网 络 安 全

——恶意代码

杭州师范大学信息科学与技术学院

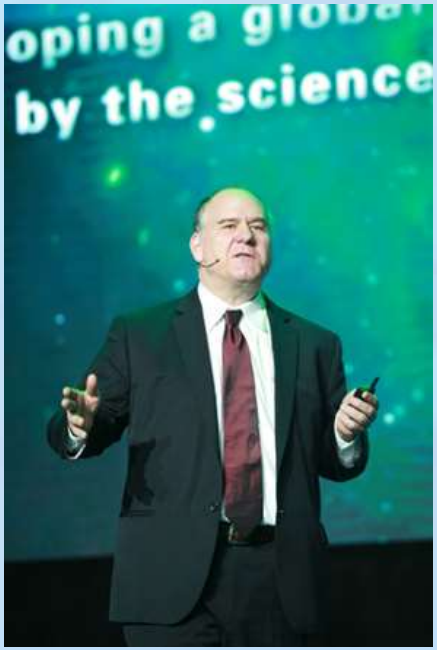
刘雪娇　邮箱：liuxuejiao0406@163.com

本章节要求

- 了解恶意代码的基本原理和发展趋势
- 了解和掌握恶意代码的分类和攻击机制
- 掌握病毒、木马和蠕虫的相关知识和区别与联系
- 认识检测和防范恶意代码的常用方法和相关技术

恶意代码

8.1 引言



信息安全技术是门理性的艺术

——弗雷德·科恩

摇篮时代：在恶意代码的诞生之初，其实并非就是一个搞破坏的“坏孩子”。其最初的概念是由电脑的先驱者冯•诺依曼在1949年发表的论文《复杂自动机组织论》中提出的：一种能够实现复制自身的自动机。

虽然有“计算机病毒之父”弗雷德•科恩给计算机病毒正名：“计算机病毒不是利用操作系统的错误或缺陷的程序，它是正常的用户程序。”但人们总是能想尽办法找到其妙用所在。技术本无罪，有罪的是人类无尽的自私和贪婪。

恶意代码的影响

研究恶意代码的必要性

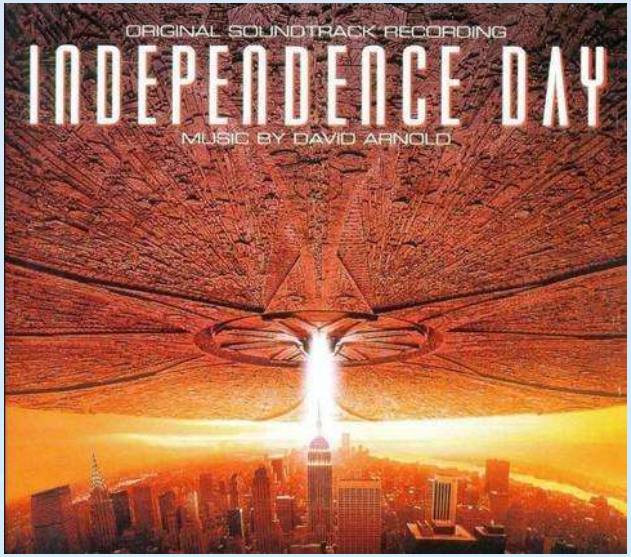
在Internet安全事件中，恶意代码造成的经济损失占有最大的比例。与此同时，恶意代码成为信息战、网络战的重要手段。日益严重的恶意代码问题，不仅使企业及用户蒙受了巨大经济损失，而且使国家的安全面临着严重威胁。

1991年在“海湾战争”中，美军第一次将计算机病毒用于实战，在空袭巴格达的战斗中，成功地破坏了对方的指挥系统，使之瘫痪，保证了战斗的顺利进行，直至最后胜利。

小球病毒是新中国成立以来发现的第一例电脑病毒（1988年），该病毒初版并不会对系统造成较明显的破坏，但在汉字模式下反应非常严重，会使程序无法正常运行，一些当时配置较低的机器会严重拖慢速度。

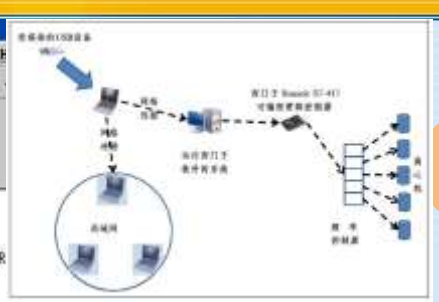
研究恶意代码的必要性

电影《独立日》中，史蒂夫和戈德布拉姆躲在一艘被盗的外星飞船里（就像是一匹木马），假装自己是外星人（正常程序），进入母舰（特洛伊）以便上传病毒，最后给敌人飞船系统注入恶意代码，使敌人飞船的保护层失效，从而拯救了地球。从中可以看出恶意代码研究的重要性。



恶意代码问题已成为信息安全需要解决的、迫在眉睫的、刻不容缓的问题。

恶意代码的发展史



南加州大学的弗雷德•科恩编写了第一个真正具有破坏性的计算机病毒。

1983

“爱虫”病毒及其后出现的50多个变种病毒，仅一年时间共感染了4000多万台计算机

2000

震网（Stuxnet）病毒被首次检测处理，其感染了全球超过 45000个网络

2010

CNCERT监测发现名为“Ramnit”的网页恶意代码被挂载在境内近600个党政机关、企事业单位网站上

2016

WannaCry蠕虫通过MS17-010漏洞在全球范围大爆发，这是一种“蠕虫式”的勒索病毒软件。

2017

恶意代码经过20多年的发展，破坏性、种类和感染性都得到了增强，对人们日常生活影响越来越大。

恶意代码的存在原因

恶意代码长期存在的原因

- 计算机技术飞速发展的同时并未使系统的安全性得到增强。计算机技术进步带来的安全增强能力最多只能弥补由应用环境的复杂性带来的安全威胁的增长程度。不但如此，计算机新技术的出现还很有可能使计算机系统的安全变得比以往更加脆弱。

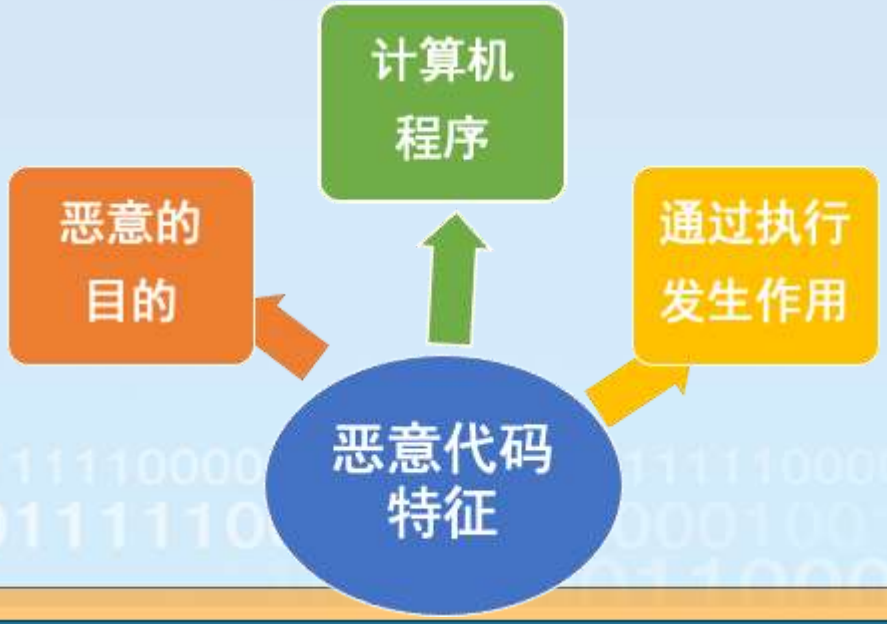
- 恶意代码的一个主要特征是针对性（针对特定的脆弱点），这种针对性充分说明了恶意代码正是利用软件的脆弱性实现其恶意目的的。造成广泛影响的1988年Morris蠕虫事件，就是利用邮件系统的脆弱性作为其入侵的最初突破点的。

恶意代码概述

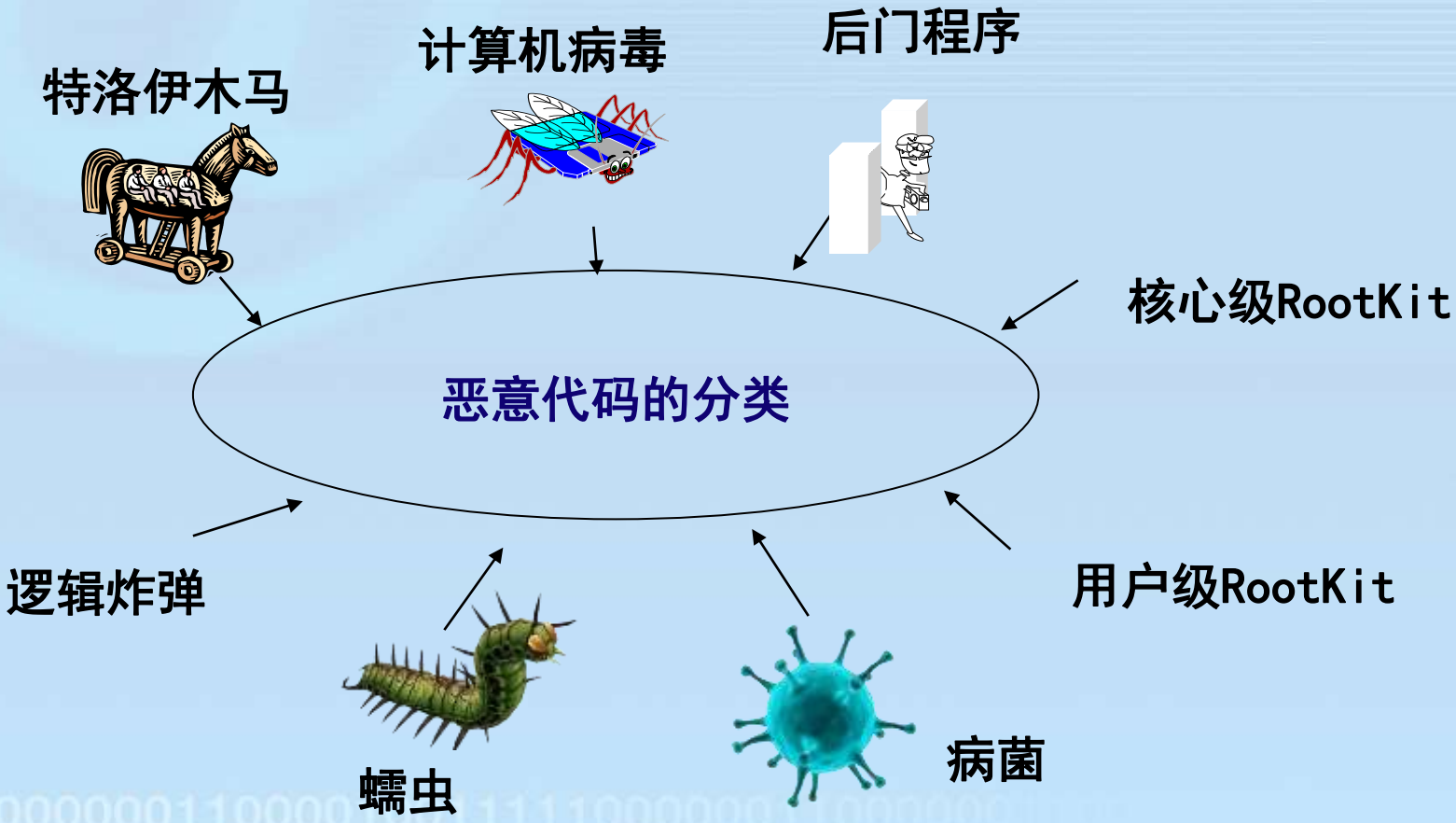
- 恶意代码，Malware，又称恶意软件

Malicious Software

- 是指一种实现某些恶意功能的代码，通常在未明确提示用户或者未经用户授权的情况下，在特定的环境下会被执行，从而破坏计算机系统、网络功能或者计算机数据的保密性、完整性和可用性



