计算机网络安全技术

- 课程代号：40240572
- 课程对象：本科生
- 授课教师：尹　霞
- 开课单位：计算机系网络所

1911–2021
清華大學 110周年校庆
110th ANNIVERSARY
TSINGHUA UNIVERSITY

软件入侵基本原理

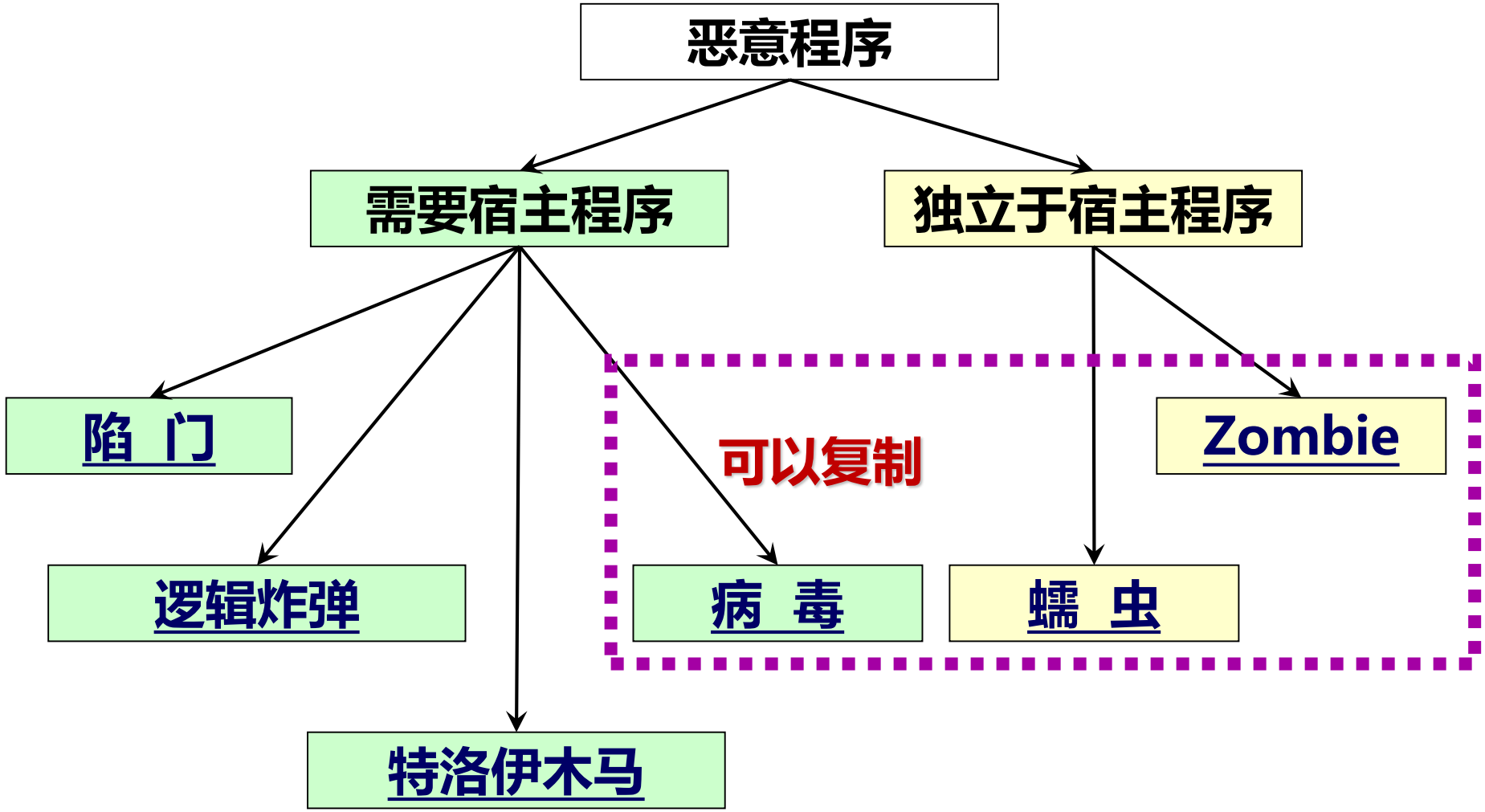
恶意软件

- 软件入侵就是一些恶意软件，利用计算机系统的弱点进行攻击的恶意程序
- 这些恶意程序可以分成两类：依赖于宿主程序的和独立于宿主程序的威胁，也可以按照是否进行复制进行区分
  - 依赖于宿主程序的恶意软件：是在宿主程序被调用来执行某一特定功能时被计划的程序段
  - 独立于宿主程序的恶意软件：是一个程序段（病毒）或者一个独立的程序（蠕虫、细菌），当它执行的时候，可能会对自身进行复制，这些复制品将会在本系统或者其它系统中被激活

病毒、蠕虫和木马

特性	病毒	蠕虫	木马
宿主	需要	不需要	需要
表现形式	不以文件形式存在	独立的文件	伪装成其他文件
传播方式	依赖宿主文件或介质	自主传播	依靠用户主动传播
主要危害	破坏数据完整性、系统完整性	侵占资源	留下后门窃取信息
传播速度	快	极快	慢

常见的恶意程序类型



恶意软件：陷门

陷门（后门）

- 陷门是一个程序的秘密入口
  - 通过这个秘密入口，用户可以不按照通常的访问步骤获得访问权
  - 要控制陷门很困难，解决办法是程序的开发和软件的更新
- 一直以来，程序员们都合理地利用陷门进行程序调试和测试；但当陷门被一些无耻之徒用来作为获得未授权的访问权的手段时，就成为一种威胁
- 从功能上，陷门可以分成三个级别：
  - 系统级：在系统层能访问数据和进程
  - 应用级：逃逸安全机制的合法程序
  - 密码级：能够以某种方式看懂密文

