Santy蠕虫 ，据信是第一只 网页蠕虫 。它借由一个叙述于 BID10701 的PhpBB漏洞以及使用 Google 以发现新目标。它在 Google 过滤并防止此蠕虫散布之前已感染了40000个网站。
2005年	8月16日	利用叙述于 MS05-039 漏洞的 Zotob蠕虫 与数个变种的 恶意软件 被发现。它的影响非常轰动，因为数个 美国 媒体出口网站被感染了。
	10月13日	Samy病毒 成为2006年最快散布的 电脑病毒 。
2006年	1月20日	Nyxem蠕虫 出现，它散布大量电子邮件，且在 2月3日 后，试图在每月的三号发作，使信息安全相关软件与文件分享机制失效，并摧毁某类型的文件，例如 Microsoft Office 文件。
	6月28日	研究者指出 Farid Essebar 可能创作了数量高于20的病毒，包括 MyDoom病毒 的变种： MyDoom-BG病毒 、 Zotob蠕虫 演变的 Mytob蠕虫 ^[1] 。
2010年	6月	震网蠕虫 被发现，此蠕虫专门针对伊朗核设施的电脑硬件。

罗伯特·泰潘·莫里斯（英语：Robert Tappan Morris，1965年—）

- 网络代号rtm，美国程式员、计算机科学家与企业家。
1988年了创造互联网上的第一只电脑蠕虫程式，散布在互联网后，造成多个电脑系统瘫痪，因此被定罪。
 - 他曾创办Viaweb，也是Y Combinator的共同创办人之一。
- 生平
 - 他的父亲罗伯特·莫里斯（Robert Morris）为贝尔实验室计算机安全专家
 - 他16岁上初中时，发现UNIX系统漏洞
 - 1983年考入哈佛大学，拿到了文学学士（A.B.）学位
 - 1988年散布了“莫里斯蠕虫”，造成约6000个系统瘫痪，给这些用户总共带来约200万到6000万美元的损失。
 - 1988年成为康奈尔大学研究生
 - 现为麻省理工学院教授

计算机病毒定义

- 《中华人民共和国计算机信息系统安全保护条例》中明确定义：
 - 病毒是指“编制或者在计算机程序中插入的破坏计算机功能或者破坏数据，影响计算机使用并且能够自我复制的一组计算机指令或者程序代码”。

- 计算机病毒特征

- (1) 非授权性
- (2) 寄生性
- (3) 传染性
- (4) 潜伏性
- (5) 破坏性
- (6) 触发性

- 计算机病毒发展新的趋势

- ① 无国界
- ② 多样化
- ③ 破坏性更强
- ④ 智能化
- ⑤ 更加隐蔽化

- 计算机病毒可以根据其工作原理和传播方式划分成

- ① 传统病毒
- ② 蠕虫病毒
- ③ 木马

6.2.2 传统病毒

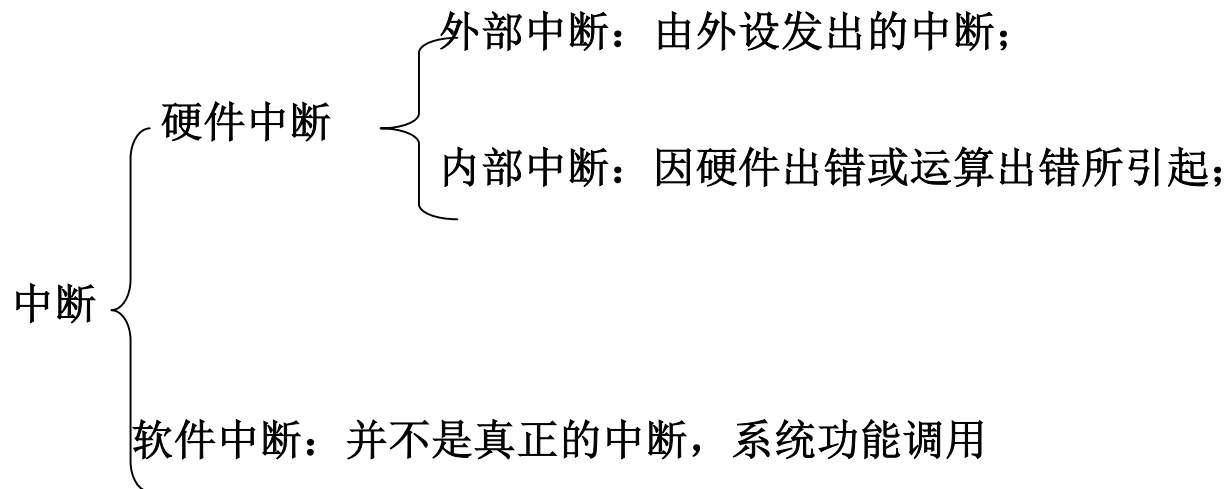
- 传统病毒的代表
 - 巴基斯坦智囊（Brain）、大麻、磁盘杀手（DISK KILLER）、CIH等。
- 传统病毒一般有三个主要模块组成，包括启动模块、传染模块和破坏模块。

传统计算机病毒的作用机制

- (1) 引导机制
- (2) 传染机制
- (3) 破坏机制

(1) 中断与计算机病毒

中断是**CPU**处理外部突发事件的一个重要技术。中断类型可划分为：

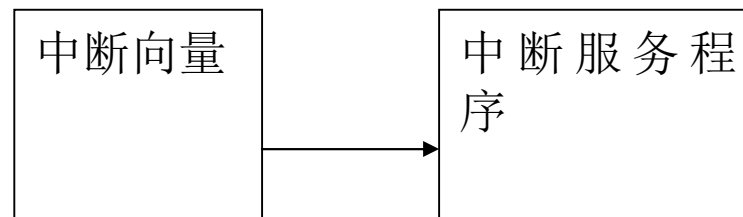


病毒有关的重要中断

- INT 08H和INT 1CH的定时中断，有些病毒用来判断激发条件；
- INT 09H键盘输入中断，病毒用于监视用户击键情况；
- INT 10H屏幕输入输出，一些病毒用于在屏幕上显示信息来表现自己；
- INT 13H磁盘输入输出中断，引导型病毒用于传染病毒和格式化磁盘；
- INT 21H DOS功能调用，绝大多数文件型病毒修改该中断。

病毒利用中断

盗用前:



盗用后:

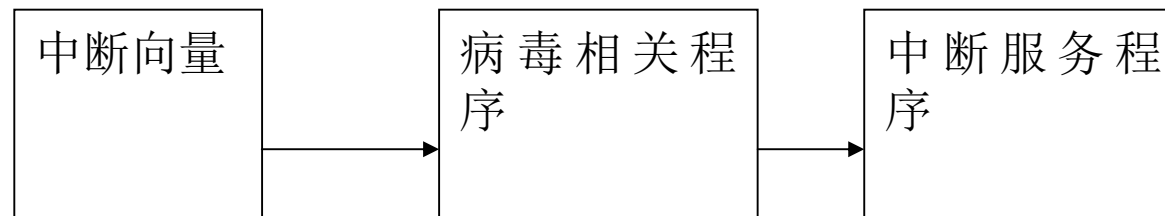


图 8-2 病毒盗用中断示意图

(2) 计算机病毒的传染机制

- 传染是指计算机病毒由一个载体传播到另一载体，由一个系统进入另一个系统的过程。计算机病毒的传染方式主要有：
 - 病毒程序利用操作系统的引导机制或加载机制进入内存；
 - 从内存的病毒传染新的存储介质或程序文件是利用操作系统的读写磁盘的中断或加载机制来实现的。