8.2 恶意代码的分类与机理

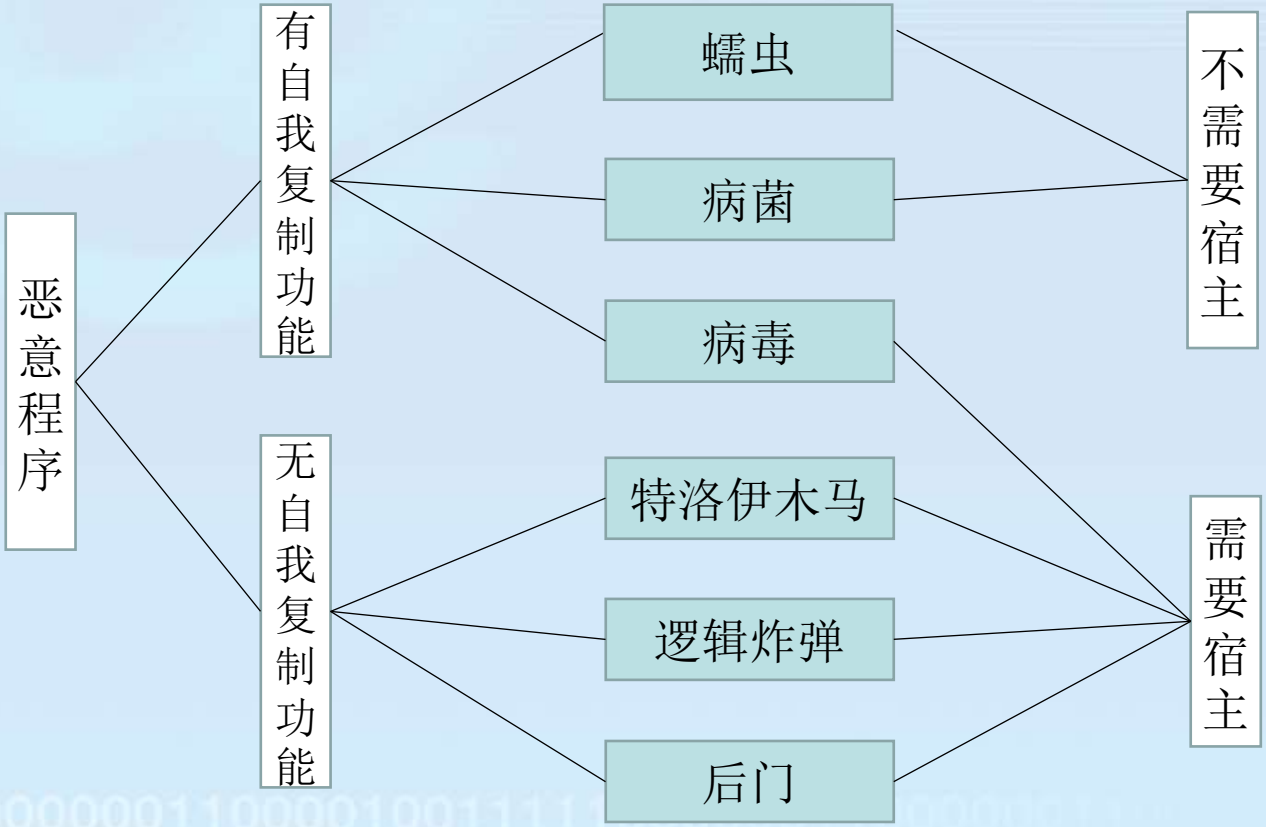


恶意代码

恶意代码类型	定义	特点
计算机病毒	指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能自我复制的一组计算机指令或者程序代码。	潜伏、传染和破坏
蠕虫	指通过计算机网络自我复制，消耗系统资源和网络资源的程序	扫描、攻击和扩散
特洛伊木马	指一种与远程计算机建立连接，使远程计算机能够通过网络控制本地计算机的程序。	欺骗、隐蔽和信息窃取
后门程序	指绕过安全性控制而获取对程序或系统访问权的程序方法	隐蔽、不易查杀
逻辑炸弹	指一段嵌入计算机系统程序的，通过特殊的数据或时间作为条件触发，试图完成一定破坏功能的程序。	潜伏和破坏
病菌	指不依赖于系统软件，能够自我复制和传播，以消耗系统资源为目的的程序。	传染和拒绝服务
用户级RootKit	指通过替代或者修改被系统管理员或普通用户执行的程序进入系统，从而实现隐藏和创建后门的程序。	隐蔽，潜伏
核心级RootKit	指嵌入操作系统内核进行隐藏和创建后门的程序	隐蔽，潜伏

恶意代码的分类

恶意代码所寄生的合法程序被称做载体，也称为宿主程序。



恶意代码的攻击机制

恶意代码的基本作用机制的整个作用过程分为以下6个部分。

➢**侵入系统：**侵入系统是恶意代码实现其恶意目的的必要条件。恶意代码入侵的途径很多，比如，从互联网下载的程序本身就可能含有恶意代码；接收已感染恶意代码的电子邮件；从光盘或软盘往系统上安装软件；黑客或攻击者故意将恶意代码植入系统等。

➢**维持或提升现有特权：**恶意代码的传播与破坏必须盗用用户或者进程的合法权限才能完成。

恶意代码的攻击机制

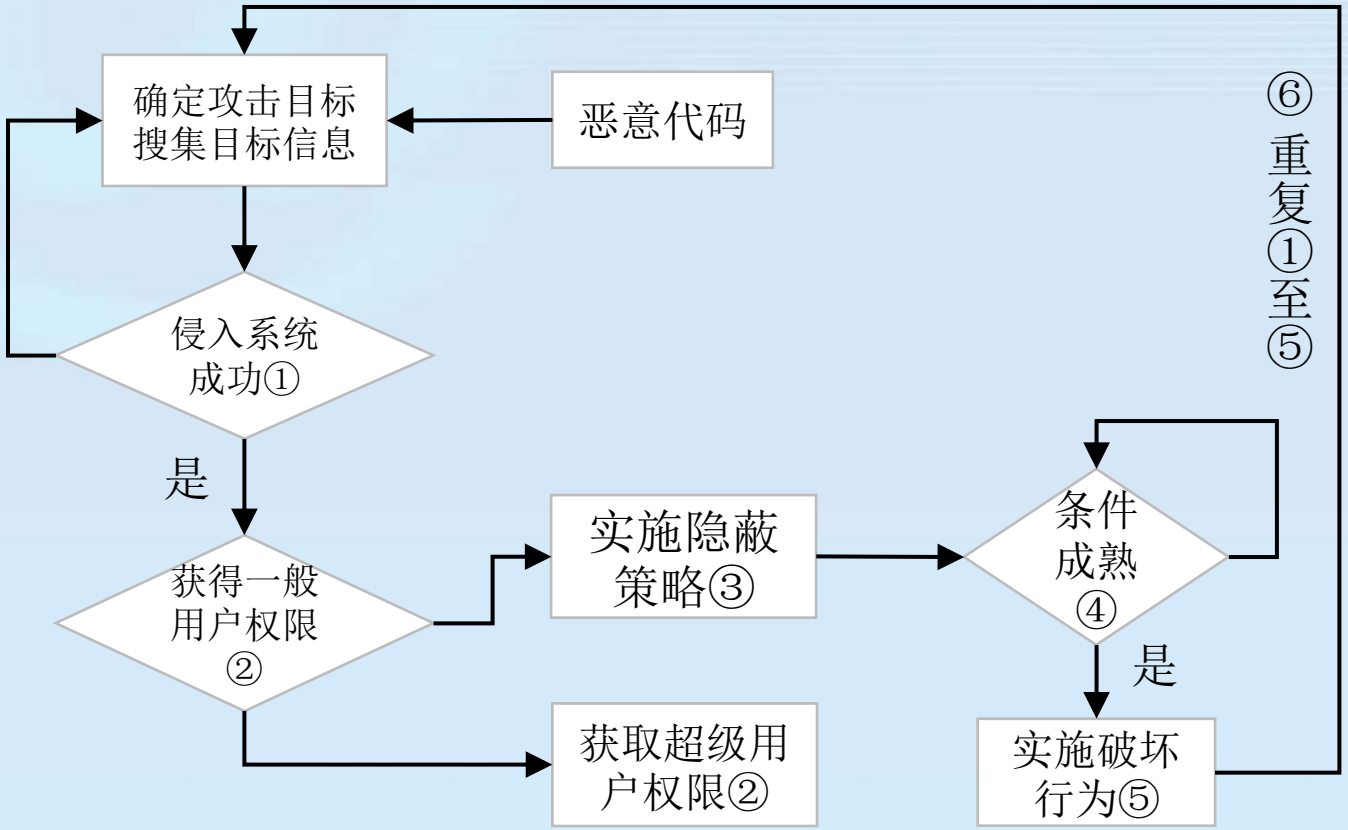
- **隐蔽策略：**为了不让系统发现恶意代码已经侵入系统，恶意代码可能会改名、删除源文件或修改系统安全策略来隐藏自己。

- **潜伏：**恶意代码侵入系统后，等待一定的条件，并具有足够的权限时，就发作并进行破坏活动。

- **破坏：**恶意代码的本质具有破坏性，其目的是造成信息丢失、泄密，破坏系统完整性等。

- **重复：**重复①至⑤对新的目标实施攻击过程。

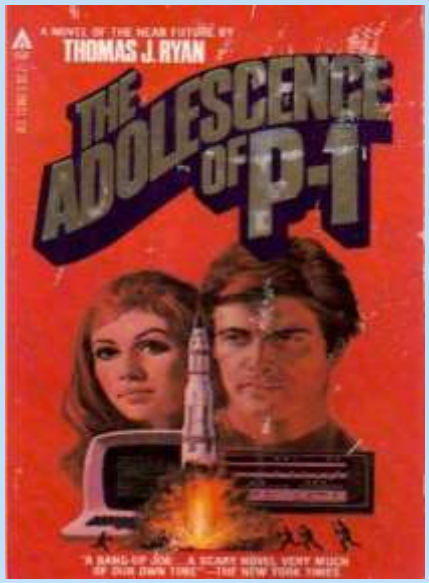
恶意代码的攻击机制



恶意代码攻击模型

8.3 计算机病毒、木马、蠕虫

“计算机病毒”由生物医学上的“病毒”一词借用而来，该词最早出现在美国作家Thomas J.Ryan于1977年出版的科幻小说《The Adolescence of P-1》中



与生物医学上“病毒”的异同：

相同点：

- 能够传送病毒载体，具有传染性
- 有独特的复制能力，具有流行性
- 攻击目标特定的弱点，具有针对性

不同点：

- 不是天然存在的，而是人为编制的具有特殊功能的程序

计算机病毒的定义

广义的定义：凡能够引起计算机故障，破坏计算机数据的程序统称为计算机病毒（Computer Virus）。

1994年2月18日，我国正式颁布实施了《中华人民共和国计算机信息系统安全保护条例》，在《条理》第28条中对计算机病毒的定义为：“指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能够自我复制的一组计算机指令或者程序代码”。

计算机病毒特性

- **传染性：**以各种渠道扩散到未被感染的计算机，是病毒的基本特征，是判别一个程序是否为计算机病毒的首要条件。

- **寄生性：**不以独立的文件形式存在，寄生在合法的程序中。当执行这个程序时，病毒就起破坏作用，而在未启动这个程序之前，它是不易被人发觉的。

- **隐蔽性：**在病毒发作前不被用户察觉，表现为存在的隐蔽性与传染的隐蔽性。

计算机病毒特性

- **潜伏性：**传染后不会立马发作。潜伏性越好，其在系统中的存在时间就会越长，其传染范围就会越大。

- **可触发性：**因某个特征或数值的出现，诱使病毒实施感染或进行攻击。(例如，系统时钟的某个时间或日期、系统运行了某些程序等。一旦条件满足，计算机病毒就会“发作”，使系统遭到破坏。)

- **破坏性：**降低系统工作效率，占用系统资源（取决于病毒的设计目的）。例如：把计算机内的文件删除或不同程度的损坏。破坏引导扇区及BIOS，硬件环境破坏。

计算机病毒功能模块

计算机病毒程序一般包括以下3个功能模块：

- **引导模块**：宿主程序开始工作时将病毒程序从外存引入内存，并使传染和破坏模块处于活动状态，以监视系统运行。
- **传染模块**：负责将病毒传染给其他程序，使病毒向外扩散。分为病毒传染的条件判断部分和病毒传染程序主体部分。
- **破坏模块**：是病毒的核心部分，它体现了病毒制造者的意图。由病毒破坏的条件判断部分和破坏程序主体部分。