陷门的分类

- 从控制方法上，陷门分成以下几类：
  - 本地权限的提升
    - 对系统有访问权的攻击者变换其权限等级成为管理员，然后攻击者可以重新设置该系统或访问人和存储在系统中的文件
  - 单个命令的远程执行
    - 攻击者可向目标计算机发送消息；每次执行一个单独的命令，陷门执行攻击者的命令并将输出返回给攻击者
  - 远程命令行解释器访问
    - 如远程**Shell**，这类陷门允许攻击者通过网络快速直接地键入受害计算机的命令提示；其比“单个命令的远程执行”要强大得多
  - 远程控制GUI
    - 攻击者可以通过网络看到目标计算机的**GUI**，控制鼠标和键盘的操作

陷门的安装和案例

- 陷门的安装渠道
  - 自己植入：物理接触或被入侵
  - 漏洞：病毒、蠕虫、恶意移动代码
  - 社会工程（欺骗受害者自己安装）：Email、远程共享、BT下载等
- 著名的案例
  - 2001年，Borland的数据库软件Interbase发现隐藏了七年的陷门，程序中包含一个万能用户名和密码：
    - **ID: politically;　PW: correct**
  - 该软件大客户包括波音、诺基亚、摩托罗拉和波士顿股票交易所、Sun等

恶意软件：逻辑炸弹

逻辑炸弹

- 逻辑炸弹是最早出现的恶意软件之一，它是嵌在合法程序中的，只有当待定的事件出现时才会进行破坏的一组程序代码，一般会预先规定了病毒和蠕虫的发作时间

- 逻辑炸弹的典型例子是Tim Lioyd事件
  - Tim Lioyd设计的逻辑炸弹使得他所在的公司Omega Enginnering至少蒙受了1000万美元的损失，严重阻碍了公司发展，并致使80位员工失业
  - Tim Lioyd被处以41个月的监禁和200万美元的赔偿金

恶意软件：特洛伊木马

特洛伊木马

- 特洛伊木马是一种实际上或者表面上看起来有某种有用功能的程序，但它内部含有隐蔽代码；当被调用时会产生一些意想不到的后果
- 特洛伊木马会欺骗用户或者系统管理员进行安装，并在计算机上与其它正常程序一起混合运行，将自己伪装得看起来是个正常程序在工作
- 如果一个程序仅仅提供远程访问，那么它只是一个后门；如果攻击者将这些后门功能伪装成某个良性程序，那么就具有了特洛伊木马属性

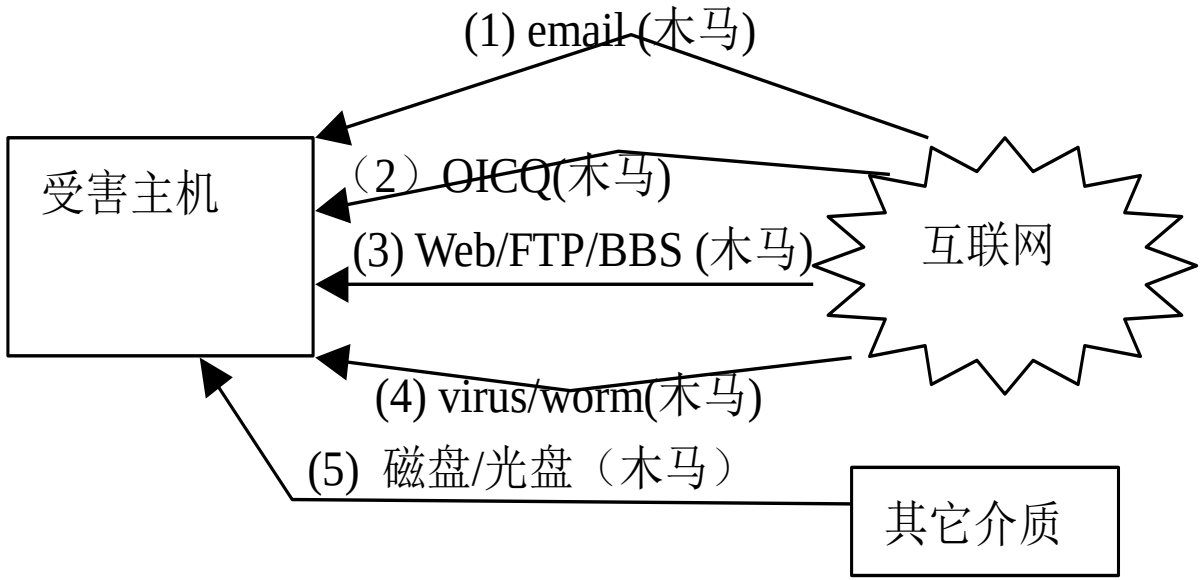
特洛伊木马

- 特洛伊木马使潜伏者执行非授权的功能
  - 例如：利用已经修改的编译程序向正在被编译的程序中插入一些附加代码，这样的特洛伊木马就很难检测出来
  - 例如：特洛伊木马对数据破坏，看起来在执行某个功能，实际在删除用户的文件，从而造成难以恢复的破坏