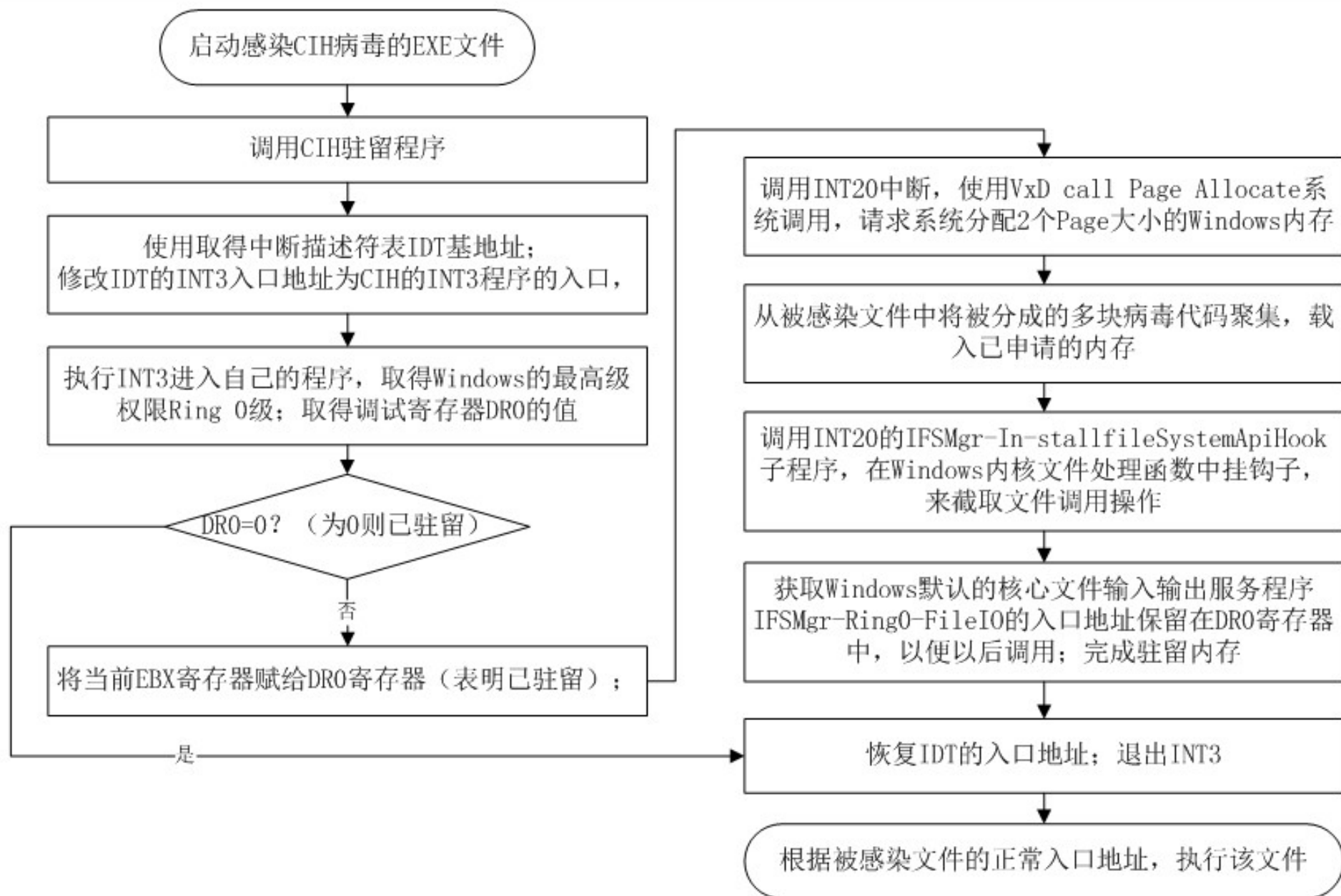
(3) 计算机病毒的破坏机制

- 破坏机制在设计原则、工作原理上与传染机制基体相同。它也是通过修改某一中断向量入口地址，使该中断向量指向病毒程序的破坏模块。

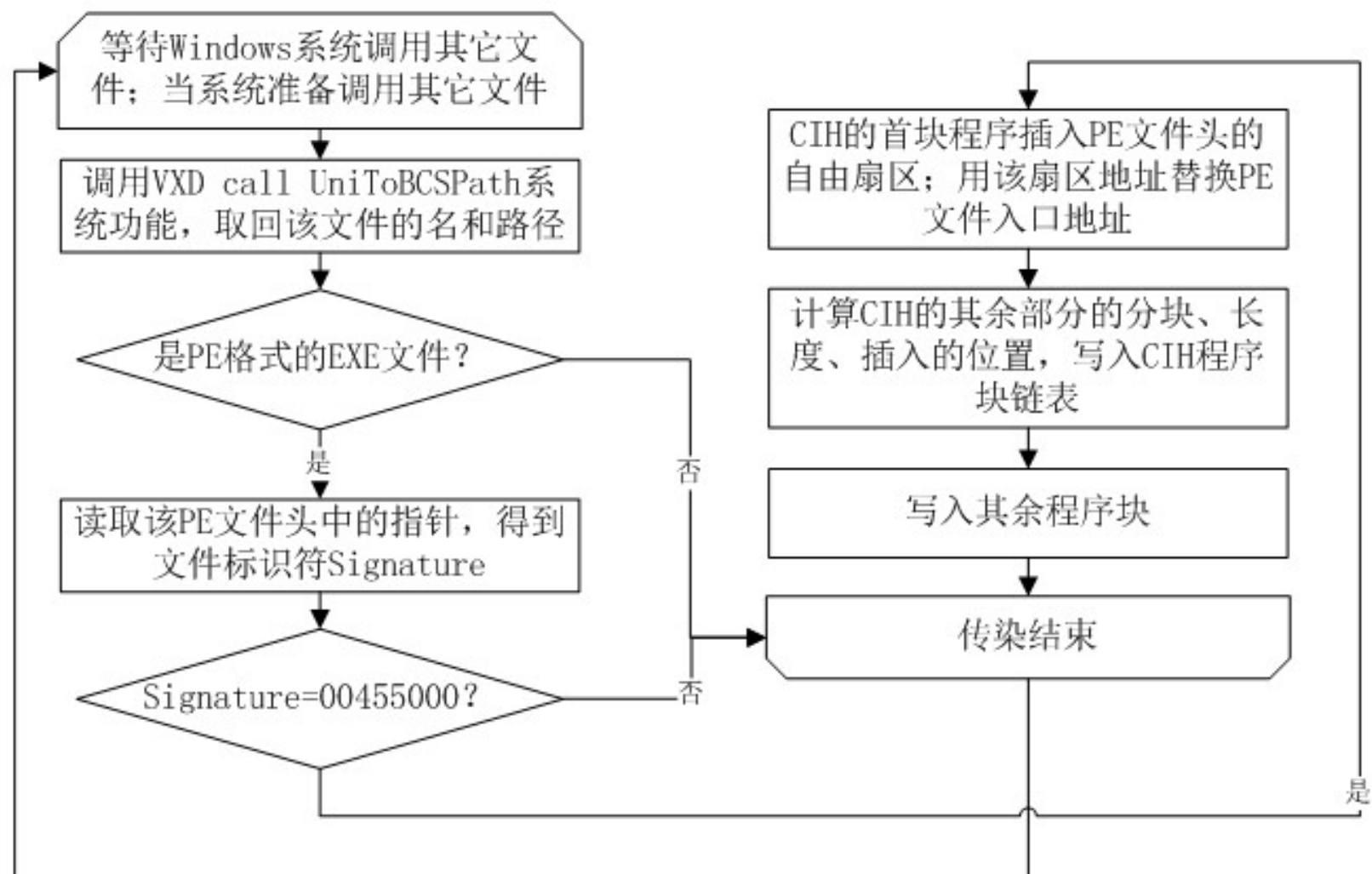
CIH

- 感染Windows95/98环境下PE格式的EXE文件（第一例）
- 病毒发作时直接攻击和破坏计算机硬件系统。
- 该病毒通过文件复制进行传播。
- 计算机开机后，运行了带病毒的文件，其病毒就驻留在Wnidows核心内存里，
- 组成：初始化驻留模块、传染模块和破坏模块。

驻留初始化模块



传染模块



破坏模块

从系统CMOS中取出当前日期DATA

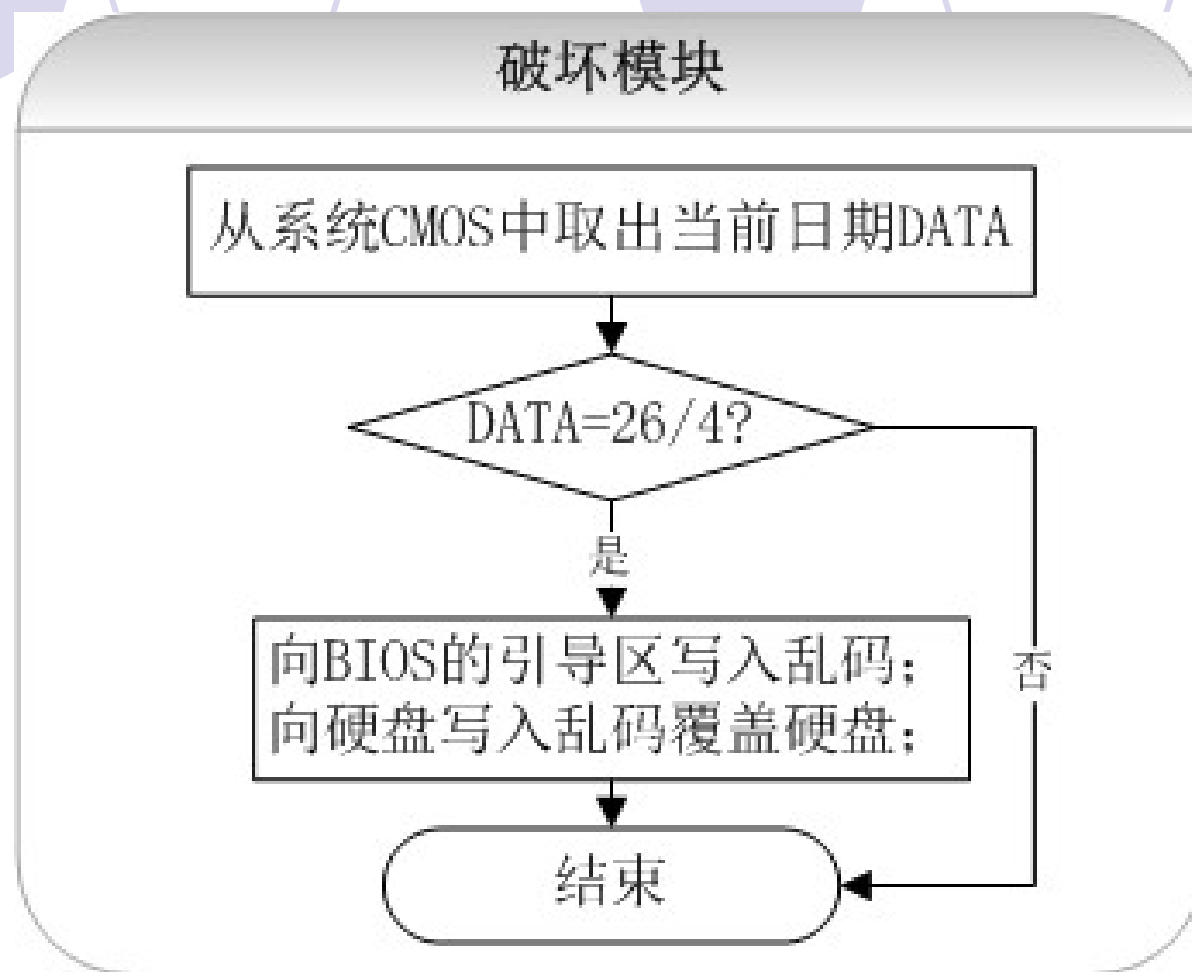
DATA=26/4?

是

向BIOS的引导区写入乱码：
向硬盘写入乱码覆盖硬盘：

否

结束



陈盈豪（1975年8月25日—）

- 台湾的电脑技术人员，[CIH病毒](#)始创人。
- [CIH](#)是他高中时，自己取的英文名字，也就是陈盈豪中文名字的缩写。从大学至工作，所有人都是以[CIH](#)称呼他。
- 1998年5月间，陈盈豪就读[大同大学](#)资讯工程学系二年级时，为能验证[杀毒软件](#)号称百分百防毒是不实广告，自行实验制作[CIH病毒](#)，在他不知情的状况下，他的同学使用了实验用电脑，而将此病毒携出，由于病毒体积小，会自行改变程式码分布，潜藏在档案未使用的空白区，档案大小不会改变，因此不易被察觉而大量被散布，1999年4月26日、2000年4月26日发作，造成全球电脑严重的损害[\[1\]\[2\]](#)。
- 1999年4月30日，CIH病毒第一次发作后4天，台湾警方追查出此病毒的原创者，当时在军队服役中的陈盈豪，便被[刑事警察局](#)约谈，碍于当时尚无相关刑事法令，陈盈豪获不起诉处分，民事部分也无受害者提出告诉[\[3\]\[4\]](#)。
- 曾经任职于技嘉科技公司工作，并与刑事局定时连络，协助开发过滤诈骗电话的预警功能。

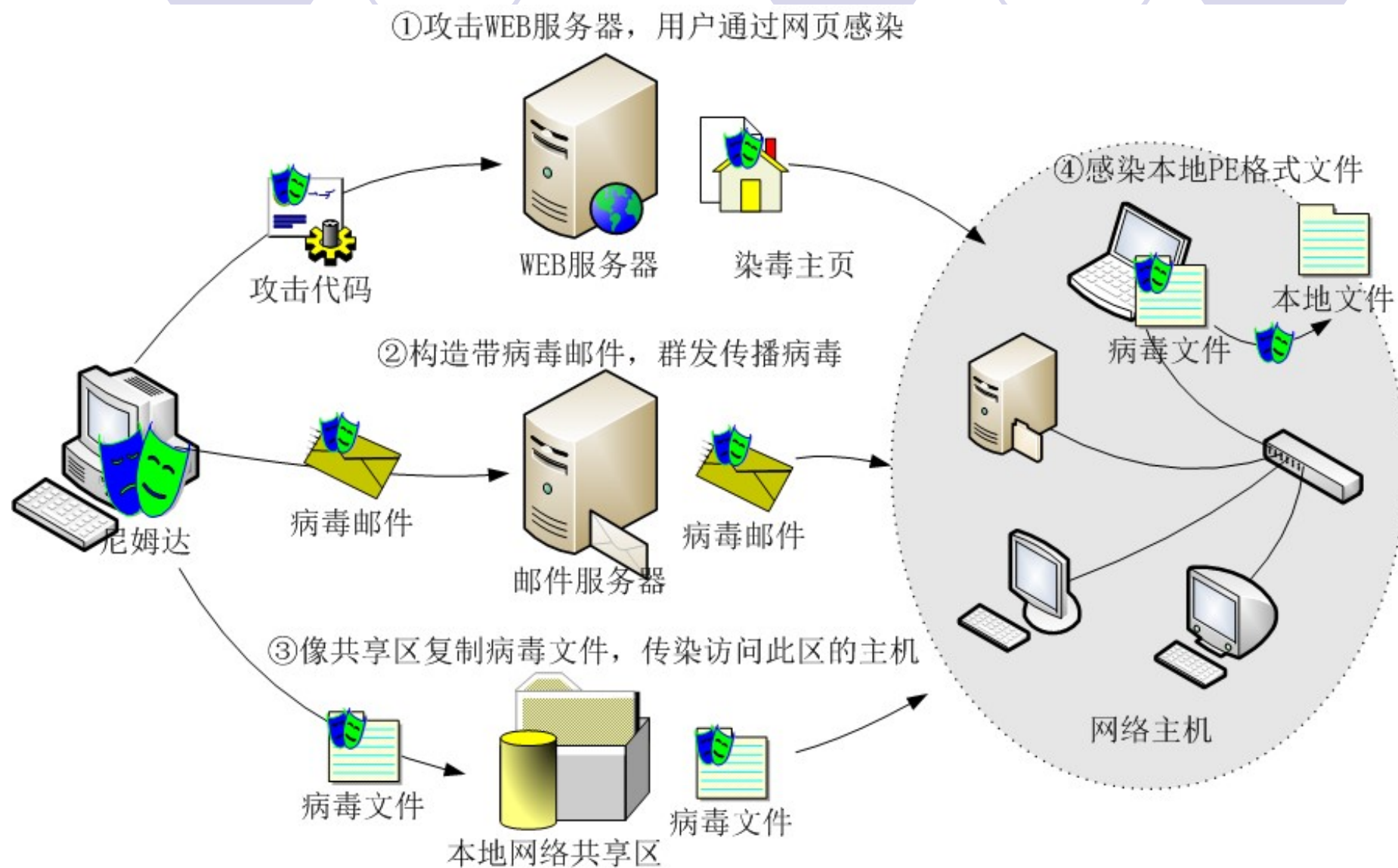
6.2.3 蠕虫病毒

- 蠕虫与传统病毒的区别：
 - 传统病毒是需要的寄生的，通过感染其它文件进行传播。
 - 蠕虫病毒一般不需要寄生在宿主文件中，传播途径主要包括局域网内的共享文件夹、电子邮件、网络中的恶意网页和大量存在着漏洞的服务器等。
 - 可以说蠕虫病毒是以计算机为载体，以网络为攻击对象。
- 蠕虫病毒能够利用漏洞，分为软件漏洞和人为缺陷
 - 软件漏洞主要指程序员由于习惯不规范、错误理解或想当然，在软件中留下存在安全隐患的代码
 - 人为缺陷主要指的是计算机用户的疏忽，这就是所谓的社会工程学（**Social Engineering**）问题。