病毒破坏性表现

- **占用CPU资源：**额外占用或消耗内存空间，或禁止分配内存、蚕食内存，导致一些大型程序执行受阻，使系统性能下降。
- **干扰系统运行：**不执行命令、干扰内部命令的执行、虚发报警信息打不开文件、内部栈溢出、占用特殊数据区、时钟倒转、重启动等。
- **攻击CMOS：**CMOS是保存系统参数（如系统时钟、磁盘类型、内存容量等）的重要场所。有的病毒（如CIH病毒）可以通过改写CMOS参数，破坏系统硬件的运行。
- **攻击系统数据区：**硬盘的主引导扇区、boot（引导）扇区、FAT（文件分配）表、文件目录等，是系统重要的数据，这些数据一旦受损，将造成相关文件的破坏。

常见计算机病毒

- **网络病毒**

  通过计算机网络传播感染网络中的可执行文件

- **文件病毒**

  感染计算机中的文件（如：COM，EXE，DOC等）

- **引导型病毒**

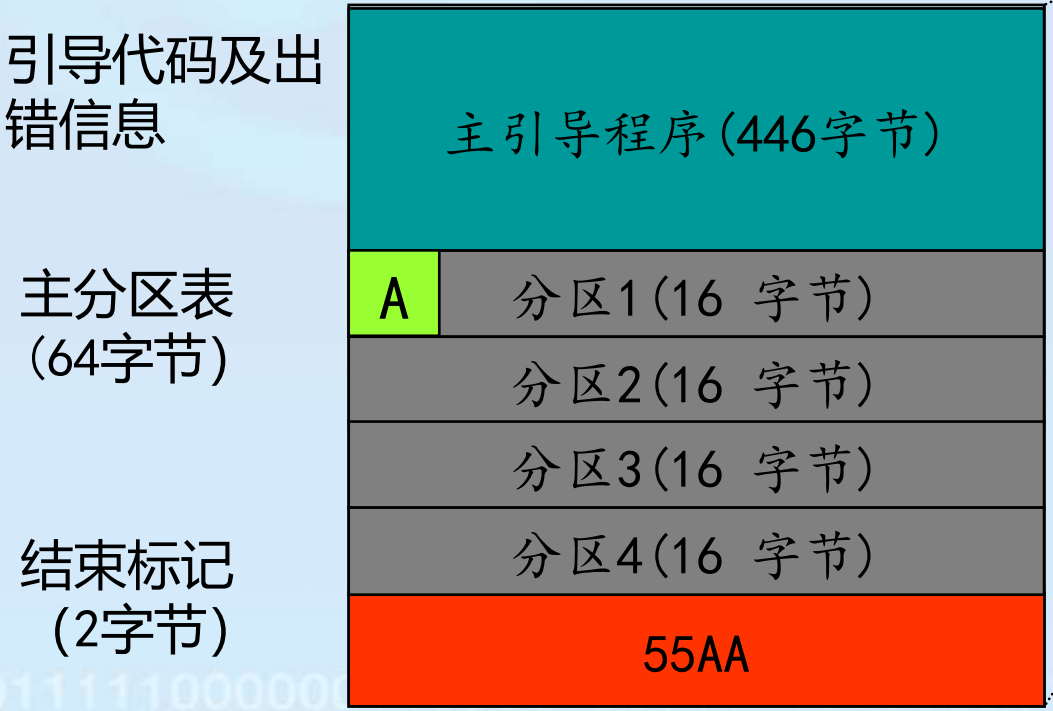
  感染启动扇区（Boot）和硬盘的系统引导扇区（MBR）

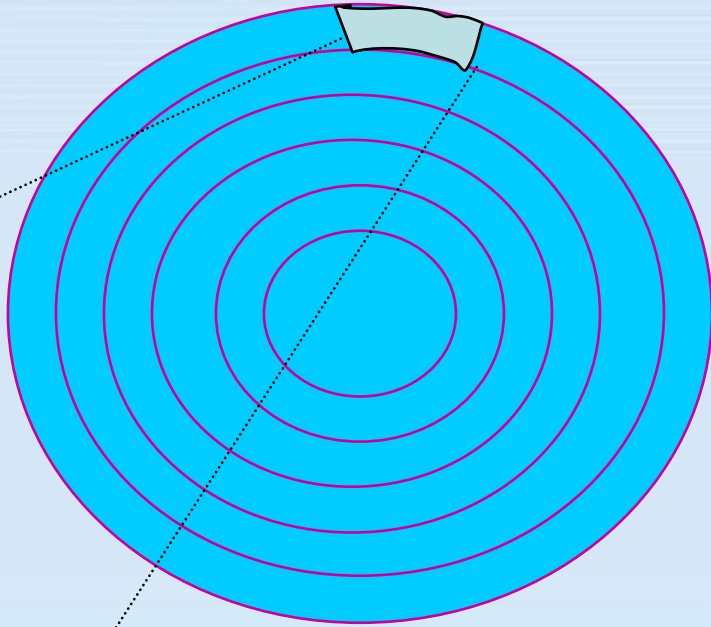
- **混合型**

  多型病毒（文件和引导型）有感染文件和引导扇区两种目标

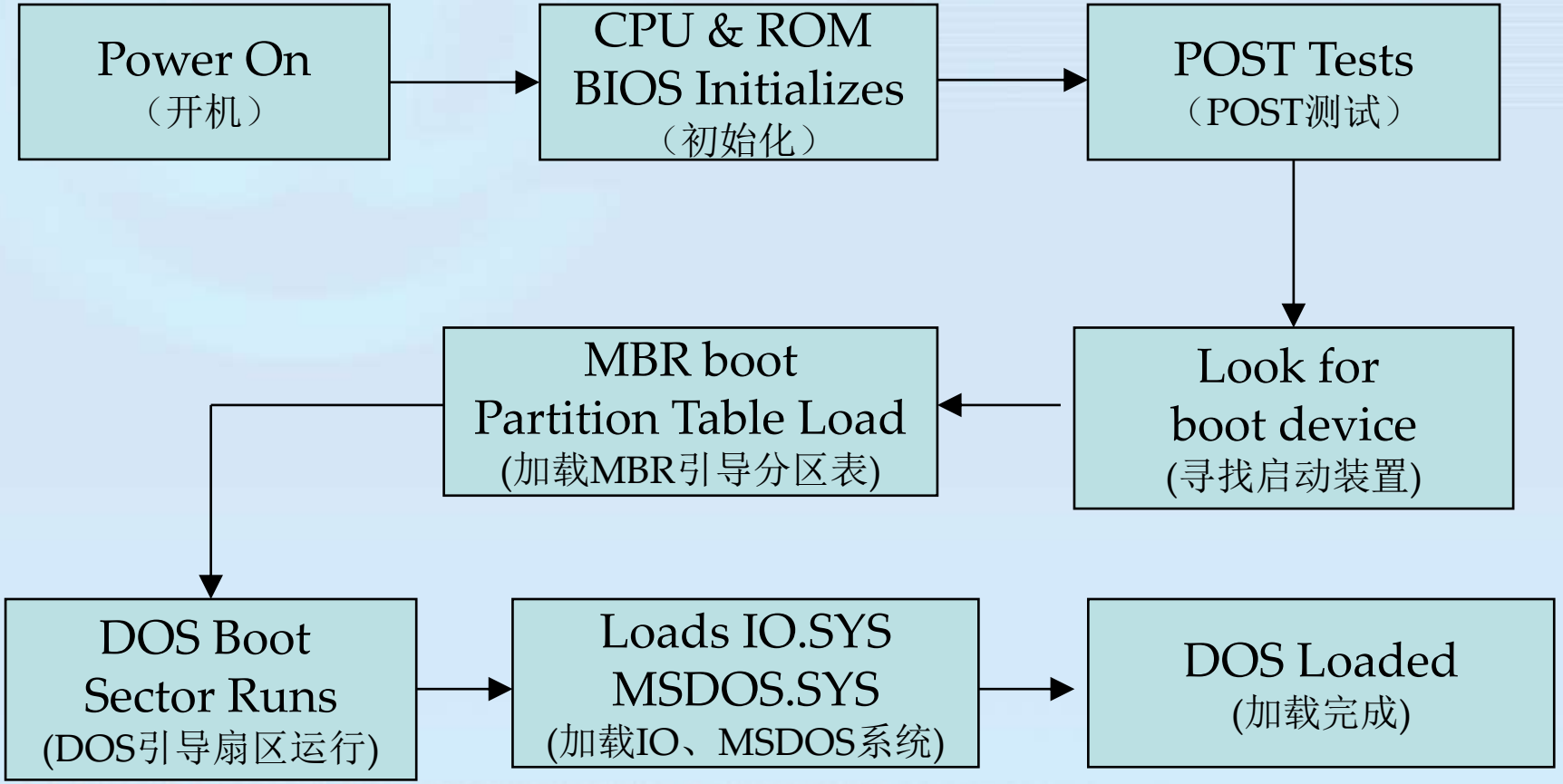
引导型病毒—引导记录

- 主引导记录（MBR）

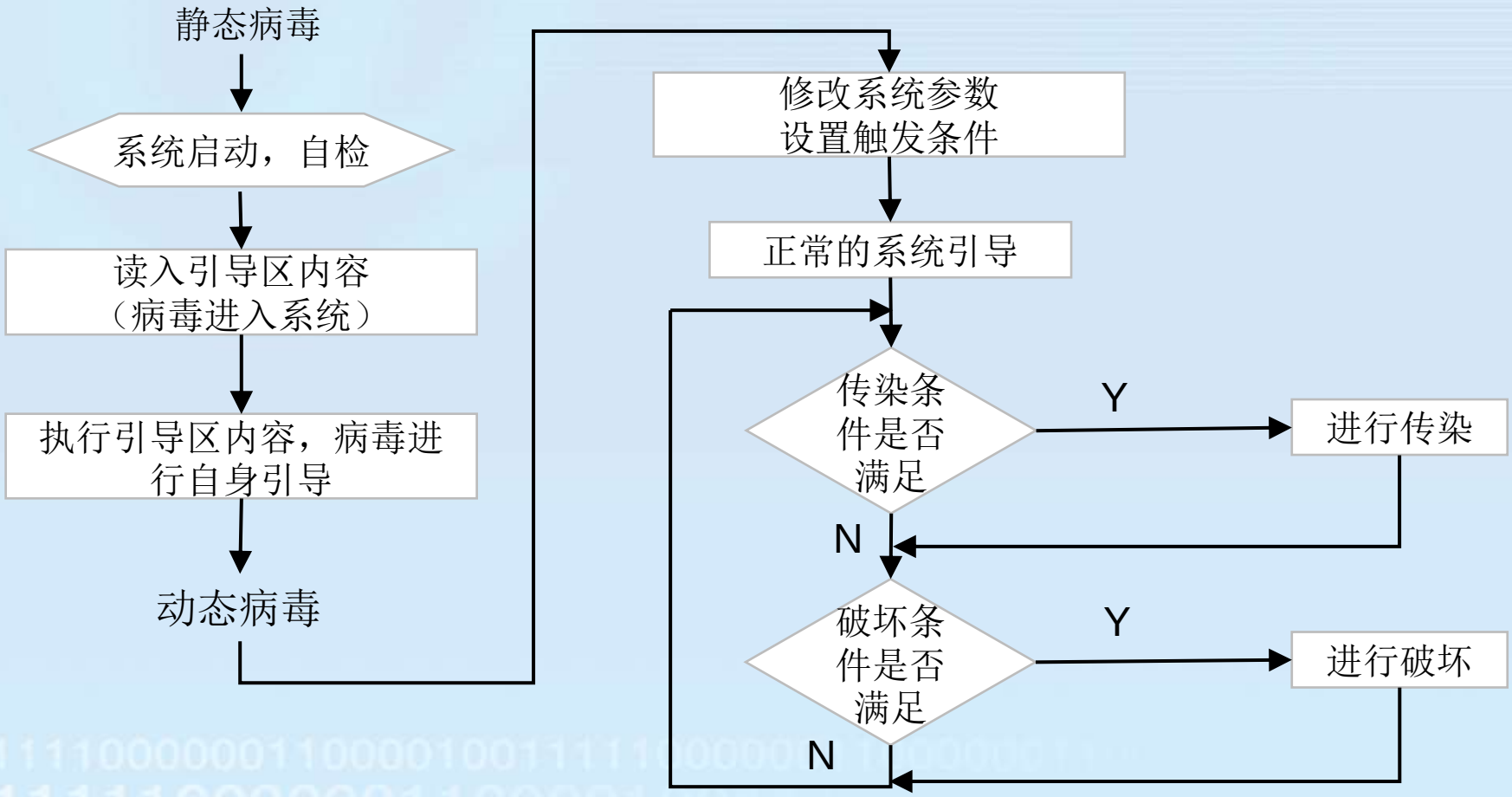




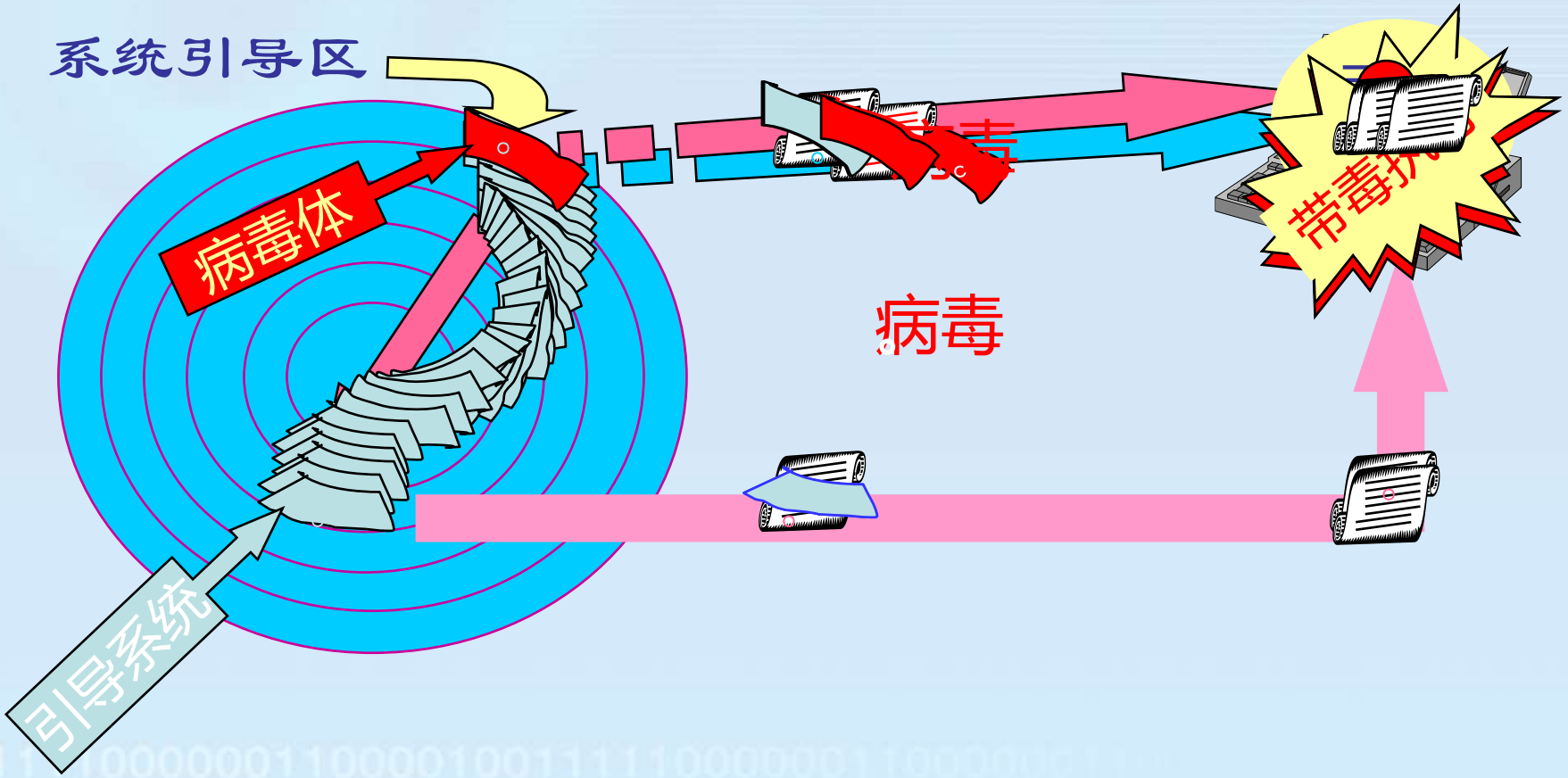
引导型病毒—系统引导过程



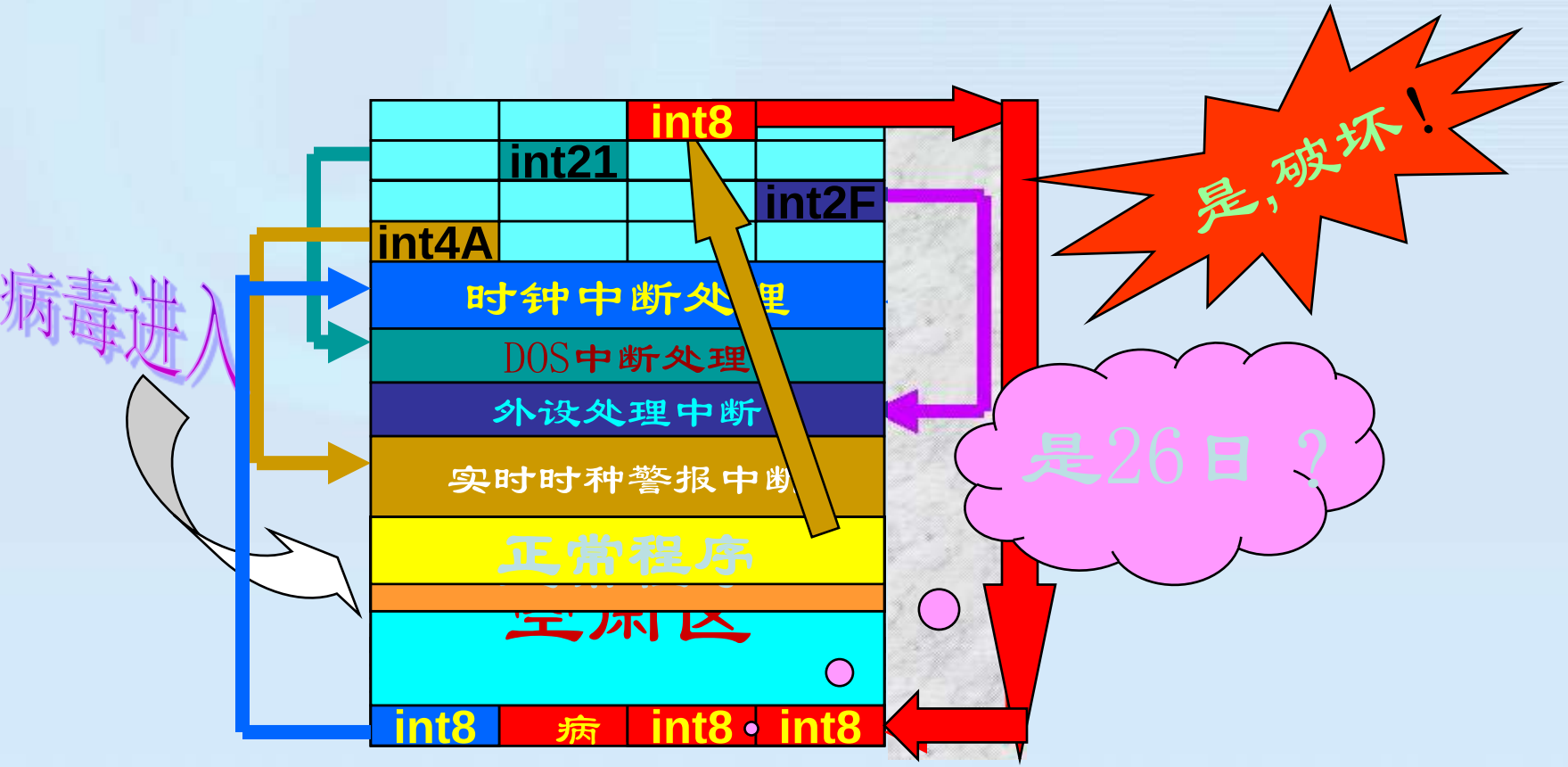
引导型病毒的工作流程


静态病毒
系统启动，自检
读入引导区内容
（病毒进入系统）
执行引导区内容，病毒进行自身引导