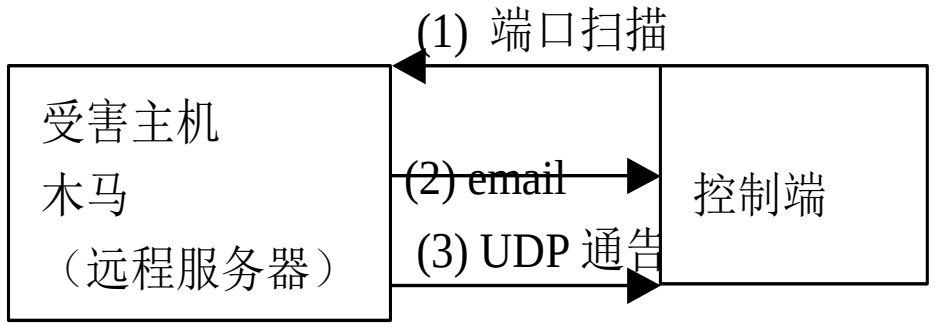
木马软件的构成

- 木马系统软件一般由木马配置程序、控制端程序和木马程序(服务器程序)等三部分组成
- 木马程序，也称服务器程序
  - 驻留在受害者的系统中，非法获取其操作权限，负责接收控制端指令，并根据指令或配置发送数据给控制端
- 木马配置程序
  - 设置木马程序的端口号、触发条件、名称等，使其在服务端藏得更隐蔽，有时该功能被集成在控制端程序内，不单独作为一个程序
- 控制端程序
  - 控制远程服务器，有些集成了木马配置的功能

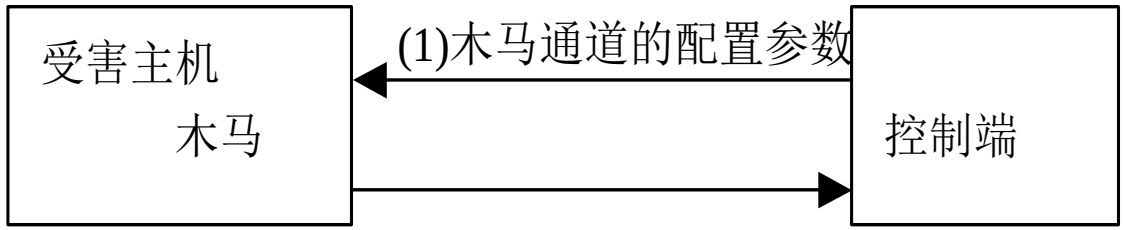
木马的工作原理



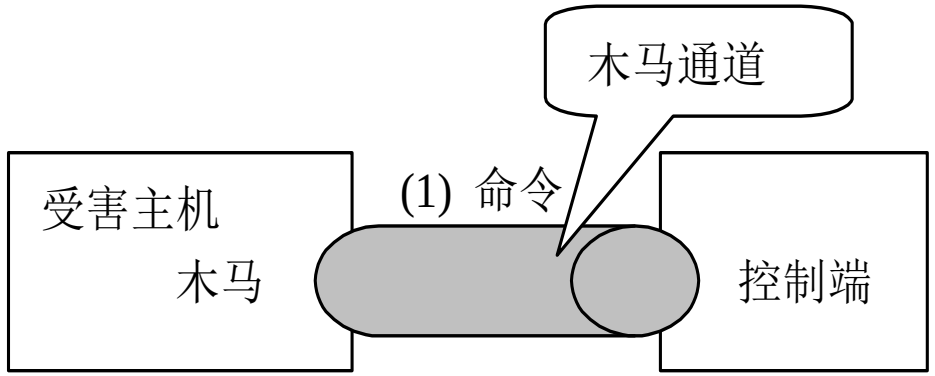
（a） 中木马的途径



（b） 木马与控制端的首次握手



（c） 建立木马通道



（d） 控制端与木马的交互

木马软件的常用手段

- 名字欺骗：修改、模仿文件名以欺骗用户
  - Beauty.jpg.exe
  - httpd,iexplore,notepad,ups,svchost…
- 文件捆绑：恶意程序与正常程序捆绑
- 软件下载，软件本身带毒
- html传播
  - 网页病毒的传播方式
  - Flash传播网页、email传播网页
  - Exe2bmp
  - 正常网页中携带

恶意软件：Zombie

Zombie

- Zombie秘密地接管其它依附在Internet上的计算机，并使该计算机发动攻击，而且这种攻击很难通过追踪Zombie的创建者查出来

- Zombie一般被用在拒绝服务攻击上，尤其是对Web站点的攻击；它被放置在成百上千的、属于第三方的计算机中，通过向Internet发送不可抵抗的攻击取得对目标Web站点的控制

恶意软件：病毒

病毒原理、引导性病毒

病毒

- 病毒是一种可以通过修改自身来感染其它程序的程序，这种修改包括对病毒程序的复制，复制后生成的新病毒同样具有感染其他程序的功能
- 通过寄居在宿主程序上，计算机病毒可以暂时控制该计算机的操作系统盘
- 没有感染的软件一旦在带毒计算机上运行，就会在新程序中产生病毒拷贝，病毒就会通过U盘或者网络传播

病毒的生命周期

- 潜伏阶段：
  - 这个阶段病毒处于休眠状态，最终会被某些条件激活；但并不是所有的病毒都会经历此阶段
- 传染阶段：
  - 病毒程序将自身复制到其他程序或者磁盘的某个区域上，每个被感染的程序又因此包含了病毒的复制品，从而也就进入了传染阶段
- 触发阶段：
  - 病毒在被激活后，会执行某个特定的功能从而达到某种既定的目的
- 发作阶段：
  - 病毒在触发条件成熟的时候，即可在系统中发作

病毒的破坏程度

- 病毒发作体现出来的破坏程度是不同的，有些是无害的，例如屏幕上的一些干扰信息；有些会带来巨大的危害，例如破坏程序或者数据

- 多数病毒都是基于某种特定的方式进行工作的，例如某个操作系统或者某个特定的硬件平台；因此攻击者经常利用系统的细节和弱点来设计病毒程序

病毒的种类

- 寄生性病毒
  - 将自身依附在可执行文件上并对自身进行复制；当受感染文件被执行后，它又会继续寻找其他的可执行文件并对其进行感染
- 引导扇区病毒
  - 此类病毒感染主引导记录或者引导记录，在系统从含有病毒的磁盘上引导装入程序时进行传播
- 常驻存储器病毒
  - 该病毒以常驻系统的程序的形式寄居在主存储器上面，它会感染所有的文件
- 隐蔽性病毒