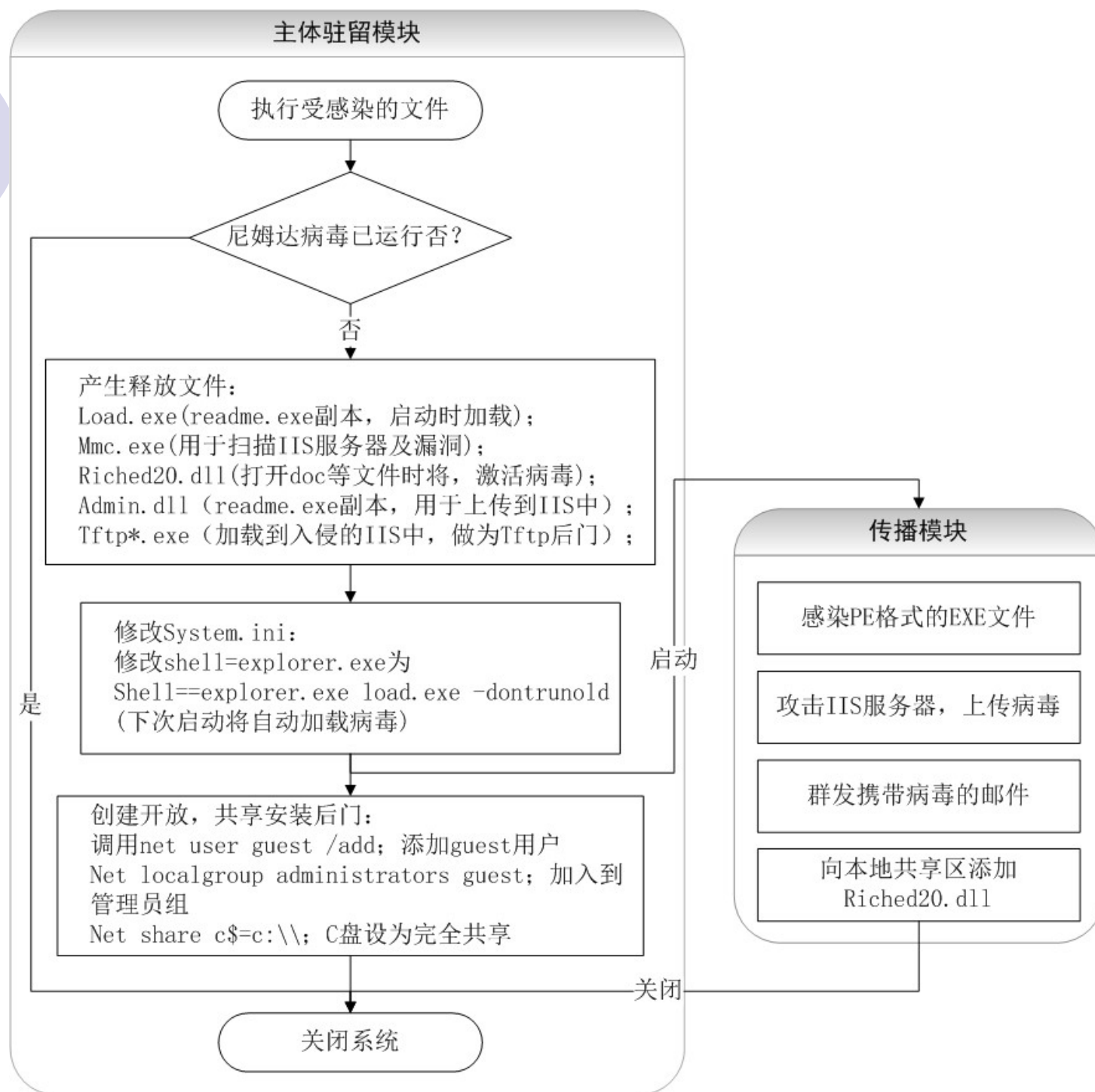
尼姆达蠕虫Worms.Nimda

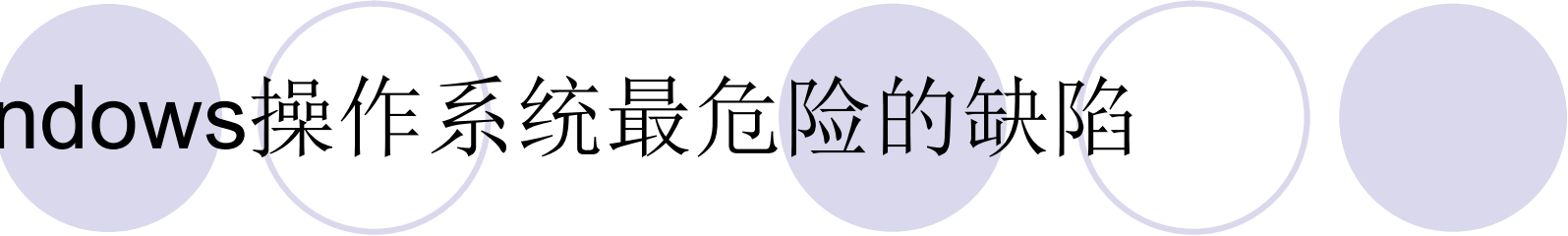
- 2001年9月18日尼姆达病毒在全球蔓延，它能够通过各种传播渠道进行传播，传染性极强，同时破坏力也极大。
 - 尼姆达病毒是一个精心设计的蠕虫病毒，其结构复杂堪称近年来之最。
 - 尼姆达病毒激活后，使用其副本替换系统文件；将系统的各驱动器设为开放共享，降低系统安全性；创建**Guest**账号并将其加入到管理员组中，安装**Guest**用户后门。
 - 由于尼姆达病毒通过网络大量传播，产生大量异常的网络流量和大量的垃圾邮件，网络性能势必受到严重影响。

Nimda 传播途径



尼姆达病毒程序

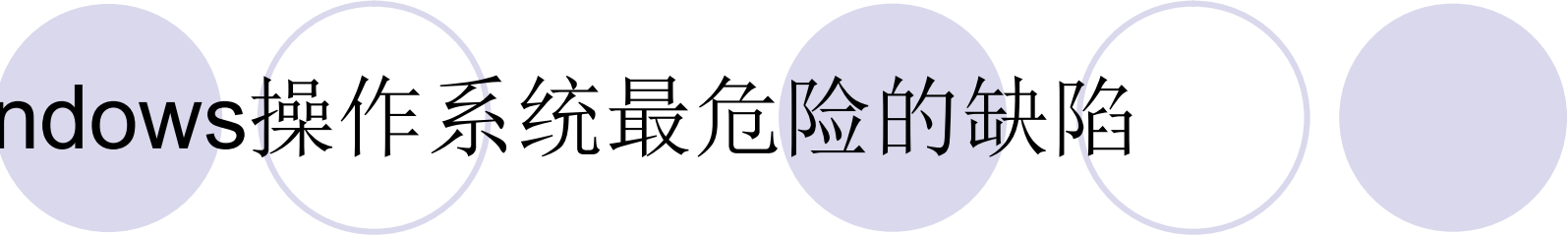




Windows操作系统最危险的缺陷

1. Unicode

- Unicode是一种编码。
- 不论何种平台，何种程序，何种语言，Unicode为每一个字符提供了一个独一无二的序号。
- Unicode标准被包括Microsoft在内的很多软件开发商所采用。



Windows操作系统最危险的缺陷

1. Unicode

- 通过向IIS服务器发出一个包括非法Unicode UTF-8序列的URL、攻击者可以迫使服务器逐字“进入或退出”目录并执行任意脚本，这种攻击被称为目录转换(Directory Traversal)攻击。

Windows操作系统最危险的缺陷

1. Unicode

- Unicode用“%2f”和“%5c”分别代表“/”和“\”。但也可以用所谓的“超长”序列来代表这些字符。
- “超长”序列是非法的Unicode表示符，它们比实际代表这些字符的序列要长。
 - “/”和“\”均可以用一个字节来表示。
 - 超长的表示法，例如用“%c0%af”代表“/”用了两个字节。

Windows操作系统最危险的缺陷

1. Unicode

- IIS不对超长序列进行检查。
- 这样在URL中加入一个超长的Unicode序列，就可以绕过Microsoft的安全检查。
 - 如果发出的请求来自一个可执行的目录，攻击者可以在服务器上运行可执行文件。
 - 安装IIS 4.0的Microsoft windows NT 4.0和安装了IIS 5.0，而没有安装service Pack 2的Windows 2000 server都存在着这个漏洞。

Windows操作系统最危险的缺陷

1. Unicode

- 这样的一个URL示例：

<http://192.168.0.27/Scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+d:\>

- 如果使用的是三字节UTF-8编码，也可以进行相同的攻击。下面的URL等价于上面的URL：

<http://192.168.0.27/Scripts/..%e0%c0%af../winnt/system32/cmd.exe?/c+dir+d:\>

Windows操作系统最危险的缺陷

1. Unicode

- 这样的攻击是如何完成的呢？
 - 这里的“%c0%af”是“/”的一个非法的Unicode表示。
 - URL使得Web服务器把这个Unicode字符理解为反斜杠，绕过了普通的Web服务器对这种事件的过滤，有效地退回了/Scripts/所在文件夹之上的两个文件夹层，并锁定了/wint/system32/cmd.exe。
 - /Scripts/文件夹通常位于c:\inetpub\Scripts目录这个位置。

Windows操作系统最危险的缺陷

1. Unicode

- 这样的攻击是如何完成的呢？
 - 在正常情况下，Web服务器绝不会允许URL访问Web文档目录（在这里是C:\inetpub）之外的任何位置。
- Web服务器在执行目录位置检验时并没有识别出“/”的Unicode表示。所以在服务器内部，..`%c0%af`../被解释为../../../../并且Web服务器访问的资源现在变成了C:\inetpub\scripts\..\..\winnt\system32\cmd.exe，而这个地址的最后又指向了c:\winnt\system32\cmd.exe，并执行了这个命令。

Windows操作系统最危险的缺陷

1. Unicode

- 那么 %c0%af 是如何转变为“/”的呢？
 - 我们需要解释非法Unicode表示是如何构造的。
 - “/”字符的十六进制ASCII码是2F，二进制表示为00101111。

Windows操作系统最危险的缺陷

1. Unicode

- 那么%c0%af是如何转变为“/”的呢？
 - Unicode编码，或者更准确地说是UTF-8编码，允许字符集包含多于256个字符，因此也就允许编码位数多于8位。UTF-8格式中表示2F的正确方法仍是2F。但是也可以使用多字节UTF-8来表示2F。