动态病毒
修改系统参数
设置触发条件
正常的系统引导
传染条件是否满足
Y
进行传染
N
破坏条件是否满足
Y
进行破坏
N

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？

- **寄生技术：** 病毒在感染的时候，将病毒代码加入正常程序之中，原正常程序功能的全部或者部分被保留。
- **驻留技术：** 当被感染的文件执行时，病毒的一部分功能模块进入内存，即使程序执行完毕，它们仍一直驻留在内存中。
- **加密变形技术：** 如没有实际用处，但会干扰代码阅读甚至反编译，却不影响程序功能的花指令；又如在二进制的程序中植入一段代码，在运行的时候优先取得程序的控制权，之后再把控制权交还给原始代码，隐藏程序真正的入口点的加壳技术。
- **隐藏技术：** 病毒在进入用户系统之后，会采取种种方法隐藏自己的行踪，使病毒不易被用户和反病毒软件发现。

木马基本概念

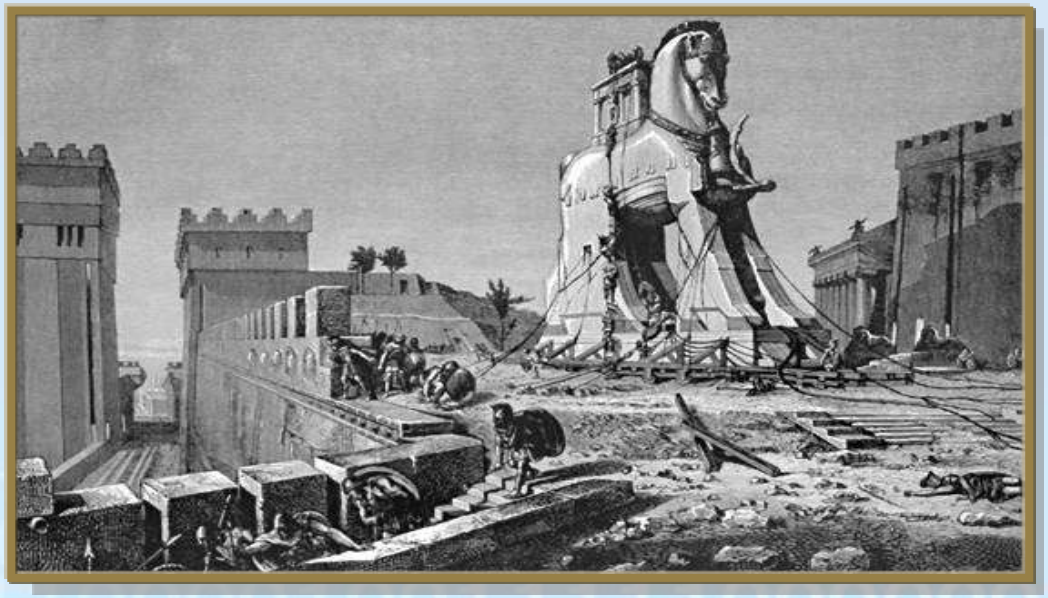
特洛伊木马（Troj），简称木马，是指寄宿在计算机内的一种非授权远程控制程序，是把自己伪装在正常程序内部的非法病毒程序，隐藏在系统中用以完成未授权功能。它原本属于一类基于远程控制的工具。

- 具有欺骗、隐蔽、伪装性强和信息窃取的特点
- 带有黑客性质，有强大的控制和破坏能力，可窃取密码、控制系统、操作文件等
- 通常使用户很难判断它到底是合法程序还是木马。

区别于其他恶意代码，木马不以感染其它程序为目的，一般也不使用网络进行主动复制传播。

➢ 特洛伊木马的传说

希腊人围攻特洛伊城，久久不能得手。后来想出了一个木马计，让士兵藏匿于巨大的木马中。大部队假装撤退而将木马摈弃于特洛伊城，让敌人将其作为战利品拖入城内。木马内的士兵则乘夜晚敌人庆祝胜利、放松警惕的时候从木马中爬出来，与城外的部队里应外合而攻下了特洛伊城。