引导型病毒：概述

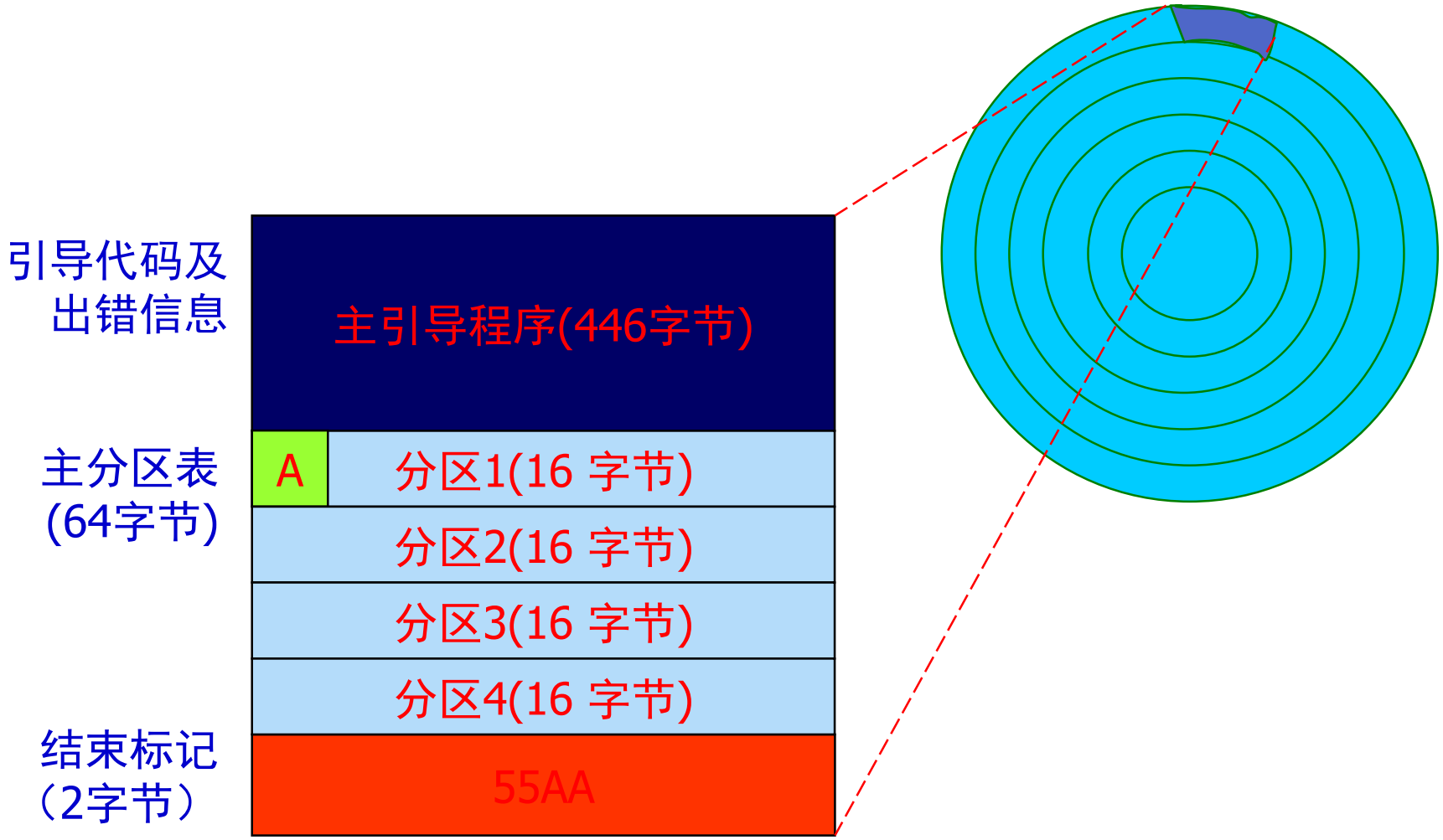
- 当系统引导时，BIOS只是机械地将引导扇区中的内容读入内存，从而使病毒程序首先获得对系统的控制权
  - 病毒将其代码安装到内存的高端驻留，系统内存总量减少若干KB
- 引导型病毒的传播对象
  - 软盘的引导扇区、硬盘的主引导扇区、硬盘分区的引导扇区
- 引导型病毒的破坏方式变化多端
  - 格式化磁盘：“磁盘杀手病毒”
  - 破坏目录区：“大麻病毒”、“磁盘杀手病毒”
  - 破坏系统和外设的连接：“2708”病毒