Windows操作系统最危险的缺陷

1. Unicode

- 那么%c0%af是如何转变为“/”的呢？
 - 字符“/”可以像这样使用单字节、双字节，和三字节的UTF-8编码格式表示。

使用的/	二进制表示	十进制表示	十六进制表示
单字节0xxxxxxx	00101111	47	2F
双字节110xxxxx 10xxxxxx	11000000 10101111	49327	C0 AF
三字节1110xxxx 10xxxxxx 10xxxxxx	11100000 10000000 10101111	14713007	E0 80 AF

Windows操作系统最危险的缺陷

1. Unicode

- UTF-8编码规范声明：“UTF-8解码器不得接受长于必要字符编码长度的UTF-8序列。任何超过长度的UTF-8序列都可能被滥用来绕过仅等待最短可能编码长度的UTF-8子字符串检测。”
- IIS没有遵循这一原则。
 - 结果，这方面的漏洞使得世界各地成千上万的黑客可以在IIS服务器上运行任何命令。

防范及清除

- 感染的用户应重新安装系统，以便彻底清除其它潜在的后门。如不能立刻重装系统，可参考下列步骤来清除蠕虫或者防止被蠕虫攻击：
 - ① 下载IE和IIS的补丁程序到受影响的主机上；
 - ② 安装杀毒软件和微软的CodeRedII清除程序；
 - ③ 备份重要数据；
 - ④ 断开网络连接(例如拔掉网线)；
 - ⑤ 执行杀毒工作，清除CodeRedII蠕虫留下的后门；
 - ⑥ 安装IE和IIS的补丁；
 - ⑦ 重启系统，再次运行杀毒软件以确保完全清除蠕虫。

6.2.4 木马

- 木马病毒，“木马计”，伪装潜伏的网络病毒。
 - 1986年的PC-Write木马是世界上第一个计算机木马
 - 木马是有隐藏性的、传播性的可被用来进行恶意行为的程序，因此，也被看作是一种计算机病毒。
 - 木马一般不会直接对电脑产生危害，以控制电脑为目的，当然电脑一旦被木马所控制，后果不堪设想。
- 木马的传播（种木马或植入木马）方式
 - 主要通过电子邮件附件、被挂载木马的网页以及捆绑了木马程序的应用软件。
 - 木马被下载安装后完成修改注册表、驻留内存、安装后门程序、设置开机加载等，甚至能够使杀毒程序、个人防火墙等防范软件失效。

木马病毒分类

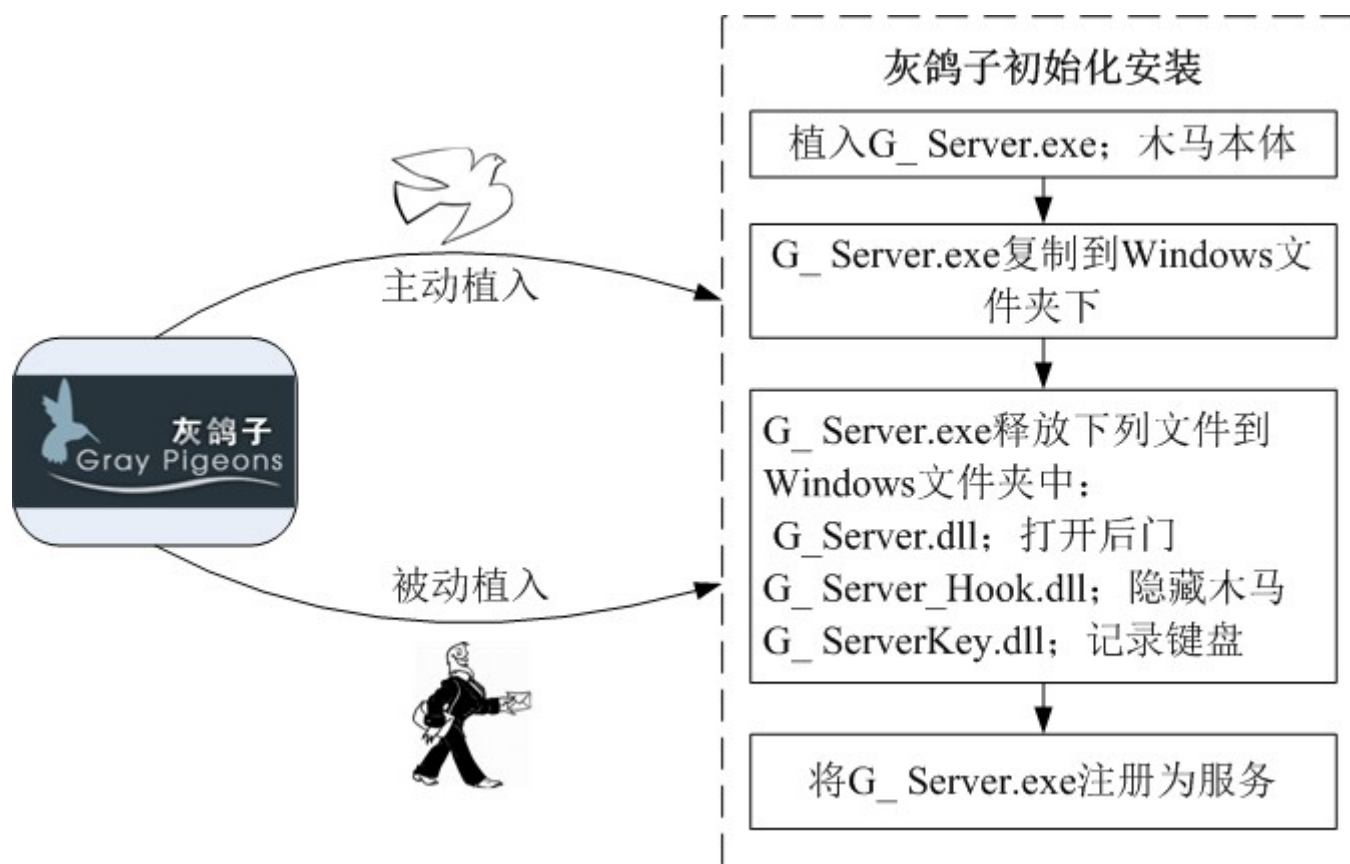
- (1) 盗号类木马
- (2) 网页点击类木马
- (3) 下载类木马
- (4) 代理类木马

木马病毒程序组成

- 控制端程序(客户端)
 - 是黑客用来控制远程计算机中的木马的程序;
- 木马程序(服务器端)
 - 是木马病毒的核心, 是潜入被感染的计算机内部、获取其操作权限的程序;
- 木马配置程序,
 - 通过修改木马名称、图标等来伪装隐藏木马程序, 并配置端口号、回送地址等信息确定反馈信息的传输路径。

灰鸽子的植入方法

- 被动植入是指植入过程必须依赖受害用户的手工操作；
- 主动植入是将灰鸽子程序通过程序自动安装到目标系统。



灰鸽子的隐藏技术

- 隐藏文件

- 隐藏进程

- 隐藏通讯

- 通讯端口复用技术是指将自己的通讯直接绑定到正常用户进程的端口，接收数据后，根据包格式判断是不是自己的，如果是它的，自己处理，否则通过**127.0.0.1**的地址交给真正的服务器应用进行处理。
- 反弹端口技术是指木马程序启动后主动连接客户，为了隐蔽起见，控制端的被动端口一般设置为**80**端口。对内部网络到外部网络的访问请求，防火墙一般不进行过于严格的检查，加之其连接请求有可能伪造成对外部资源的正常访问，因此可以通过防火墙。

灰鸽子背景

- 自称是一款远程控制软件，有时也被视为一种[木马](#)程序。
- 其为[程序员](#)葛军所开发，该软件除了支持正向连接外还支持反向连接，即客户端可以自动请求服务端连接，此外还有[摄像头](#)控制功能。
- [金山软件公司](#)在2007年曾经指责灰鸽子是“一条制造病毒、贩卖病毒、病毒培训为一体的黑色产业链”、“危害超出[熊猫烧香](#)10倍”，并摆明立场全面围剿灰鸽子。
- 灰鸽子工作室在该年停止版本更新、注册服务，并关闭官方网站。
- 常见的灰鸽子免杀技术有：[加壳压缩](#)、[加花](#)、修改[特征码](#)、修改程序入口点。
 - 另外在互联网上也存在着免杀版本的灰鸽子，并且不断更新。因为大多做了免杀[汇编](#)处理，所以对其客户端的查杀比较困难，大多数杀毒软件也无法识别它为病毒。
- 2013年起，灰鸽子相关（TM）商标由潍坊灰鸽子安防工程有限公司注册。意在将灰鸽子开发成为一款合理的正规的远程控制软件。

熊猫烧香

- 一种经过多次变种的计算机蠕虫病毒，2006年10月16日由25岁的中国湖北武汉新洲区人李俊编写，2007年1月初肆虐中国大陆网络，它主要透过网络下载的文件植入计算机系统。
- 2007年2月12日，湖北省公安厅宣布，李俊以及其同伙共8人已经落网，这是中国警方破获的首例计算机病毒大案。

熊猫烧香

● 病毒特征

- 使用Windows系统的用户中毒后，后缀名为.exe的文件无法执行，并且文件的图标会变成熊猫举着三根烧着的香的图案。^[1]，但不会感染操作系统的可执行文件。
- 而扩展名为.gho的赛门铁克公司软件Norton Ghost的系统磁盘备份文件也会被病毒自动检测并删除；大多数知名的网络安全公司的杀毒软件以及防火墙会被病毒强制结束进程，甚至会出现蓝屏、频繁重启的情况，
- 病毒还利用Windows2000/XP系统共享漏洞以及用户的弱口令如系统管理员密码为空，不少安全防范意识低的网吧以及局域网环境全部计算机遭此病毒的感染。同时病毒执行后在各盘释放autorun.inf以及病毒体自身，造成中毒者硬盘磁盘分区以及U盘、移动硬盘等可移动磁盘均无法正常打开。^[3]
- 由于此病毒具有在htm、html、asp、php、jsp、aspx等格式的网页文件中使用HTML的iframe标记元素嵌入病毒网页代码的能力，所以网页设计制作工作者的机器一旦中毒，那么使用过低版本或未更新安全补丁的Windows系列操作系统的网友访问他们设计的网站均会中此病毒。

熊猫烧香



● 病毒特征

- 李俊创建了病毒更新[服务器](#)，在更新最勤时一天要对病毒更新升级8次^[5]，与[俄罗斯](#)反病毒软件[卡巴斯基](#)反病毒库每3小时更新一次的更新速度持平，所以凭借更新的速度杀毒软件很难识别此计算机病毒的多种变种。

病毒案件的侦破与病毒作者

- 有人通过对此毒脱壳后的特征码分析发现有“whboy”的标识，而此标识也曾出现在2004年的一只病毒“[武汉男生](#)”上，所以该病毒也被称为“武汉男生”
- 通过查看李俊的早期作品可以看到他的QQ号码以及他创建的网站信息，有了这些信息，侦破案件的湖北[公共信息网络安全监察](#)的工作就容易了许多。

病毒案件的侦破与病毒作者

- 该病毒作者是李俊（1982年-），[武汉新洲区](#)人，
- 据家人以及朋友介绍，他在[初中](#)时[英语](#)和[数学](#)成绩都很不错，但还是没能考上[高中](#)，[中专](#)在[黄石职业技术学院](#)就读，学习的是[水泥工艺](#)专业，毕业后曾上过网络技术职业培训班
- 他朋友讲他是“自学成才，他的大部分电脑技术都是看书自学的”。

病毒案件的侦破与病毒作者

- 2004年李俊到[北京](#)、[广州](#)的[网络安全](#)公司求职，但都因学历低的原因遭拒，于是他开始抱着报复社会以及赚钱的目的编写病毒了。
- 曾在2003年编写了病毒“武汉男生”，2005年他还编写了病毒[QQ尾巴](#)，并对“武汉男生”版本更新成为“武汉男生2005”。