特洛伊木马

特征：

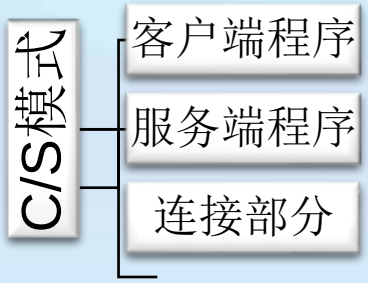
- 驻留在目标计算机，未经授权就可获得目标计算机的使用权
- 程序小，执行时不占用太多资源，执行时很难停止
- 随计算机自动启动并在某一端口进行侦听
- 一次执行后，就在系统中驻留，之后每次在系统加载时自动执行
- 对目标计算机执行特定操作

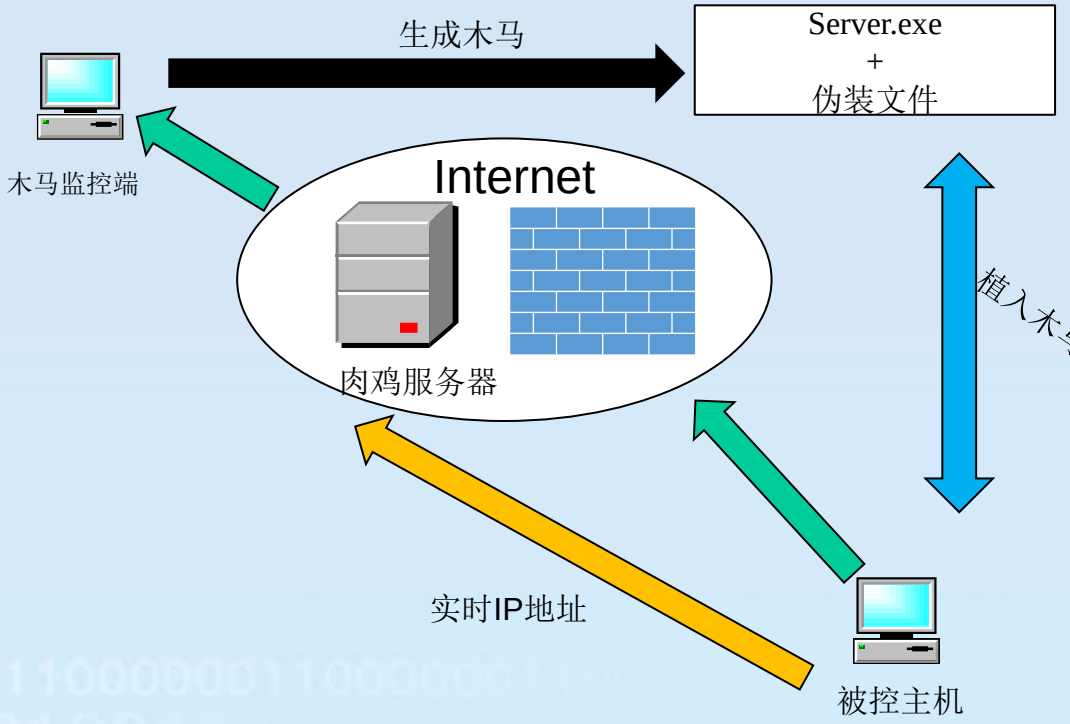
原理：

- C/S模式，服务器提供服务，客户机接受服务
- 作为服务器的主机一般会打开一个默认的端口进行监听，如果有客户机向服务器的这一端口提出连接请求，服务器上的相应程序就会自动运行，来应答客户机的请求。这个程序被称为守护进程。

特洛伊木马原理

木马一般为C/S模式，分为客户端程序、服务端程序、连接部分。被植入木马的机器称为服务器端，黑客控制客户端。

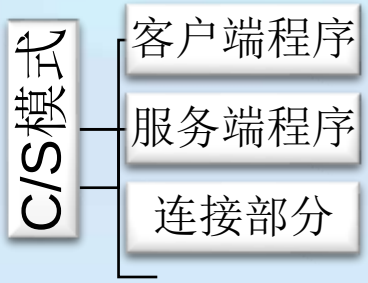


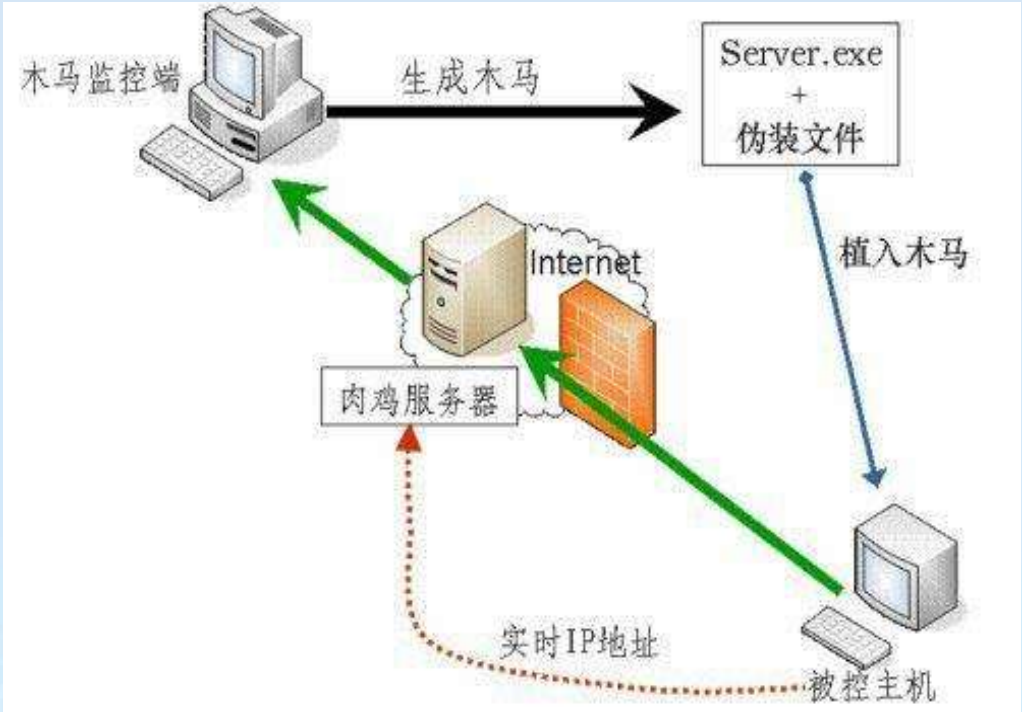


- 客户端：黑客控制
- 服务端：被植入木马

特洛伊木马原理

木马一般为C/S模式，分为客户端程序、服务端程序、连接部分。被植入木马的机器称为服务器端，黑客控制客户端。



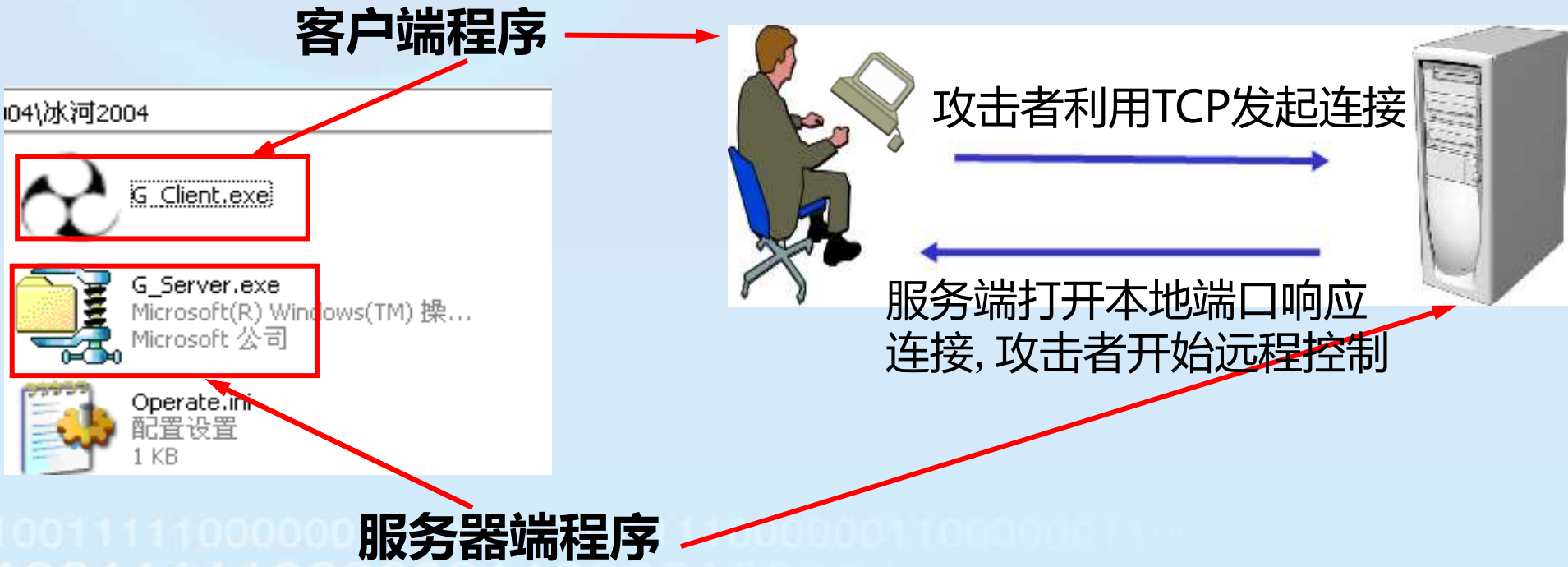


- 客户端：黑客控制
- 服务端：被植入木马

特洛伊木马原理

木马依照以下步骤实施攻击：

配置木马→ 传播木马→ 运行木马(自动安装、自启动) → 信息泄露→ 建立连接 → 远程控制



特洛伊木马分类

键盘记录木马	记录受害者的键盘敲击并且在日志文件里查找密码。随着Windows 的启动而启动并记录按键情况。多数情况会将信息发送到黑客指定的邮箱。
FTP木马	最简单最古老，惟一功能就是打开21端口，等待用户连接。新FTP 木马加入了密码功能，只有攻击者本人才知道正确的密码，从而进入对方计算机。
破坏型	唯一的功能是破坏并且删除文件，如DLL、INI、EXE文件，简单、易使用、非常危险。
密码发送型	可以隐蔽找到目标机的隐藏密码并发送到指定的信箱。比如木马在启动Windows时运行，利用Windows的密码记忆功能获取目标机的密码。
远程访问型	使用最广泛，可以远程访问被攻击者的硬盘。只要运行了服务端程序，客户端通过扫描等手段知道了服务端的IP 地址，就可以实现远程控制。

计算机病毒与木马的区别与联系

区别与联系

- 计算机病毒是恶意代码，能破坏和删除文件或自我复制；
- 木马是控制程序，黑客通过木马植入控制电脑进行操作，如盗号等

计算机病毒	特洛伊木马
恶意代码。破坏文件。自我复制	控制程序。控制电脑进行操作
直接威胁电脑安全，产生危害	以控制为主，协助破坏
具有感染性（最大区别）	不具有感染性