引导型病毒：传播

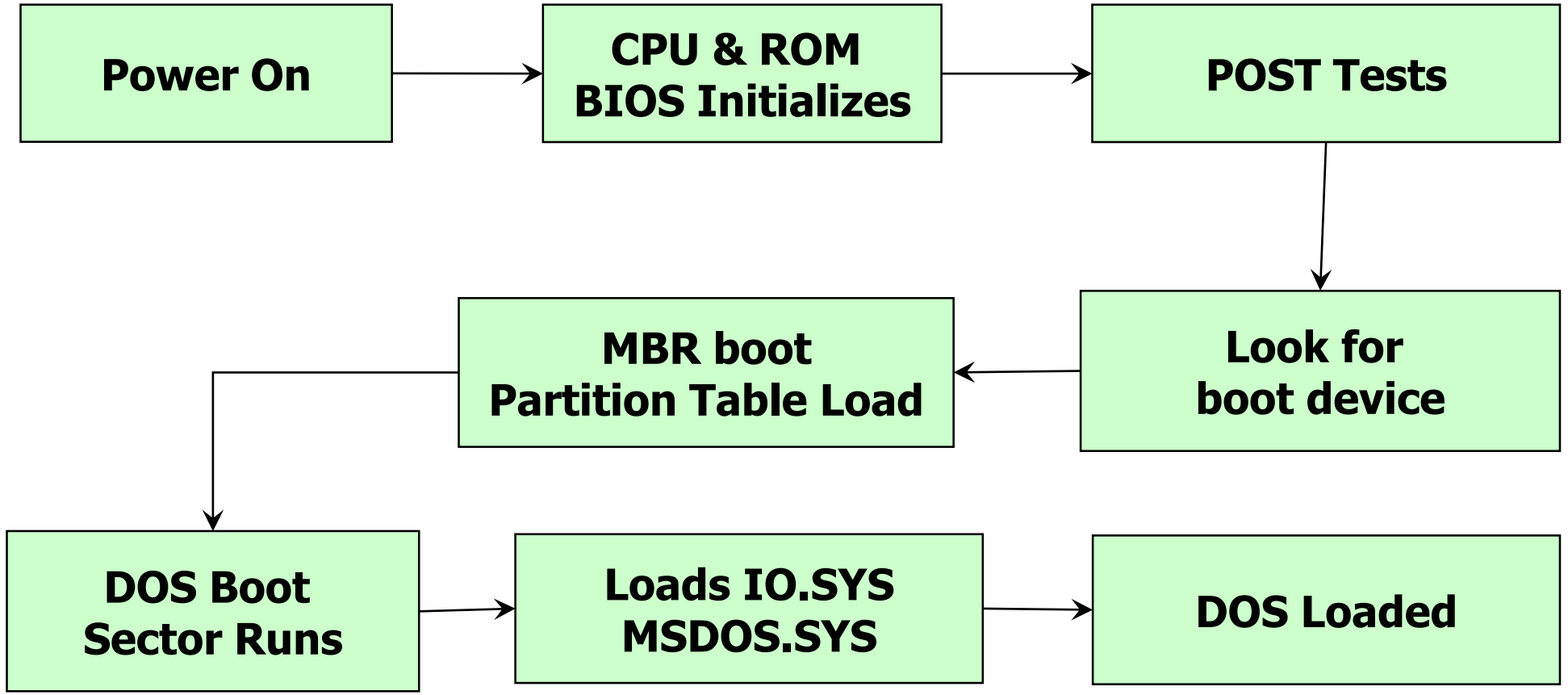
- 引导型病毒为了实现向外进行传播和破坏等作用，一般要修改系统的中断向量，使之指向病毒程序相应服务部分；从而可将病毒程序由静态转变为动态，具有随时向外进行传播和对系统进行破坏的能力
- 通过截获系统的中断向量实现
  - 病毒在引导时，已修改中断向量，使之指向病毒程序的传播部分
  - 若系统有对中断的请求，均转到病毒程序的传播部分
  - 病毒程序的传播部分将操作磁盘的部分内容读入内存，再判断其特定位置上是否有此病毒的特征代码，若无则进行传染，否则放弃传染，转去执行系统的中断请求