病毒案件的侦破与病毒作者

- 此次传播的“熊猫烧香”病毒，作者李俊是先将此病毒在网络中卖给了120余人，每套产品要价500~1000元人民币，每日可以收入8000元左右，最多时一天能赚1万余元人民币，作者李俊因此直接非法获利10万余元。
- 然后由这120余人对此病毒进行改写处理并传播出去的，这120余人的传播造成100多万台计算机感染此病毒，他们将盗取来的网友网络游戏以及QQ帐号进行出售牟利，并使用被病毒感染沦陷的机器组成“僵尸网络”为一些网站带来流量。

被逮捕的几位重要犯罪嫌疑人资料：

姓名	性别	到案年龄	住所	出生
李俊	男	25岁	武汉新洲区 人	1982年（33–34岁）
雷磊	男	25岁	武汉新洲区人	1982年（33–34岁）
王磊	男	22岁	山东威海 人	1985年（30–31岁）
叶培新	男	21岁	浙江温州 人	1986年（29–30岁）
张顺	男	23岁	浙江 丽水 人	1984年（31–32岁）
王哲	男	24岁	湖北仙桃 人	1983年（32–33岁）

被逮捕的几位重要犯罪嫌疑人资料

- 2007年9月24日，湖北省仙桃市人民法院一审以破坏计算机信息系统罪判处
 - 李俊有期徒刑四年
 - 王磊有期徒刑二年六个月
 - 张顺有期徒刑二年
 - 雷磊有期徒刑一年
 - 并判决李俊、王磊、张顺的违法所得予以追缴。

被逮捕的几位重要犯罪嫌疑人资料

- 作者李俊出狱后，开始经营一家企业网络安全软件公司。
- 2013年6月13日晚，据“丽水发布”官方微博消息，“熊猫烧香”病毒制造者张顺、李俊在浙江丽水设立网络赌场，敛财数百万元，已被当地检察机关批捕。
 - 该事件发生在2013年初，涉案者共17人。

6.2.5 病毒防治

- 病毒防治技术略滞后于病毒技术
- 通常需要选择一个有效的防病毒产品，并及时进行产品升级。
- 计算机病毒防治技术主要包括：
 - 检测
 - 清除
 - 预防
 - 免疫

检测



- 特征代码法

- 特征代码查毒就是检查文件中是否含有病毒数据库中的病毒特征代码。

- 校验和法

- 对正常状态下的重要文件进行计算，取得其校验和，以后定期检查这些文件的校验和与原来保存的校验和是否一致。

检测(续)

- 行为监测法

- 利用病毒的特有**行为特征**来监测病毒的方法，称为行为监测法。当一个可疑程序运行时，监视其行为，如果发现了病毒行为，立即报警。

- 软件模拟法

- 软件模拟法是为了对付多态型病毒。
- 软件模拟法是通过模拟病毒的执行环境，为其构造虚拟机，然后在虚拟机中执行病毒引擎解码程序，安全地将多态型病毒解开并还原其本来面目，再加以扫描。
- 软件模拟法的**优点**是可识别未知病毒、病毒定位准确、误报率低；**缺点**是检测速度受到一定影响、消耗系统资源较高。

计算机中毒的常见症状

- 系统运行速度减慢；
- 系统经常无故发生死机
- 文件长度发生变化；
- 存储的容量异常减少；
- 丢失文件或文件损坏；
- 屏幕上出现异常显示；
- 系统的蜂鸣器出现异常声响；
- 磁盘卷标发生变化；
- 系统不识别硬盘；
- 对存储系统异常访问；
- 键盘输入异常；
- 文件的日期、时间、属性等发生变化；
- 文件无法正确读取、复制或打开；
- 命令执行出现错误；
- **WINDOWS**操作系统无故频繁出现错误；
- 系统异常重新启动；
- 一些外部设备工作异常；
- 出现异常的程序驻留内存

清除

- 清除病毒主要分为

- 使用防病毒软件和手工清除病毒两种方法。

- 防病毒软件

- 防病毒软件由安全厂商精心研制，可以有效查杀绝大多数计算机病毒，多数用户应采用防病毒软件来清除病毒。

防病毒软件

- 对检测到的病毒一般采取三种处理方案，分别是清除、隔离和删除。

- 清除是指在发现文件被感染病毒时，采取的清除病毒并保留文件的动作。
- 隔离是指在发现病毒后，无法确认清除动作会带来什么后果，又不想直接删除文件，故采取监视病毒并阻止病毒运行的方法。
- 某类病毒清除失败、删除失败、隔离失败，对个人用户来讲，格式化硬盘、重建系统可能就是最后的有效选择。

蠕虫、木马等病毒的清除

- 结束所有可疑进程
- 删除病毒文件并恢复注册表
- 内核级后门的清除
- 重启后扫描
 - 完成了上述三步，随后需要重新启动系统，并使用带有最新病毒库的防病毒软件对全盘进行扫描（这一步非常重要，做不好的话前功尽弃）



预防

- 安装防毒软件
 - 打开你的防毒软件的自动升级服务，定期扫描计算机
- 注意软盘、光盘以及U盘等存储媒介
 - 在使用软盘、光盘、U盘或活动硬盘前，病毒扫描
- 关注下载安全
 - 下载要从比较可靠的站点进行，下载后做病毒扫描。
- 关注电子邮件安全
 - 来历不明的邮件决不要打开，决不要轻易运行附件
- 使用基于客户端的防火墙
- 警惕欺骗性的病毒
- 备份

免疫

- 计算机病毒免疫

- 提高计算机对计算机病毒的抵抗力，从而达到防止病毒侵害的目的
- 具体方法
 - 提高计算机系统的健壮性
 - 给计算机注射“病毒疫苗”

提高系统健壮性

- 主要途径包括以下内容：
 - 及时升级操作系统，保证系统安装最新的补丁；
 - 安装防病毒软件，及时升级病毒定义文件和防病毒引擎；
 - 定期扫描系统和磁盘文件；
 - 打开个人防火墙；
 - 使用软盘或U盘写保护
 - 重要的数据信息写入只读光盘；

注射“病毒疫苗”

- 实施免疫的主要方法包括以下几个方面：

- 感染标识免疫

- 人为地为正常对象中加上病毒感染标识，使计算机病毒误以为已经感染从而达到免疫的目的。

- 文件扩展名免疫

- 将扩展名改为非COM、EXE、SYS、BAT等形式，
 - 将系统默认的可执行文件的后缀名改为非COM、EXE、SYS、BAT等形式。

- 外部加密免疫

- 外部加密免疫是指在文件的存取权限和存取路径上进行加密保护，以防止文件被非法阅读和修改。

- 内部加密免疫

- 对文件内容加密变换后进行存储，在使用时再进行解密。

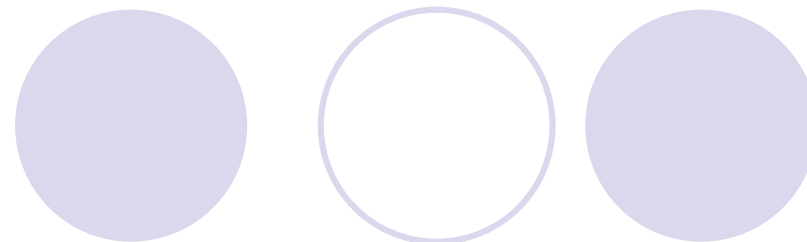
6.3 网络入侵

- 1980年，James P Anderson首次提出了“入侵”的概念，
 - “入侵”是指在非授权的情况下，试图存取信息、处理信息或破坏系统，以使系统不可靠或不可用的故意行为。
 - 网络入侵一般是指具有熟练编写、调试和使用计算机程序的技巧的人，利用这些技巧来获得非法或未授权的网络或文件的访问，进入内部网的行为。
 - 对信息的非授权访问一般被称为破解cracking。

网络入侵

● 步骤

- 前期准备
- 实施入侵
- 后期处理



网络入侵

● 准备阶段

- 需要完成的工作主要包括明确入侵目的、确定入侵对象以及选择入侵手段，
 - 入侵目的一般可分为控制主机、瘫痪主机和瘫痪网络；
 - 入侵对象一般分为主机和网络两类；
 - 根据目的和后果分为：拒绝服务攻击、口令攻击、嗅探攻击、欺骗攻击和利用型攻击。

网络入侵

● 实施入侵阶段

○ 是真正的攻击阶段，主要包括扫描探测和攻击。

- 扫描探测主要用来收集信息，为下一步攻击奠定基础；
- 攻击：根据入侵目的、采用相应的入侵手段向入侵对象实施入侵。

网络入侵

- 后期处理

- 主要是指由于大多数入侵攻击行为都会留下痕迹，攻击者为了清除入侵痕迹而进行现场清理。

6.3.1 拒绝服务攻击

- 拒绝服务攻击DoS (Denial of Service)
 - DoS并不是某一种具体的攻击方式，而是攻击所表现出来的结果最终使得目标系统因遭受某种程度的破坏而不能继续提供正常的服务，甚至导致物理上的瘫痪或崩溃。
- 通常拒绝服务攻击可分为两种类型，
 - 第一类攻击是**利用网络协议的缺陷**，通过发送一些非法数据包致使主机系统瘫痪；
 - 第二类攻击是通过**构造大量网络流量**致使主机通讯或网络堵塞，使系统或网络不能响应正常的服务。

Ping of Death



- TCP/IP的规范，一个包的长度最大为65536字节。
- 利用多个IP包分片的叠加能做到构造长度大于65536的IP数据包。
- 攻击者通过修改IP分片中的偏移量和段长度，使系统在接收到全部分段后重组报文时总的长度超过了65535字节。
- 一些操作系统在对这类超大数据包的处理上存在缺陷，当安装这些操作系统的主机收到了长度大于65536字节的数据包时，会出现内存分配错误，从而导致TCP/IP堆栈崩溃，造成死机。

Tear drop



- IP数据包在网络传递时，数据包可能被分成多个更小的IP分片。
- 攻击者可以通过发送两个（或多个）IP分片数据包来实现Tear Drop攻击。
- 第一个IP分片包的偏移量为0，长度为N，第二个分片包的偏移量小于N，未超过第一个IP分片包的尾部，这就出现了偏移量重叠现象。
- 一些操作系统无法处理这些偏移量重叠的IP分片的重组，TCP/IP堆栈会出现内存分配错误，造成操作系统崩溃。