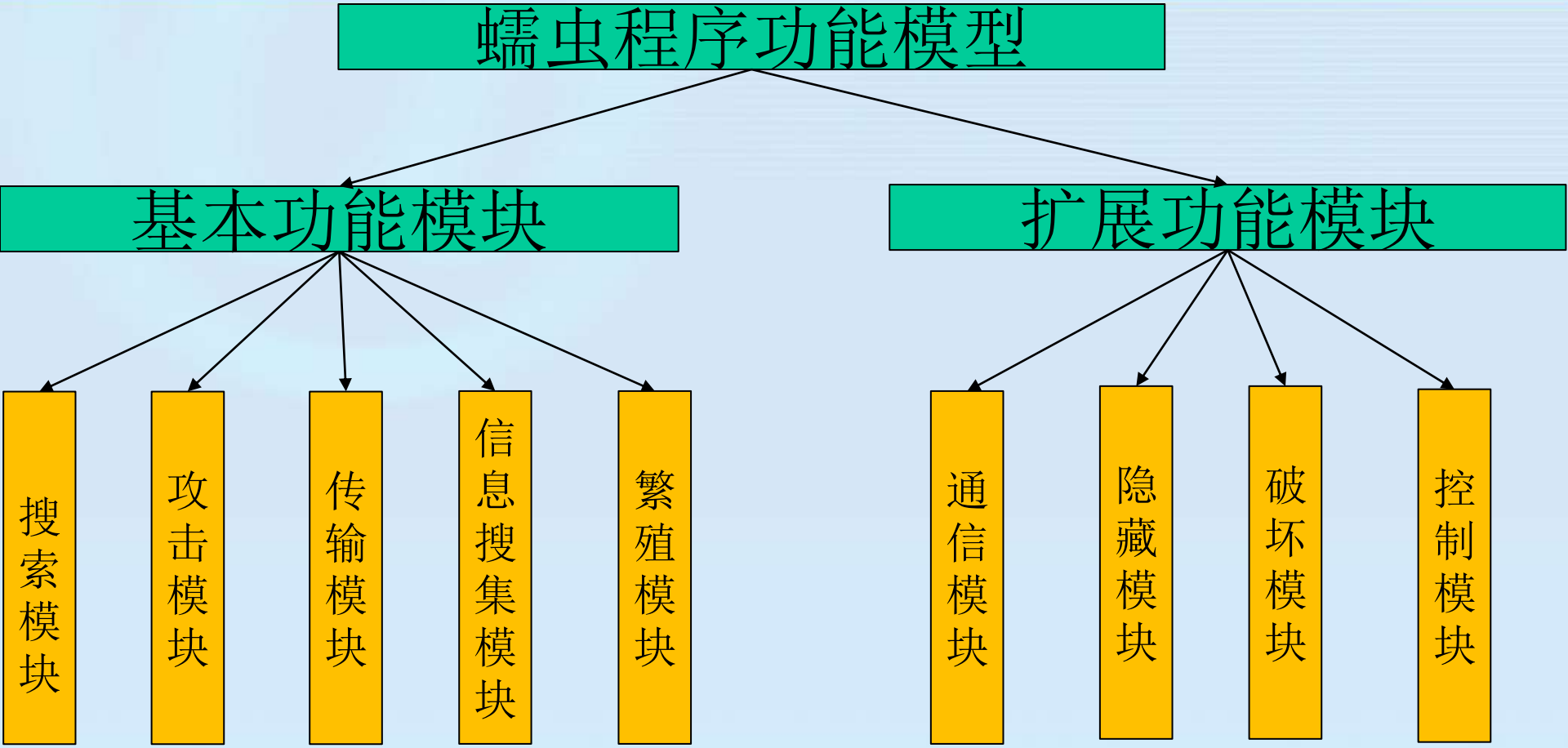
- 最大的区别就是病毒具有感染性，而木马一般不具有感染性。

蠕虫的特点

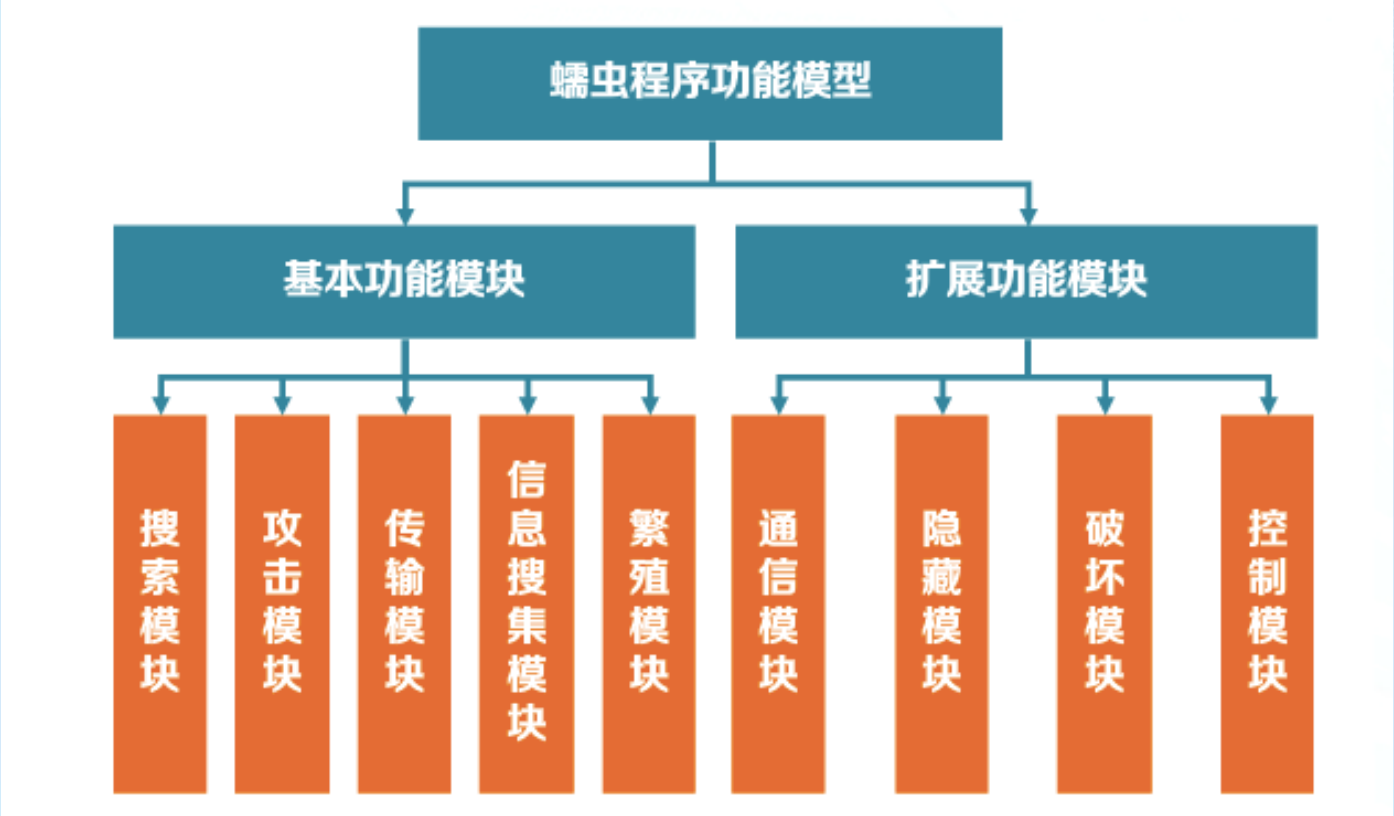
蠕虫的特点：

- 蠕虫的传播无需人为干预，并可通过网络进行自我复制，在复制过程中可能有改动。
- 传播不必通过“主机”程序或文件，因此它可以潜入用户的系统并允许其他用户或程序远程操控由蠕虫感染的计算机。
- 不使用驻留文件即可在系统间复制自身，而病毒需要传播受感染的驻留文件。通常，蠕虫将发布其中已包含“蠕虫”宏的文档，这样整个文档将在计算机之间传播，应将整个文档视为蠕虫。

蠕虫功能模块


蠕虫程序功能模型
基本功能模块
扩展功能模块
搜索模块
攻击模块
传输模块
信息搜集模块
繁殖模块
通信模块
隐藏模块
破坏模块
控制模块

蠕虫功能模块


蠕虫程序功能模型
基本功能模块
扩展功能模块
搜索模块
攻击模块
传输模块
信息搜集模块
繁殖模块
通信模块
隐藏模块
破坏模块
控制模块

计算机病毒与蠕虫

恶意代码类型	计算机病毒	蠕虫
复制性	自我复制、感染性	自我复制、感染性
定义特性	感染宿主文件/扇区	通过网络的自主传播
宿主	需要寄生宿主	不需要宿主，独立程序
传播路径	感染文件或扇区，通过文件交换或共享传播	直接通过网络传播，包括因特网和内网
传播是否需要用户交互	需要用户交互，如打开程序或文档	一般不需要用户交互，利用目标系统的漏洞或错误配置进行传播。但邮件蠕虫需要交互
防治措施	从宿主文件中摘除	系统打补丁（patch）系统打补丁（patch）

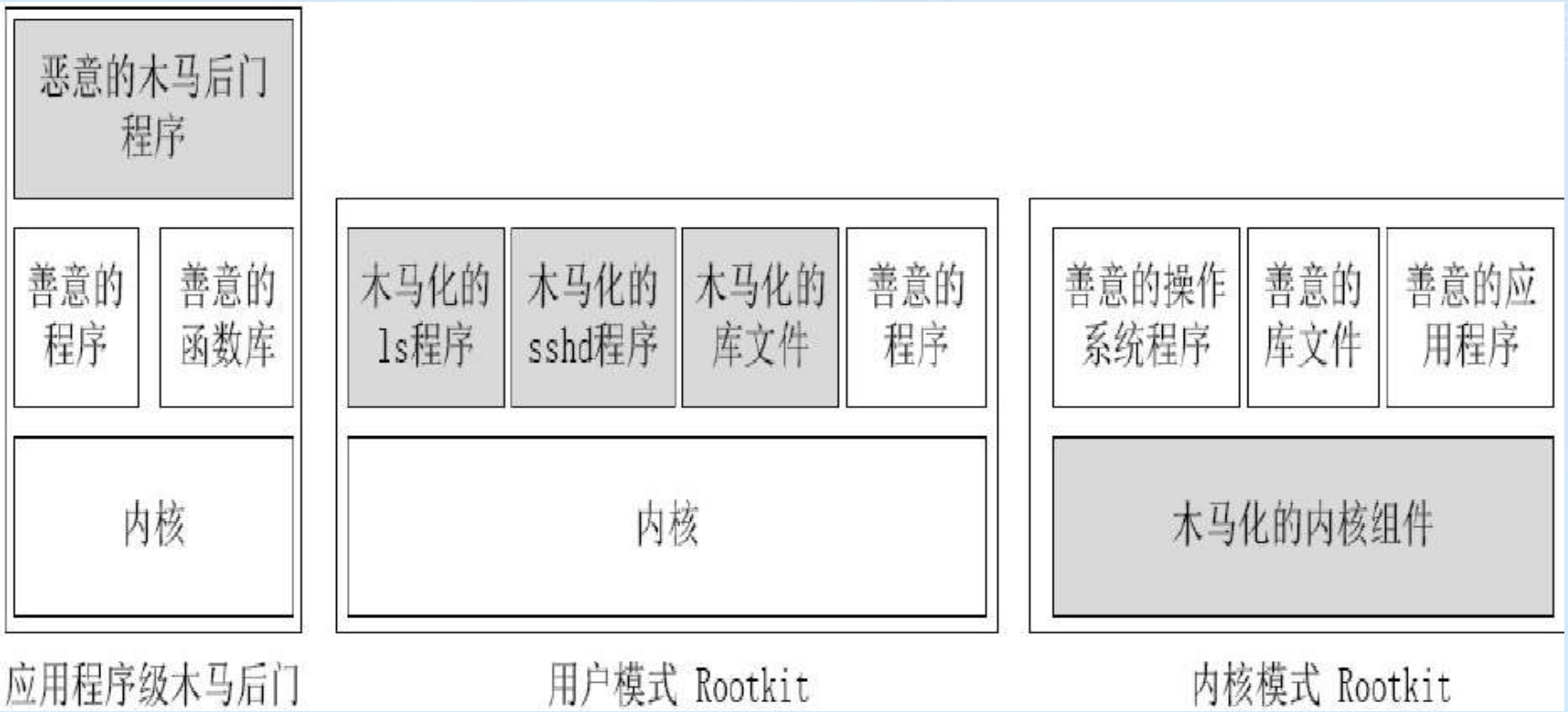
此外，由于蠕虫是一种独立程序，所以它们也可以作为病毒的寄生体，携带病毒，并在发作时释放病毒，进行双重感染。