引导型病毒：引导记录

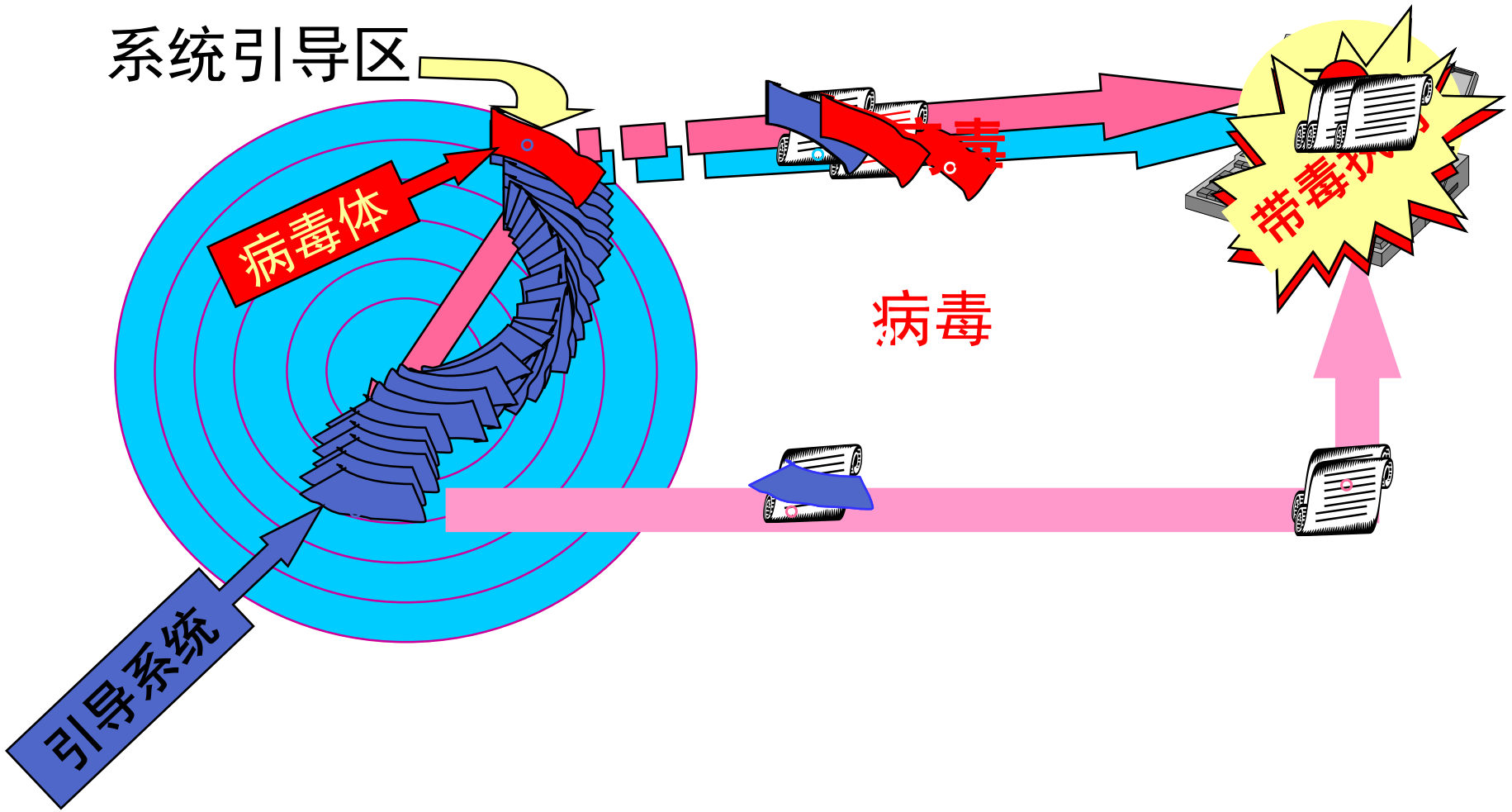
- 主引导记录（MBR）



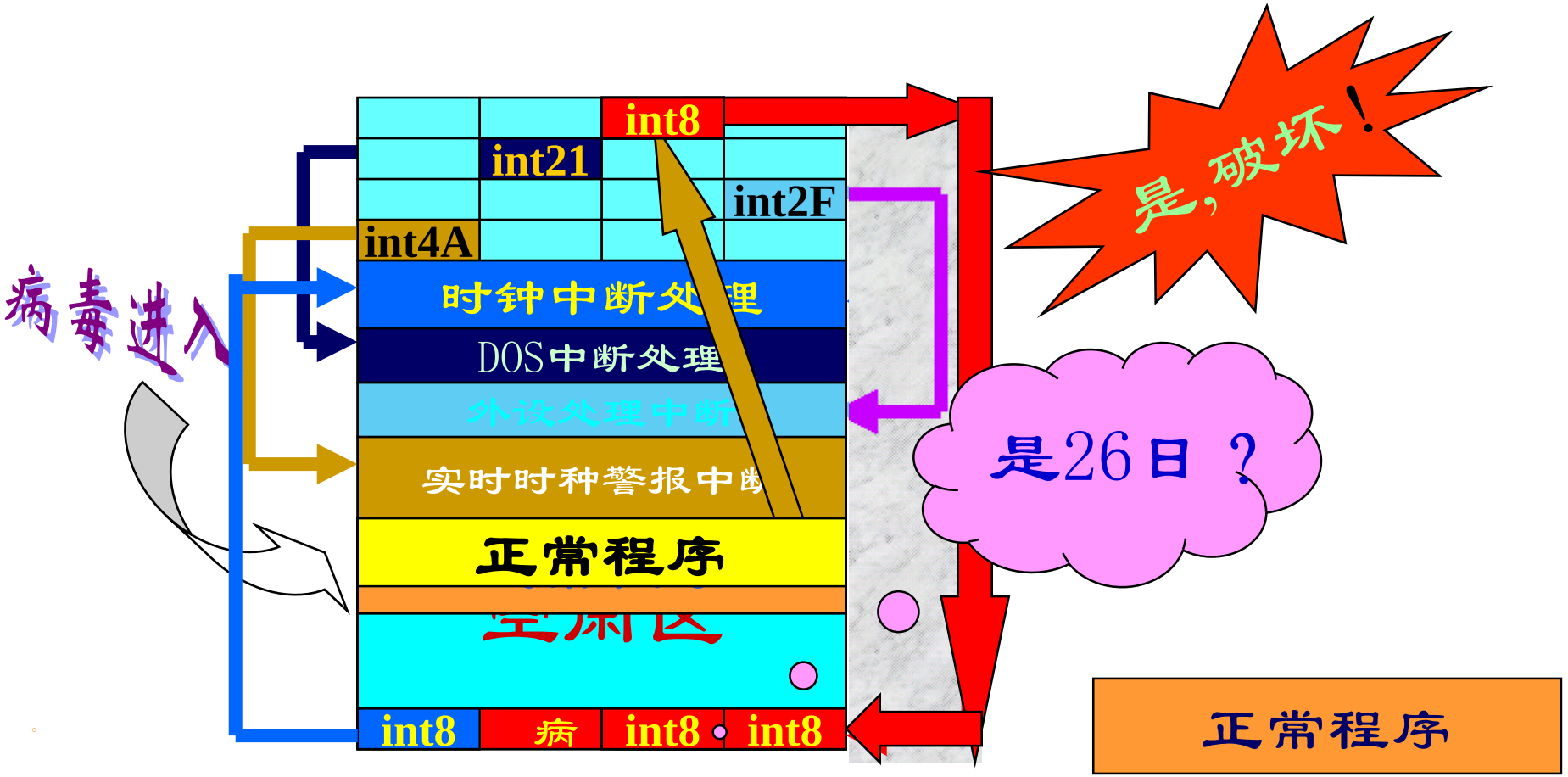
引导型病毒：系统引导过程(DOS)



引导型病毒：感染与执行过程



病毒的激活过程

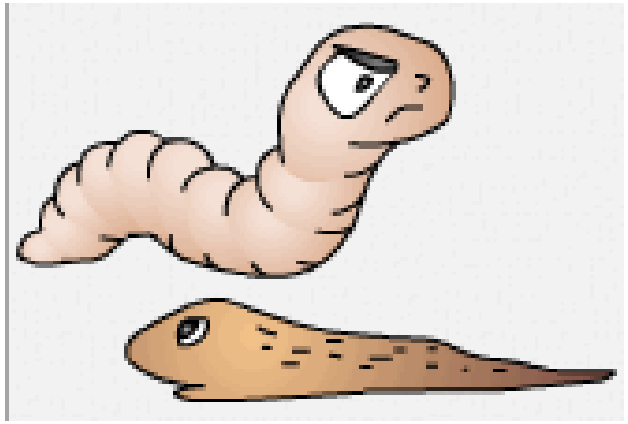

病毒进入
int8
int21
int2F
int4A
时钟中断处理
DOS中断处理
外设处理中断
实时时种警报中断
正常程序
int8
病
int8
int8
是，破坏！
是26日？
正常程序

恶意软件：蠕虫

蠕虫原理、Morris 蠕虫

什么是蠕虫Worm

- 美国传统词典
  - A program that replicates itself and interferes with software function or destroys stored information.