Syn Flood

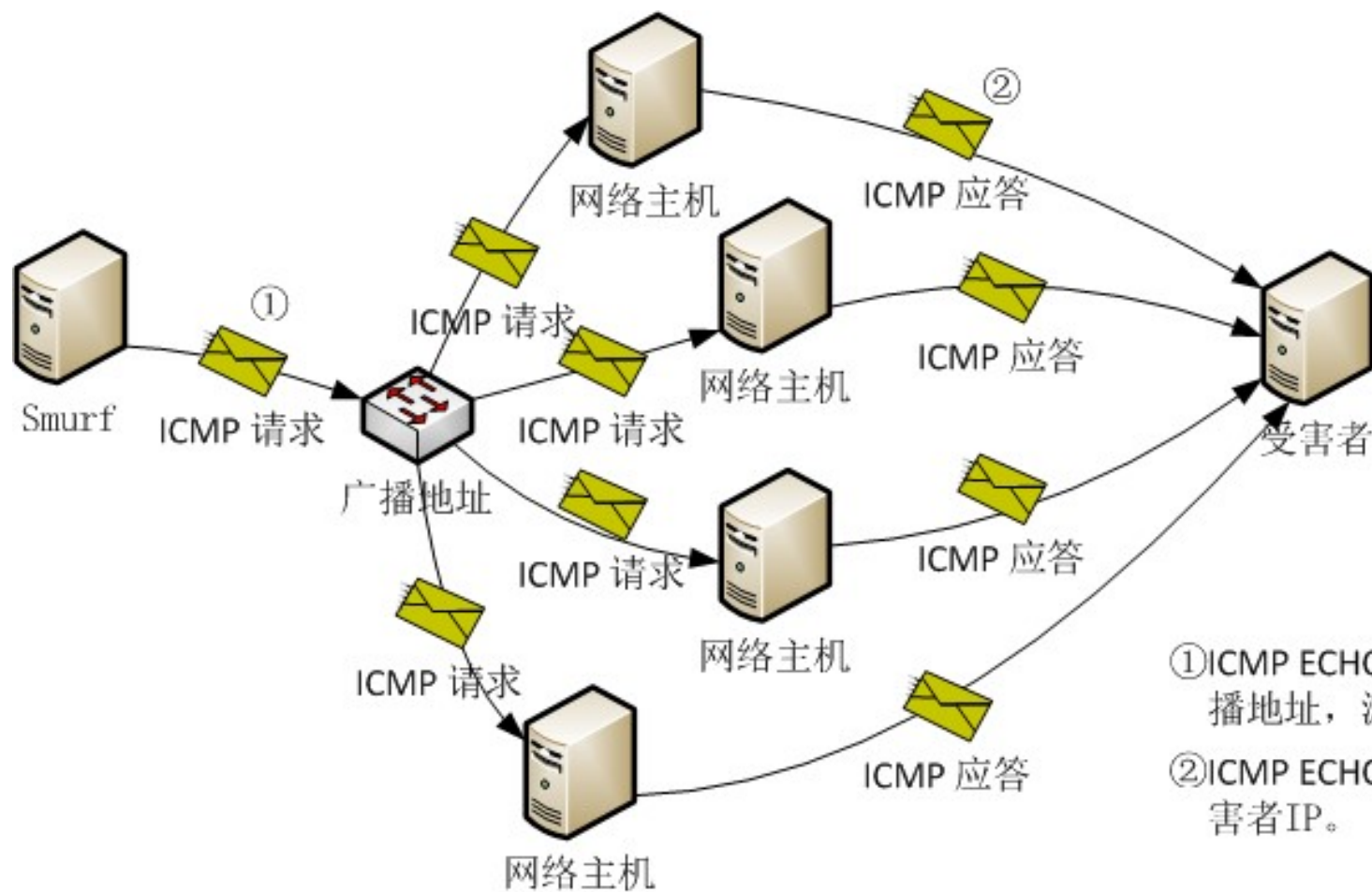


- 攻击者伪造**TCP**的连接请求，向被攻击的设备正在监听的端口发送大量的**SYN**连接请求报文；
- 被攻击的设备按照正常的处理过程，回应这个请求报文，同时为它分配了相应的资源。
- 攻击者不需要建立**TCP**连接，因此服务器根本不会接收到第三个**ACK**报文，现有分配的资源只能等待超时释放。
- 如果攻击者能够在超时时间到达之前发出足够多的攻击报文，被攻击的系统所预留所有**TCP**缓存将被耗尽。

Smurf攻击

- **Smurf**攻击是以最初发动这种攻击的程序**Smurf**来命名的，这种攻击方法结合使用了**IP**地址欺骗和**ICMP**协议。
- 当一台网络主机通过广播地址将**ICMP ECHO**请求包发送给网络中的所有机器，网络主机接收到请求数据包后，会回应一个**ICMP ECHO**响应包，这样发送一个包会收到许多的响应包。
- **Smurf**构造并发送源地址为受害主机地址、目的地址为广播地址的**ICMP ECHO**请求包，收到请求包的网络主机会同时响应并发送大量的信息给受害主机，致使受害主机崩溃。
- 如果**Smurf**攻击将回复地址设置成受害网络的广播地址，则网络中会充斥大量的**ICMP ECHO**响应包，导致网络阻塞。

Smurf攻击过程示意图



- ①ICMP ECHO请求包：目的地址为广播地址，源地址为受害者IP；
- ②ICMP ECHO应答包：目的地址为受害者IP。

电子邮件炸弹

- 实施电子邮件炸弹攻击的特殊程序称为**Email Bomber**。

- 邮箱容量是有限的，用户在短时间内收到成千上万封电子邮件，每个电子邮件的容量也比较大，那么经过一轮邮件炸弹轰炸后电子邮箱的容量可能被占满。
- 另外一方面，这些电子邮件炸弹所携带的大容量信息不断在网络上来回传输，很容易堵塞网络；
- 而且邮件服务器需要不停地处理大量的电子邮件，如果承受不了这样的疲劳工作，服务器随时有崩溃的可能。

DDoS



- DDoS攻击就是很多DoS攻击源一起攻击某台服务器或网络，迫使服务器停止提供服务或网络阻塞。
- DDoS攻击需要众多攻击源，而黑客获得攻击源的主要途径就是传播木马，网络计算机一旦中了木马，这台计算机就会被后台操作的人控制，也就成了所谓的“肉鸡”，即黑客的帮凶。
- 使用“肉鸡”进行DDoS攻击还可以在在一定程度上保护攻击者，使其不易被发现。

对于DoS的防御

- 及时为系统升级，减少系统漏洞，很多DoS攻击对于新的操作系统已经失效，如Ping of Death攻击；
- 关掉主机或网络中的不必要的服务和端口，如对于非WEB主机关掉80端口；
- 局域网应该加强防火墙和入侵检测系统的应用和管理，过滤掉非法的网络数据包。

6.3.2 口令攻击

- 口令攻击过程一般包括以下几个步骤。
 - 步骤一、获取目标系统的用户帐号及其它有关信息；
 - 获取目标系统的用户帐号及其它有关信息一般可以利用一些网络服务来实现，如Finger、Whois、LDAP等信息服务。
 - 步骤二、根据用户信息猜测用户口令；
 - 步骤三、采用字典攻击方式探测口令；
 - 使用一些程序，自动地从电脑字典中取出一个单词，作为用户的口令输入给远端的主机，进入系统。
 - 如果口令错误，就按序取出下一个单词，进行下一个尝试。并一直循环下去，直到找到正确的口令或字典的单词试完为止。
 - 由于这个破译过程由计算机程序来自动完成，几个小时就可以把字典的所有单词都试一遍。
 - 步骤四、探测目标系统的漏洞，伺机取得口令文件，破解取得用户口令。



- 系统中可以用作口令的字符有**95**个，
 - 10个数字、33个标点符号、52个大小写字母。
 - 采用任意5个字母加上一个数字或符号则可能的排列数约为163亿，即 $52^5 \times 43 = 16,348,773,000$ 。
- 这个数字对于每秒可以进行上百万次浮点运算的计算机并不是什么困难问题，也就是说一个**6位的口令将不是安全的**
- 一般建议使用**10位以上**并且是字母、数字加上标点符号的混合体。

防范口令攻击的方法

- 口令的长度不少于10个字符；
- 口令中要有一些非字母；
- 口令不在英语字典中；
- 不要将口令写下来；
- 不要将口令存于电脑文件中；
- 不要选择易猜测的信息做口令；
- 不要在不同系统上使用同一口令；
- 不要让其他人得到口令；
- 经常改变口令；
- 永远不要对自己的口令过于自信。

6.3.3 嗅探攻击

- 嗅探攻击也称为**网络嗅探**，是指利用计算机的网络接口截获目的地为其它计算机的数据包的一种手段。
- 网络嗅探的工具被称为**嗅探器**（**sniffer**），是一种常用的收集网络上传输的有用数据的方法，
- 嗅探攻击一般是指黑客利用嗅探器获取网络传输中的重要数据。网络嗅探也被形象地称为网络窃听。

共享网络环境

- 以太网卡共有四种工作方式：
 - 广播方式：网卡能够接收网络中的广播数据；
 - 组播方式：网卡能够接收组播数据；
 - 直接方式：只有目的网卡才能接收该数据；
 - 混杂模式：网卡能够接收一切通过它的数据。
- 如果攻击者获得其中一台主机的root权限，并将其网卡置于混杂模式，这就意味着不必打开配线盒来安装偷听设备，就可以在对共享环境下的其它计算机的通信进行窃听，在共享网络中网络通信没有任何安全性可言。

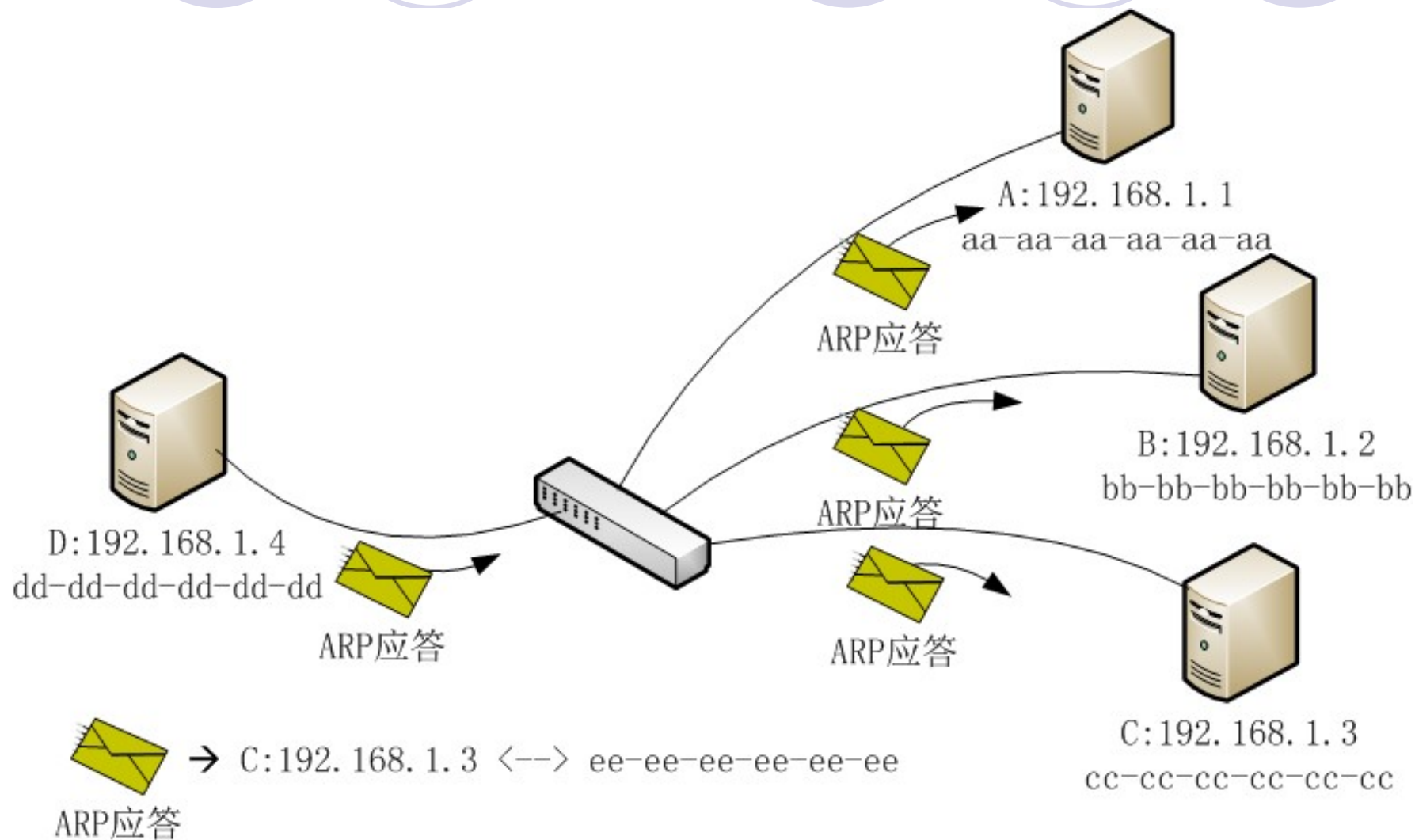
交换网络环境

- Arp协议

- 当主机接收到**ARP**应答数据包的时候，就使用应答数据包内的数据对本地的**ARP**缓存进行更新或添加。

Internet	地址	物理地址
192.168.1.100	00-30-48-31-26-98	动态
192.168.1.101	00-00-00-00-01-89	动态
192.168.1.102	00-24-dc-b8-47-f0	动态
192.168.1.255	ff-ff-ff-ff-ff-ff	静态