应用程序级木马后门

用户模式Rootkit

内核模式Rootkit

Rootkit、木马和后门之间的位置对比



Rootkit、木马和后门之间的位置对比

僵尸网络

- Botnet（僵尸网络）是由“robot”（机器人）和“network”（网络）两个单词组合而成。

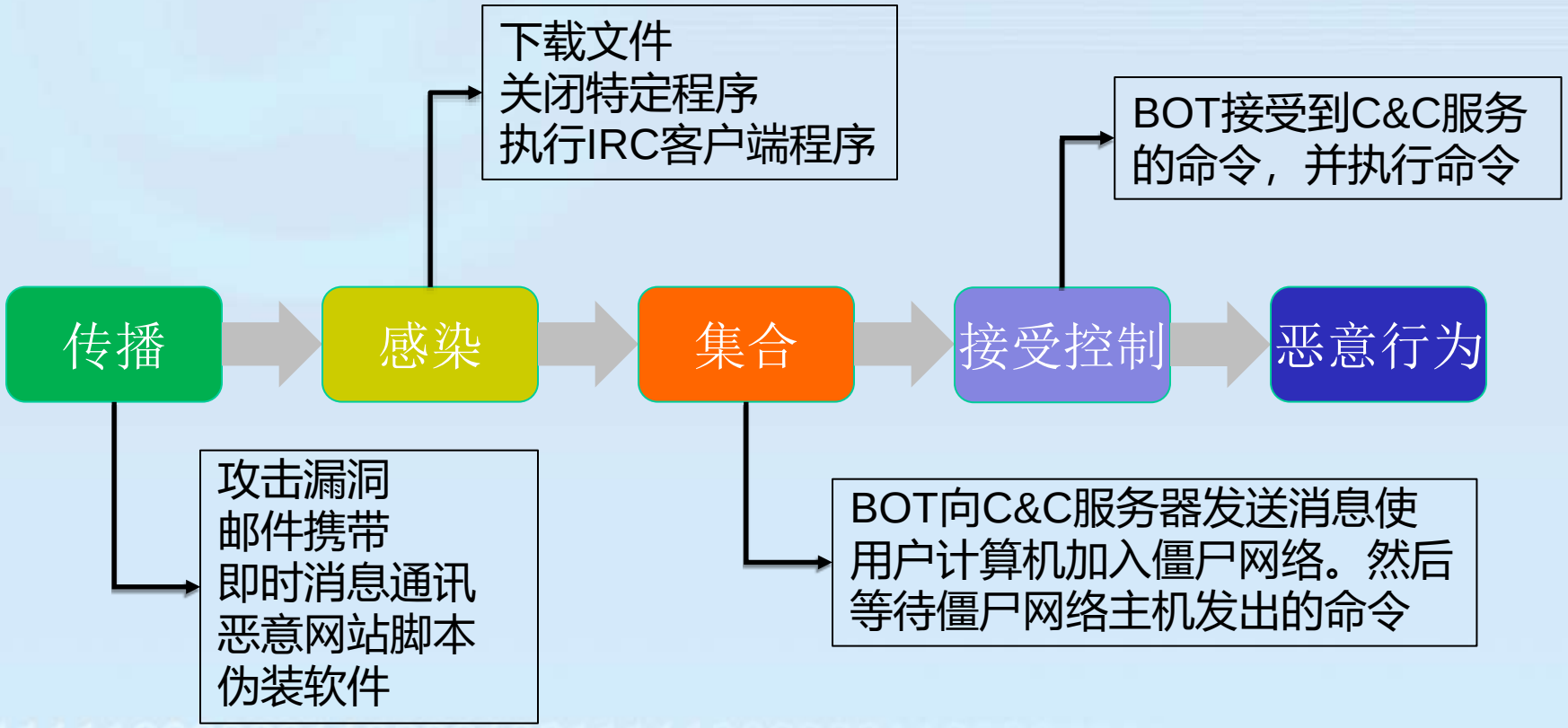
僵尸网络：是在网络蠕虫、特洛伊木马、后门工具等传统恶意代码形态的基础上发展、融合而产生的一种新型攻击方式。

定义：Botnet 是指采用一种或多种传播手段，将大量主机感染bot程序（僵尸程序）病毒，从而在控制者和被感染主机之间所形成的一个可一对多控制的网络。

手段：攻击者使用特殊的木马病毒破坏多台用户计算机的安全，控制每一台计算机，然后将所有受感染的计算机组成一个能够让犯罪分子远程管理的“机器人”网络。

僵尸网络(Botnet)

Botnet工作流程


下载文件
关闭特定程序
执行IRC客户端程序
BOT接受到C&C服务的命令，并执行命令
传播
感染
集合
接受控制
恶意行为
攻击漏洞
邮件携带
即时消息通讯
恶意网站脚本
伪装软件
BOT向C&C服务器发送消息使用户计算机加入僵尸网络。然后等待僵尸网络主机发出的命令

工作原理

传播

手段：
1.主动攻击漏洞
2.邮件病毒
3.即时通信软件
4.恶意网站脚本
5.特洛伊木马

加入

被感染主机都会随着隐藏在自身上的bot程序的发作而加入到Botnet中去。

在基于IRC协议的Botnet中，感染bot程序的主机会登录到指定的服务器和频道中去，在登录成功后，在频道中等待控制者发来的恶意指令。

控制

攻击者通过中心服务器发送预先定义好的控制指令，让被感染主机执行恶意行为，如发起DDos攻击、窃取主机敏感信息、更新升级恶意程序等。

Internet Relay Chat

其特点是速度快、低延时、低带宽占用。所有用户可以在一个被称为\"Channel\"（频道）的地方就某一话题进行交谈或密谈。每个IRC的使用者都有一个Nickname（昵称）。