- 一种能够自我复制的计算机程序，它利用网络上计算机系统的漏洞自主的将自己复制到其它计算机上
  - 一个独立的计算机程序，不需要宿主
  - 自我复制，自主传播
  - 占用系统或网络资源、破坏其他程序
  - 不伪装成其他程序，靠自主传播

蠕虫

- 蠕虫会时刻寻找更多的计算机并伺机传染，那些被感染的机器又会作为自动发射缓冲器，进一步感染其它机器
- 只要系统中的蠕虫处于活动状态，就会造成巨大的破坏
- 蠕虫的传播造成网络拥塞，还会对计算机系统和信息安全造成破坏，例如：
  - CodeRed蠕虫在爆发后的9小时内就攻击了25万台计算机，造成的损失估计超过20亿美元
  - Nimda造成的损失评估数据从5亿美元攀升到26亿美元后，继续攀升，到现在已无法估计

网络蠕虫

- 网络蠕虫利用以下网络工具进行传播：
  - 电子邮件设备：蠕虫将自身的拷贝以邮件的形式发送到其它系统
  - 远程执行功能：蠕虫在其它系统执行自身的拷贝
  - 远程登录功能：蠕虫以用户身份进入远程系统，然后使用命令将自身从一台计算机复制到另外一台计算机

蠕虫的早期历史

- 1980年 Xerox PRAC
- 1988年11月2日 Morris Worm
- 1989年10月16日 WANK Worm
- 1998年5月 ADM Worm
- 1999年9月 Millennium
- 2001年1月 Ramen Worm
- 2001年3月23日 Lion Worm
- 2001年4月3日 Adore Worm
- 2001年5月 Cheese Worm
- 2001年5月 Sadmind/IIS Worm
- 2001年7月19日 CodeRed Worm
  - 催生国家级网络安全应急组织
- 2001年 Nimda Worm
- 2002年 Slapper Worm
- 2003年 冲击波蠕虫和Blast清除者蠕虫
- 2003年 口令蠕虫
- 2003年 SQL SLAMMER蠕虫
- 2004年 震荡波蠕虫
- 2005年 黛蛇蠕虫
- 2006年 “魔波”蠕虫
- 2007年 “Nimaya(熊猫烧香)”病毒
- 2008年 “机器狗”病毒
- 2009年 “飞客”蠕虫

。。。。。。。。。。

Morris Worm的诞生

- 1988年11月3日，当数千名计算机系统管理员像往日一样轻松自在地赶到工作岗位时，他们惊呆了：
  系统瘫痪！原因不明！

- 当他们知道类似系统瘫痪的机器有6000多台时，所有人都怀疑这是一次阴谋！互联网络受到了攻击，十分之一的机器完全瘫痪，其它系统也是缓慢如牛
- 在人们正对互连网络充满憧憬之时，这样一次事件无异于迎面兜盆冷水，让人充满恐惧

Morris Worm的诞生

- 调查很快展开，结果更出人意外：是美国国家安全局一位专家的儿子Robert Morris干的！他还只是康奈尔大学一个二十出头的研究生

- Robert Morris 编的这个程序叫Worm，只有99行；严格地说它不是病毒，因为病毒需要一个宿主程序，而Worm程序则是自我复制，自行传播，更具危害性

Morris Worm的诞生

- 刚开始Robert Morris并不是第一嫌疑犯
  - 因为他是一个好学生，没有前科；更重要的一点是他的父亲曾经对Internet的设计有过意义深远的影响
- Robert Morris很快发现Worm程序以一种比他的预计快得多的速度正在进行自我复制和感染网络中的机器，原因是：程序中的计数器设计有缺陷
- 认识到事情严重性后，他与哈佛的朋友联系并讨论解决的办法；最后，他们从哈佛向整个网络发了一封匿名信，指导程序员们如何杀死Worm程序以及如何防止被感染

Morris Worm的诞生

- Robert Morris为什么编制这个程序？
  - Robert Morris原本只想通过worm程序，估计当时的互联网规模
  - 他在程序中设计了计数功能，可以控制“蠕虫”在同一台主机上的繁殖次数，但是这位天才编制的计数器竟然有一处小小的疏忽
  - 本来设定只繁殖一次，可结果是“蠕虫”爬遍互联网络，过度繁殖耗尽了众多主机的系统资源，基本“当”掉了整个Internet，这一切就发生在1988年11月2日晚上
  - 莫里斯释放了这条“小虫”后回到宿舍安然入睡的几个小时之内，就引发了一场对当时约有6万台电脑的互联网的毁灭性攻击
  - 他让人们彻底认识到了Internet的弱不禁风！