Arp欺骗



防范嗅探攻击

- 检测嗅探器

- 检测混杂模式网卡来检查嗅探器的存在，**AntiSniff**。

- 安全的拓扑结构

- 嗅探器只能在当前网络段上进行数据捕获。将网络分段工作进行得越细，嗅探器能够收集的信息就越少。

- 会话加密

- 即使嗅探器嗅探到数据报文，也不能识别其内容。

- 地址绑定

- 在客户端使用**arp**命令绑定网关的真实**MAC**地址；
- 在交换机上做端口与**MAC**地址的静态绑定；
- 在路由器上做**IP**地址与**MAC**地址的静态绑定；
- 用静态的**ARP**信息代替动态的**ARP**信息。

6.3.4 欺骗类攻击

- 欺骗类攻击是指构造虚假的网络消息，发送给网络主机或网络设备，企图用假消息替代真实信息，实现对网络及主机正常工作的干扰破坏。
- 常见的假消息攻击有**IP**欺骗、**ARP**欺骗、**DNS**欺骗、伪造电子邮件等

IP欺骗

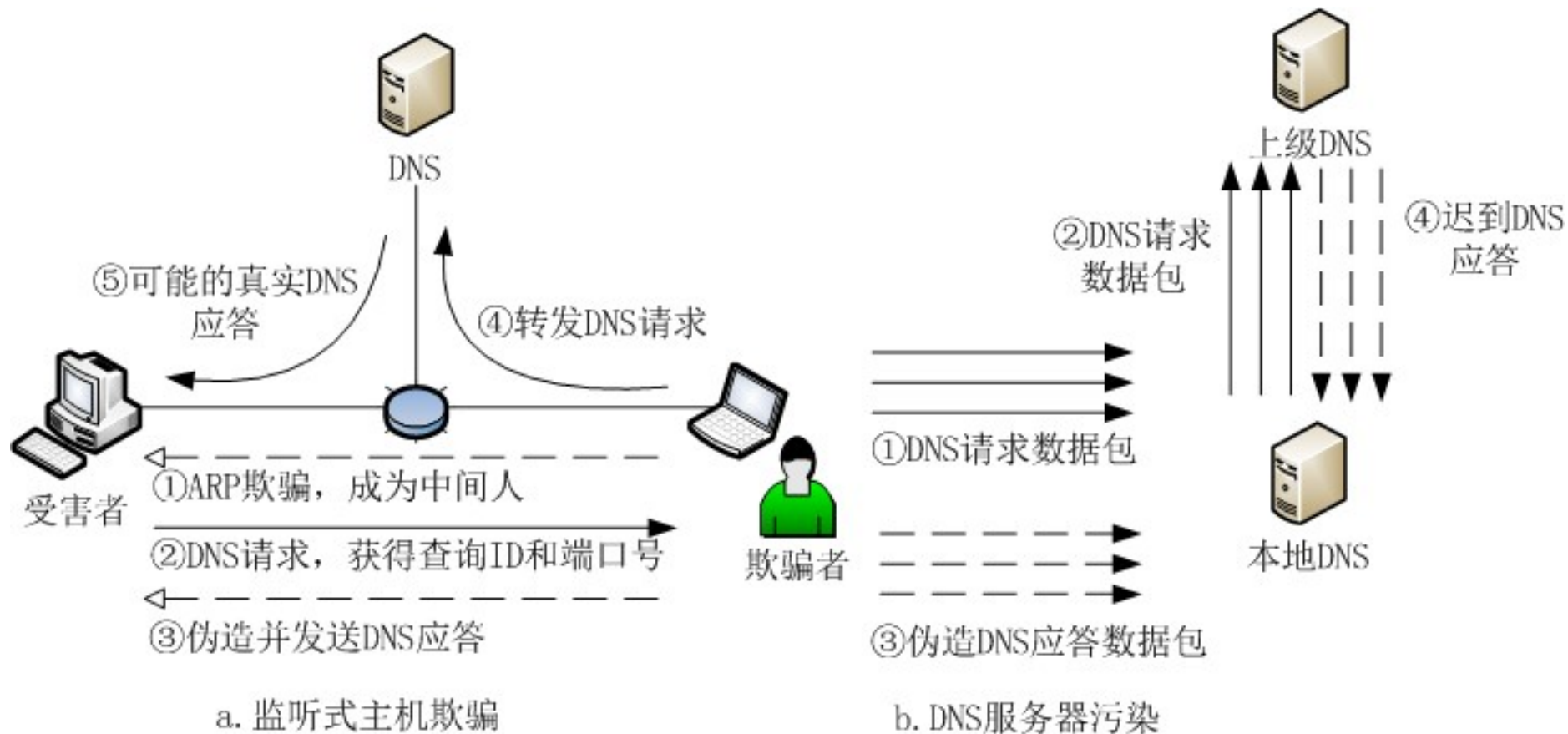


- IP欺骗简单地说就是一台主机设备冒充另外一台主机的IP地址，与其它设备通信。
- IP欺骗主要是基于远程过程调用RPC的命令，比如rlogin、rcp、rsh等，
- 这些命令仅仅根据信源IP地址进行用户身份确认，以便允许或拒绝用户RPC。
- IP欺骗的目的主要是获取远程主机的信任及访问特权。

IP欺骗攻击主要步骤

- 第一步 选定目标主机并发现被该主机信任的其它主机；
- 第二步 使得被信任的主机丧失工作能力；
- 第三步 使用被目标主机信任的主机的IP地址，伪造建立TCP连接的SYN请求报文，试图以此数据报文建立与目标主机的TCP连接；
- 第四步 序列号取样和猜测。
- 第五步 使用被目标主机信任的主机的IP地址和计算出的TCP 序列号，构造TCP连接的ACK报文，发送给目标主机，建立起与目标主机基于地址验证的应用连接。
 - 如果成功，攻击者可以使用一种简单的命令放置一个系统后门，以进行非授权操作。

DNS欺骗



伪造电子邮件

- 由于**SMTP**并不对邮件的发送者的身份进行鉴定，攻击者可以冒充别的邮件地址伪造电子邮件。
- 攻击者伪造电子邮件的目的主要包括：
 - 攻击者想隐藏自己的身份，匿名传播虚假信息，如造谣中伤某人；
 - 攻击者想假冒别人的身份，提升可信度，如冒充领导发布通知；
 - 伪造用户可能关注的发件人的邮件，引诱收件人接收并阅读，如传播病毒、木马等。

对于欺骗类攻击的防范方法

- 抛弃基于地址的信任策略，不允许使用r类远程调用命令。
- 配置防火墙，拒绝网络外部与本网内具有相同IP地址的连接请求；过滤掉入站的DNS更新。
- 地址绑定，在网关上绑定IP地址和MAC地址；在客户端使用arp命令绑定网关的真实MAC地址命令。
- 使用PGP等安全工具并安装电子邮件证书。

6.3.5 利用型攻击

- 利用型攻击是通过非法技术手段，试图获得某网络计算机的控制权或使用权，达到利用该机从事非法行为的一类攻击行为的总称。
- 利用型攻击常用的技术手段主要包括：
 - 口令猜测、木马病毒、僵尸病毒以及缓冲区溢出等。

僵尸病毒（Bot）

- 通过特定协议的信道连接僵尸网络服务器的客户端程序，
 - 被安装了僵尸程序的机器称为**僵尸主机**，
 - 僵尸网络（**BotNet**）是由这些受控的僵尸主机依据特定协议所组成的网络。
- 程序结构与木马程序基本一致，
 - 木马程序是被控制端连接的服务器端程序，
 - 僵尸程序是向控制服务器发起连接的客户端程序。
- 传播和木马相似
 - 途径包括电子邮件、含有病毒的**WEB**网页、捆绑了僵尸程序的应用软件以及利用系统漏洞攻击加载等。
- 黑客经常利用其发起大规模的网络攻击，
 - 如分布式拒绝服务攻击（**DDoS**）、海量垃圾邮件等，

缓冲区溢出

- 指当计算机向缓冲区内填充数据位数时超过了缓冲区本身的容量，溢出的数据覆盖了合法数据。
 - 是一种非常普遍、非常危险的程序漏洞，在各种操作系统、应用软件中广泛存在。
 - 利用缓冲区溢出攻击，可以导致程序运行失败、系统宕机、重新启动等后果，更为严重的是可以利用它执行非授权指令，甚至可以取得系统特权并控制主机，进行各种非法操作。

缓冲区的理论基础

- 缓冲区溢出的产生存在着必然性，现代计算机程序的运行机制、C语言的开放性及编译问题是其产生的理论基础。
 - 程序在**4GB**或更大逻辑地址空间内运行时，一般会被装载到相对固定的地址空间，使得攻击者可以估算用于攻击的代码的逻辑地址；
 - 程序调用时，可执行代码和数据共同存储在一个地址空间（堆栈）内，攻击者可以精心编制输入的数据，通过运行时缓冲区溢出，得到运行权；
 - **CPU call**调用时的返回地址和**C**语言函数使用的局部变量均在堆栈中保存，而且**C**语言不进行数据边界检察，当数据被覆盖时也不能被发现。

缓冲区溢出

- 缓冲区溢出(Buffer Overflow)是目前最普遍的攻击手段
 - 黑客利用某些程序的设计缺陷，通过缓冲区溢出非法获得某些权限。
 - 溢出攻击通常可以分为远程溢出和本地溢出，其中尤其以远程溢出的威胁性最大。
 - 利用远程溢出，黑客可以在没有任何系统账号的情况下获得系统的最高控制权。

缓冲区溢出

- 缓冲区溢出原理：

- 简单地说，缓冲区溢出就是向一个有限空间的缓冲区拷贝了过长的字符串，覆盖相邻的存储单元，这将会引起程序运行失败。
- 因为变量保存在堆栈当中，当发生缓冲区溢出的时候，存储在堆栈中的函数返回地址也会被覆盖，造成缓冲区的溢出，从而破坏程序的堆栈，使程序转而执行其它指令，以达到攻击的目的。

缓冲区溢出

- 造成缓冲区溢出的原因

- 程序中没有仔细检查用户输入的参数。所以说缓冲区溢出的缺陷属于输入确认错误。

缓冲区溢出

- 为了理解缓冲区溢出的机制，我们先看一个例子：

```
#include <string.h>
void function(char *str)
{
    char buffer[16];

    strcpy(buffer,str);
}
```

```
void main()
{
    char large_string[256];
    int i;

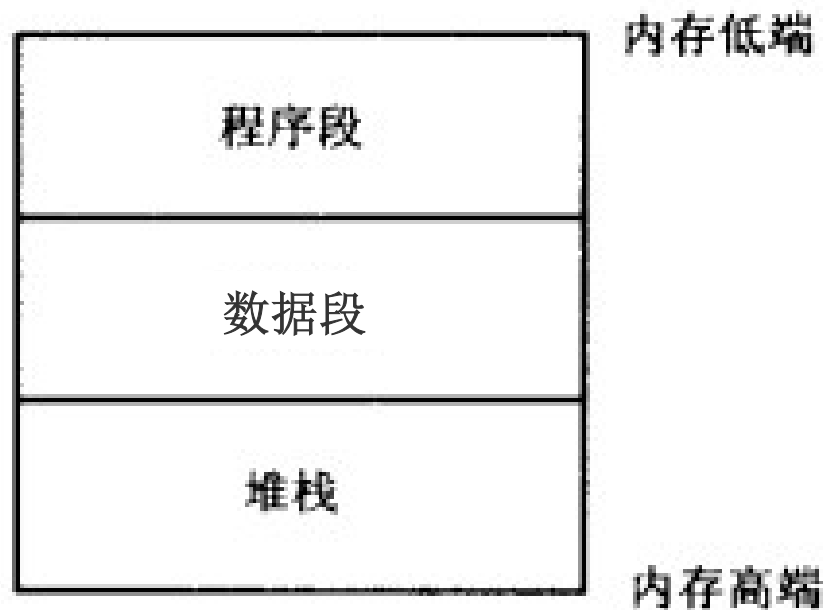
    for(i=1;i<255;i++)
        large_string[i]='A';
    function(large_string);
}
```