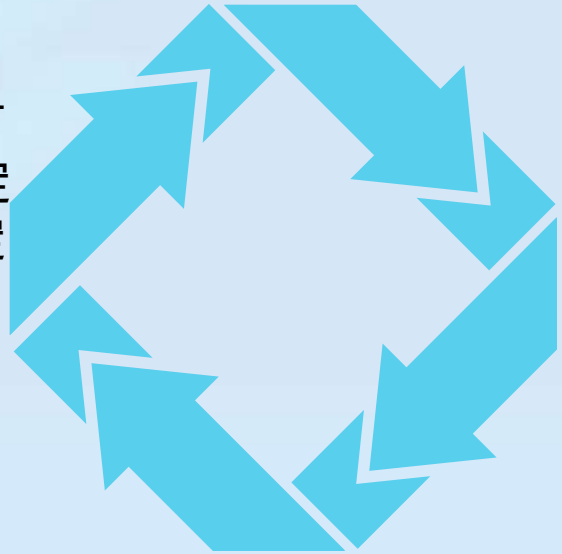
协议流程如下：

- 1.解析域名
- 2.建立TCP连接
- 3.发送指令
- 4.加入预定义频道

危害

01

拒绝服务攻击(DDos)
攻击者可以向自己控制的所有bots发送指令，让它们在特定的时间同时开始连续访问特定的网络目标



02

发送垃圾邮件
一些bots会设立sockv4、v5代理，这样就可以利用Botnet发送大量的垃圾邮件

03

窃取秘密
Botnet的控制者可以从僵尸主机中窃取用户的各种敏感信息和其他秘密，例如个人帐号、机密数据等

04

滥用网络资源和挖矿
例如种植广告软件，点击指定的网站以及虚拟货币的挖掘

应对措施

01. 采用Web过滤服务（最有力武器）

02. 转换浏览器

03. 禁用脚本（极端，不利于工作效率）

04. 部署防御系统

05. 保护用户生成的内容

06. 使用补救工具

恶意代码的检测与防范

（1）静态分析方法

在不运行恶意代码的情况下，利用分析工具对恶意代码的静态特征和功能模块进行分析的方法。

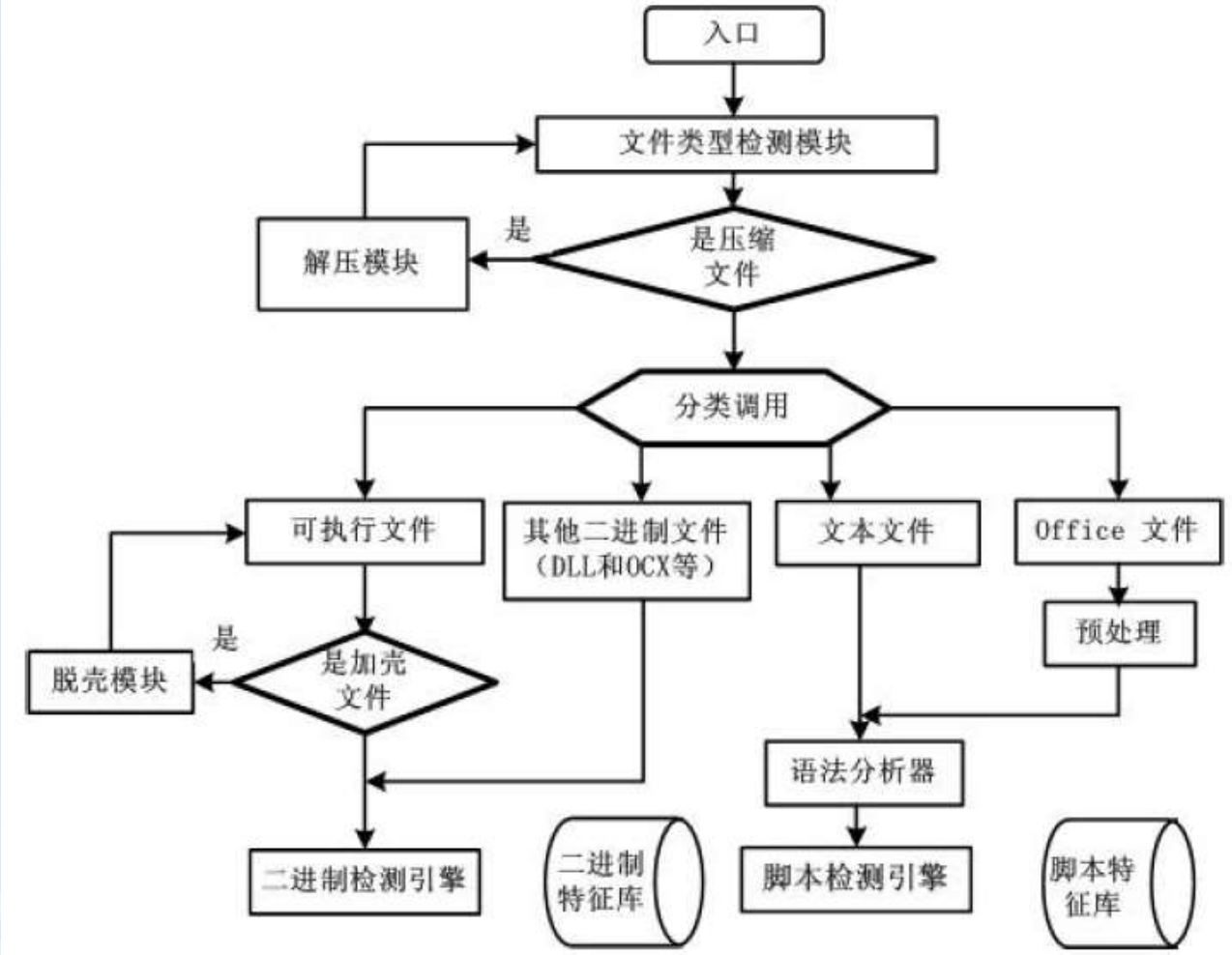
- 基于代码特征的分析方法

  分析过程中，不考虑恶意代码的指令意义，而是分析指令的统计特性、代码的结构特性等。

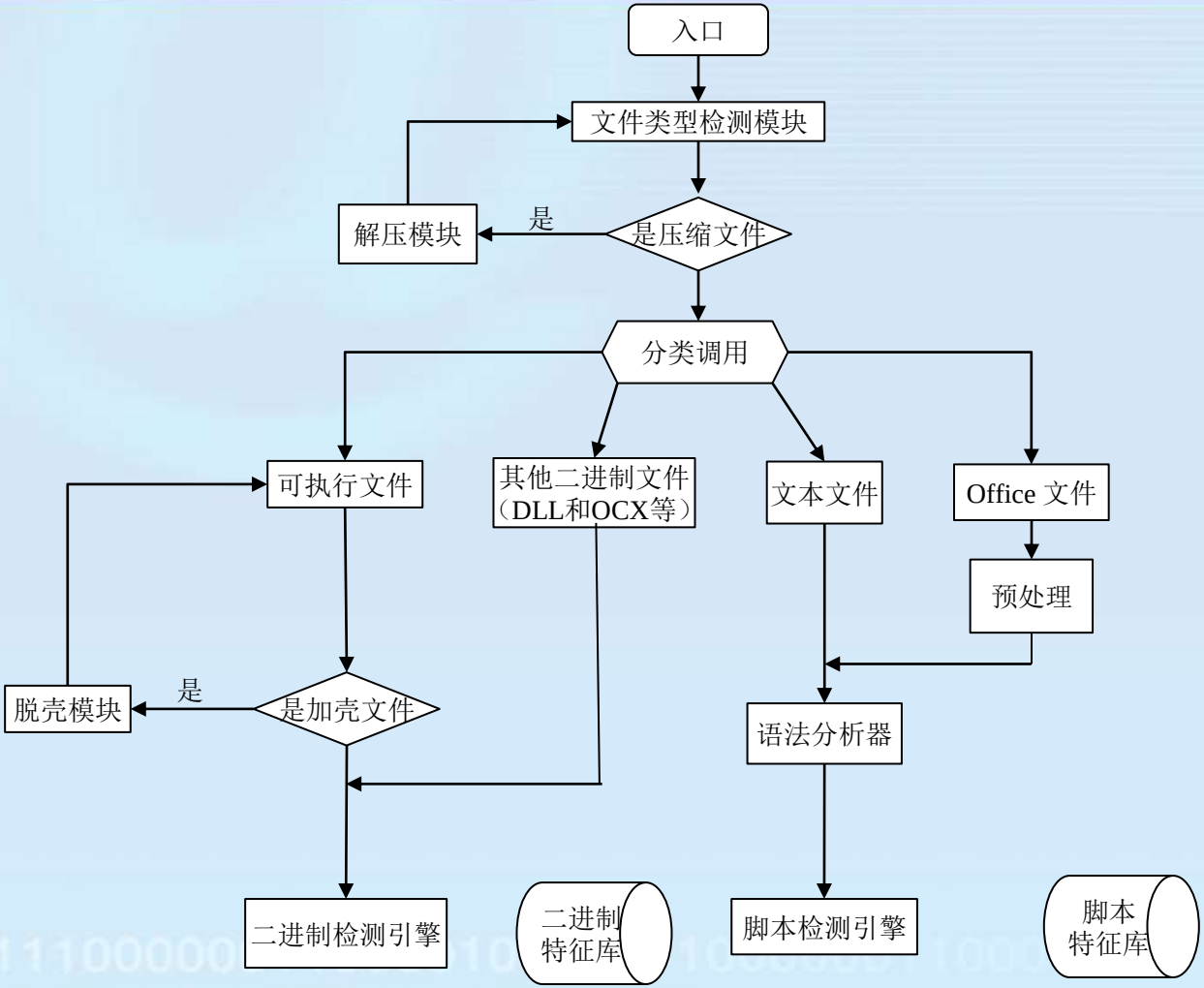
- 基于代码语义的分析方法

  要求考虑构成恶意代码的指令含义，通过理解指令语义建立恶意代码的流程图和功能框图，进一步分析恶意代码的功能结构。

基于特征的扫描技术


入口
文件类型检测模块
是
解压模块
是压缩文件
分类调用
可执行文件
其他二进制文件（DLL和OCX等）
文本文件
Office 文件
是
脱壳模块
是加壳文件
预处理
语法分析器
二进制检测引擎
二进制特征库
脚本检测引擎
脚本特征库

基于特征的扫描技术


入口
文件类型检测模块
是压缩文件
是
解压模块
分类调用
可执行文件
其他二进制文件
（DLL和OCX等）
文本文件
Office 文件
预处理
是加壳文件
是
脱壳模块
语法分析器
二进制检测引擎
二进制
特征库
脚本检测引擎
脚本
特征库

（2）动态分析方法

通过监视恶意代码运行过程从而了解恶意代码功能。

- 外部观察法

利用系统监视工具观察恶意代码运行过程中系统环境的变化，通过分析这些变化判断恶意代码的功能。

- 跟踪调试法

通过跟踪恶意代码执行过程使用的系统函数和指令特征分析恶意代码功能的技术。

（3）基于网络的恶意代码检测方法

- 基于GRIDS的恶意代码检测

著名的GRIDS主要针对大规模网络攻击和自动化入侵设计的，它收集计算机和网络活动的数据以及它们之间的连接，在预先定义的模式库的驱动下，将这些数据构建成网络活动行为来表征网络活动结构上的因果关系。

它通过建立和分析节点间的行为图（Activity Graph），通过与预定义的行为模式图进行匹配，检测恶意代码是否存在，是当前检测分布式恶意代码入侵的有效工具。

➢ 校验和

校验和是一种保护信息资源完整性的控制技术，例如Hash值和循环冗余码等。只要文件内部有一个比特发生了变化，校验和值就会改变。

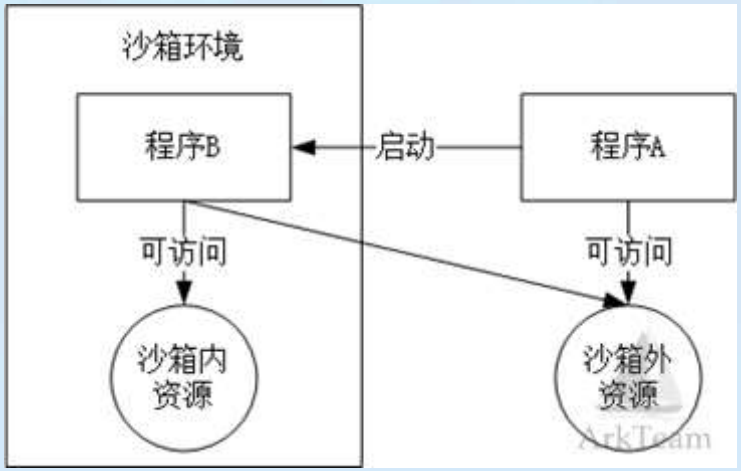
1.在恶意代码检测软件中设置校验和法。对目标文件计算其正常状态的校验和并写入被查文件中或检测工具中，而后进行比较。

2.在应用程序中嵌入校验和法。将文件正常状态的校验和写入文件本身中，每当应用程序启动时，比较前后两次校验和，实现自我检测功能。

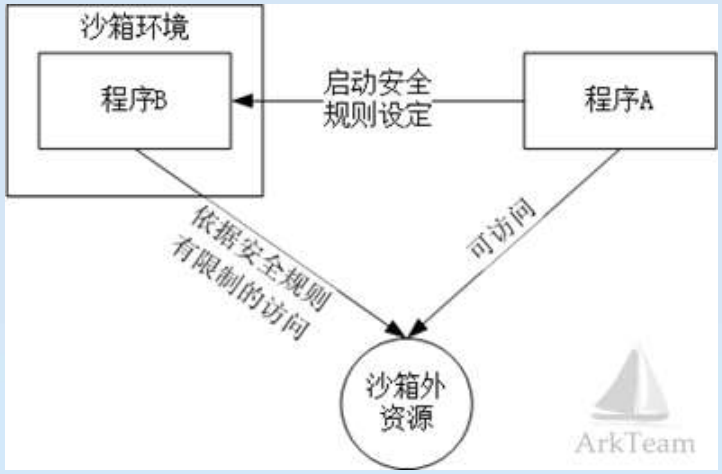
3.将校验和程序常驻内存。每当应用程序开始运行时，自动比较检查应用程序内部或别的文件中预留保存的校验和。

➢ 沙箱技术

沙箱技术是指根据系统中每一个可执行程序的访问资源，以及系统赋予的权限建立应用程序的“沙箱”，限制恶意代码的运行。



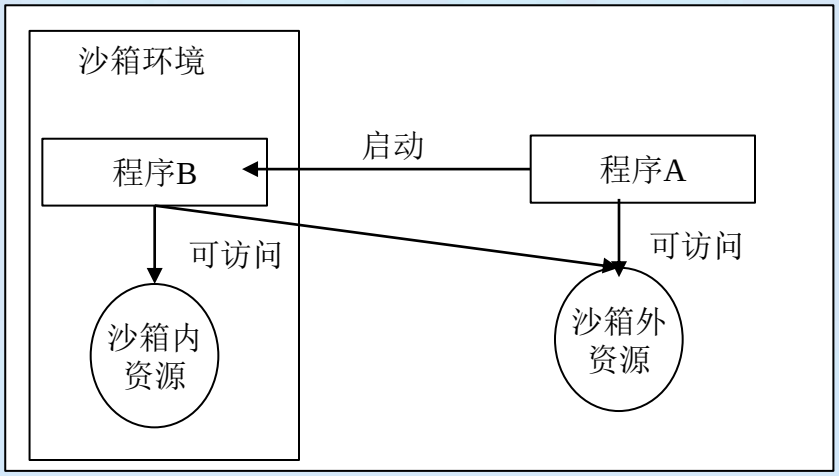
基于虚拟化



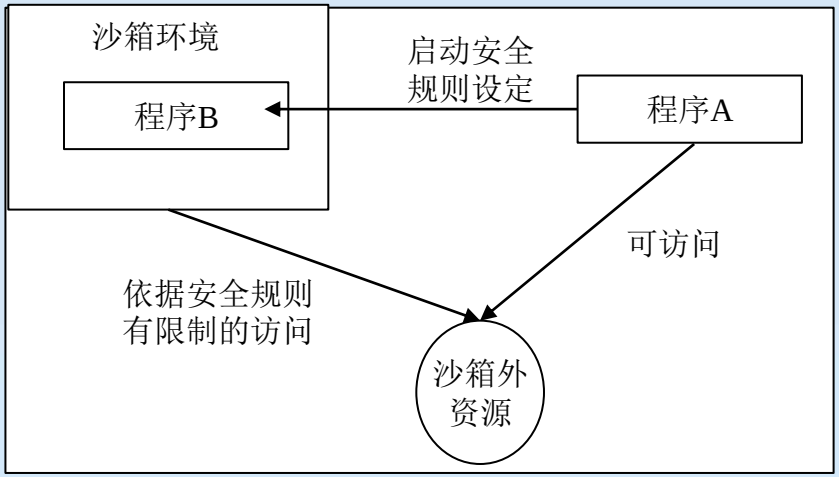
基于规则

➢ 沙箱技术

沙箱技术是指根据系统中每一个可执行程序的访问资源，以及系统赋予的权限建立应用程序的“沙箱”，限制恶意代码的运行。



基于虚拟化

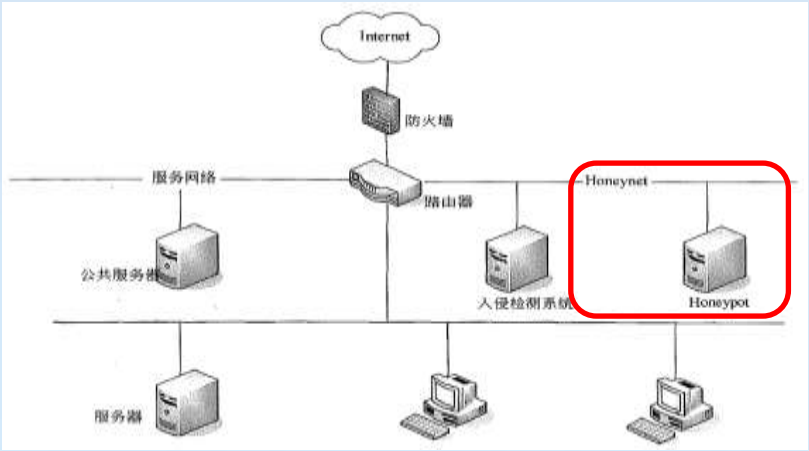


基于规则

➢ 基于HoneyPot的检测防御

蜜罐是一种被侦听、被攻击或已经被入侵的资源。注意：Honeypot并非一种安全解决方案，它只是一种工具，而且只有Honeypot受到攻击，它的作用才能发挥出来。

- 要想更好地防御网络攻击，则需要更清楚地了解攻击者的意图和手段。Honeypots（蜜罐）系统即为此而生。
- 蜜罐系统好比是情报收集系统。蜜罐好像是故意让人攻击的目标，引诱黑客前来攻击。所以攻击者入侵后，你就可以知道他是如何得逞的，



➢ 基于HoneyPot的检测防御

蜜罐是一种被侦听、被攻击或已经被入侵的资源。注意：Honeypot并非一种安全解决方案，它只是一种工具，而且只有Honeypot受到攻击，它的作用才能发挥出来。

- 要想更好地防御网络攻击，则需要更清楚地了解攻击者的意图和手段。Honeypots（蜜罐）系统即为此而生。
- 蜜罐系统好比是情报收集系统。蜜罐好像是故意让人攻击的目标，引诱黑客前来攻击。所以攻击者入侵后，你就可以知道他是如何得逞的，