Morris Worm的诞生

- Morris蠕虫所造成的损失是巨大的，直接经济损失上百亿美元，仅就上万名工程师夜以继日地恢复工作来计算就是一笔不小的支出

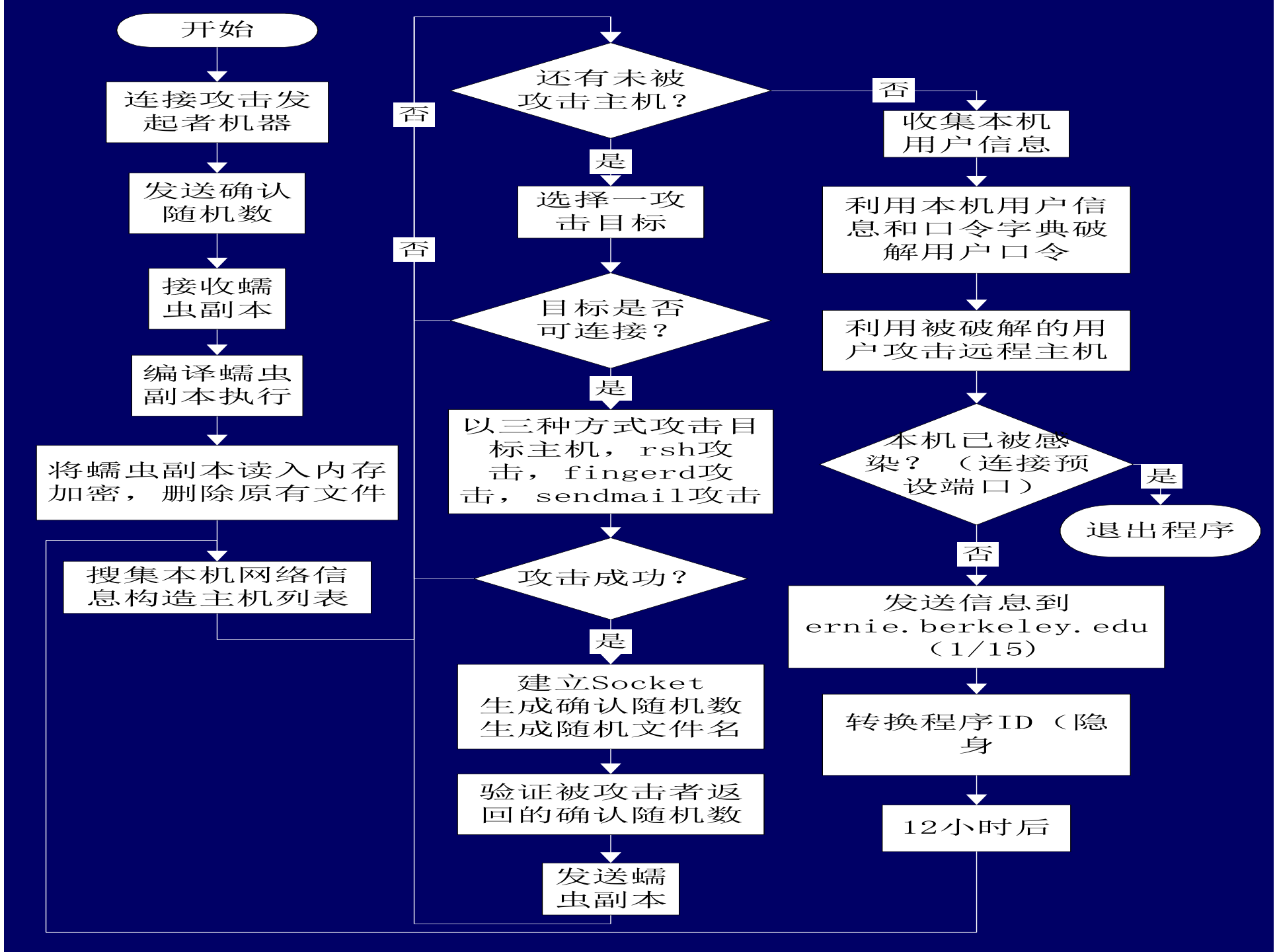
- 1990年， Robert Morris成了首位依据美国反黑客法律《计算机欺诈和滥用法》被判决的人
  - 专家们分析后得出结论： Robert Morris确实不是故意的，据此从轻发落：罚款一万美元，三年徒刑，400小时社区劳动

蠕虫之父?

- 真正的“蠕虫之父”是John Shoch， Robert Morris是完成了一次代价不菲的演习
  - John Shoch当时设计了一个程序：网络空闲时，该程序就在各计算机之间“游荡”而不会造成任何破坏；当有机器负荷过重时，该程序可以从空闲计算机“借取资源”，达到整个网络的负载平衡
  - 莫里斯的蠕虫程序就是受此启发而设计的，只不过是将“借取空闲资源”改为“耗尽所有资源”
- 当莫里斯事件发生时，JohnShoch说：“我们创造出了蠕虫，但我们还没有学会很好地控制它。”
- 那一年，美国总统里根因此而签署了《计算机安全法令》

Morris 蠕虫主要攻击方法

- 主要的攻击方法
  - Rsh，rexec：用户的缺省认证
  - Sendmail 的debug模式
  - Finger的缓冲区溢出
  - 口令猜测


开始
连接攻击发起者机器
发送确认随机数
接收蠕虫副本
编译蠕虫副本执行
将蠕虫副本读入内存加密，删除原有文件
搜集本机网络信息构造主机列表
还有未被攻击主机？
否
是
选择一攻击目标
否
目标是否可连接？
是
以三种方式攻击目标主机，rsh攻击，fingerd攻击，sendmail攻击
攻击成功？
是
建立Socket
生成确认随机数
生成随机文件名
验证被攻击者返回的确认随机数
发送蠕虫副本
否
收集本机用户信息
利用本机用户信息和口令字典破解用户口令
利用被破解的用户攻击远程主机
本机已被感染？（连接预设端口）
是
退出程序
否
发送信息到ernie.berkeley.edu（1/15）
转换程序ID（隐身
12小时后

Morris 蠕虫

计算机网络安全技术

Activity is the only road to knowledge

Computer Network Security @ 2020Fall