缓冲区溢出

- 编译并运行程序的结果是：
- “0x41414141”指令引用的“0x41414141”内存。该内存不能为“read”。
- 为什么呢？ 这跟内存存储数据的原理有关。

缓冲区溢出

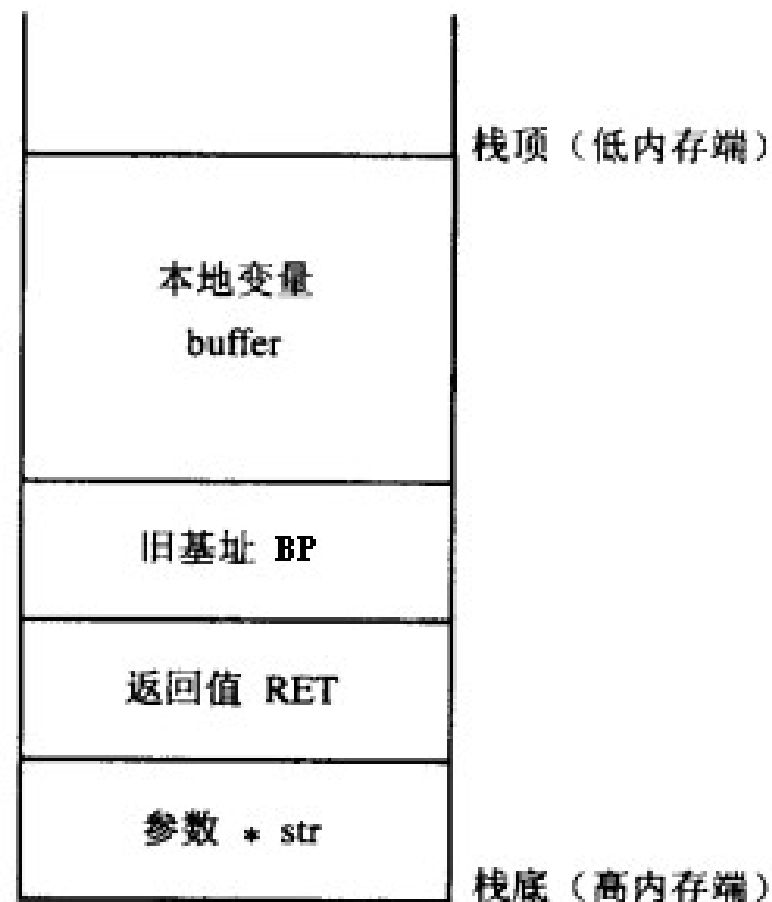
- 一个程序在内存中通常分为程序段，数据段和堆栈段3部分。
 - 程序段里放着程序的机器码和只读数据。
 - 数据段放的是程序中的静态数据。
 - 动态数据则通过堆栈来存放。



一个程序在内存中的存放

缓冲区溢出

- 当程序中发生函数调用时，计算机做如下操作：
 - 首先把参数压入堆栈；
 - 然后保存指令寄存器(IP)中的内容作为返回地址(RET)；
 - 第三个放入堆栈的是基址寄存器(BP)：然后把当前的栈指针(SP)拷贝到BP，作为新的基地址；
 - 最后为本地变量留出一定空间，把SP减去适当的数值。



调用一个函数后的堆栈

缓冲区溢出

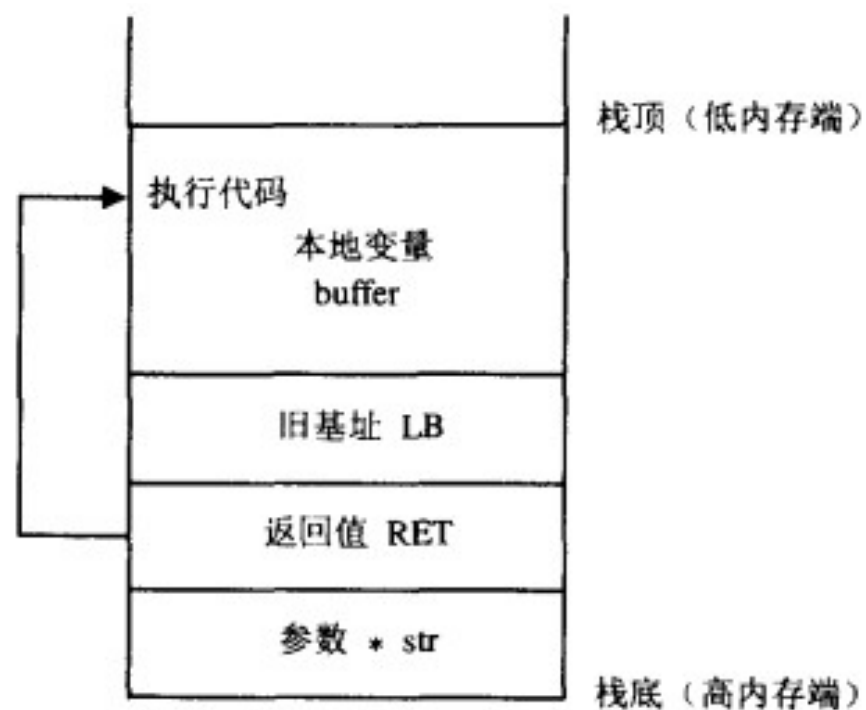
- 很显然，程序执行的结果是“Segmentation fault(core dumped)”或类似的出错信息。
 - 因为从buffer开始的256个字节都将被*str的内容‘A’覆盖，包括BP，RET，甚至*str。‘A’的十六进值为0x41，所以函数的返回地址变成了0x41414141，这超出了程序的地址空间，所以出现上面的错误。

缓冲区溢出

- 是否能够通过控制返回地址，转到想要执行的程序入口，进而可以控制整个系统呢？
 - 这正是缓冲区溢出方法的精华所在。

缓冲区溢出

- 如果在溢出的缓冲区中写入我们想执行的代码，再覆盖返回地址(**ret**)的内容，使它指向缓冲区的开头，就可以达到运行其他指令的目的。



通过堆栈返回指令执行代码

缓冲区溢出

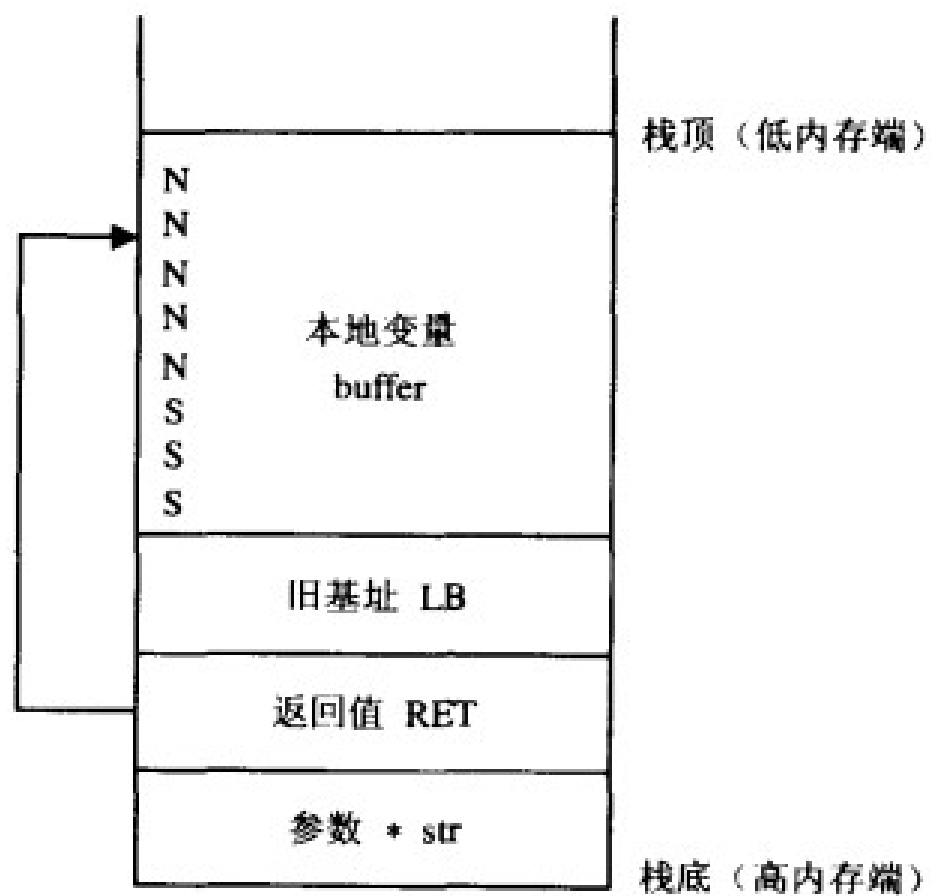
- 利用缓冲区溢出进行的系统攻击

- 如果已知某个程序有缓冲区溢出的缺陷，如何知道缓冲区的地址，在哪儿放入攻击代码呢？

- 由于每个程序的堆栈起始地址是固定的，所以理论上可以通过反复重试缓冲区相对于堆栈起始位置的距离来得到。但这样的盲目猜测可能要进行数百上千次，实际上是不现实的。

- 解决的办法是利用空指令**NOP**。在攻击代码前面放一长串的**NOP**，返回地址可以指向这一串**NOP**中任一位置，执行完**NOP**指令后程序将激活攻击进程。这样就大大增加了猜中的可能性。

缓冲区溢出



通过堆栈返回指令执行代码

6.4 诱骗类威胁

- 指攻击者利用社会工程学的思想，利用人的弱点（如人的本能反应、好奇心、信任、贪便宜等）通过网络散布虚假信息，诱使受害者上当受骗，而达到攻击者目的的一种网络攻击行为。
 - 准确地说，社会工程学不是一门科学，而是一门艺术和窍门，它利用人的弱点，以顺从你的意愿、满足你的欲望的方式，让你受骗上当。

6.4.1 网络钓鱼Phishing

- 指攻击者通过伪造以假乱真的网站和发送诱惑受害者按攻击者意图执行某些操作的电子邮件等方法，使得受害者“自愿”交出重要信息（例如银行账户和密码）的手段。
 - Phishing是英单词Fishing（钓鱼）和Phone（电话，因为黑客起初以电话作案）的综合体，所以被称为网络钓鱼。

电子邮件诱骗

- 电子邮件服务是合法的**Internet**经典服务，攻击者进行电子邮件诱骗，一般需要经过以下几个步骤。

第一步 选定目标用户群。

第二步 构造欺骗性电子邮件。。

第三步 搭建欺骗性网站。。

第四步 群发邮件，等待上当的受害者。

假冒网站

- 建立假冒网站，骗取用户帐号、密码实施盗窃，这是对用户造成经济损失最大的恶劣手段。
- 为了迷惑用户，攻击者有意把网站域名注册成与真实机构的域名很相似，
 - 网址为“<http://www.1cbc.com.cn>”，而真正银行网站是“<http://www.icbc.com.cn>”，

虚假的电子商务

- 攻击者建立电子商务网站，或是在比较知名、大型电子商务网站上发布虚假的商品销售信息。
- 网上交易多是异地交易，通常需要汇款。
 - 不法分子一般要求消费者先付部分款，再以各种理由诱骗消费者付余款或者其他各种名目的款项，得到钱款或被识破时，犯罪分子就销声匿迹。

6.4.2 对于诱骗类威胁的防范

- 诱骗类威胁不属于传统信息安全的范畴，传统信息安全办法解决不了非传统信息安全的威胁。
 - 一般认为，解决非传统信息安全威胁需要运用社会工程学来反制。
 - 防范诱骗类威胁的首要方法是加强安全防范意识，多问“为什么”，减少“天上掉馅饼”的心理，那么绝大多数此类诱骗行为都不能得逞。
- 另外，用户还应该注意以下几点：
 - 确认对方身份
 - 慎重对待个人信息
 - 谨防电子邮件泄密
 - 注意网站的URL地址



Any question?