第12章
恶意代码攻击及检测

中国科学技术大学
曾凡平
billzeng@ustc.edu.cn

第12章 恶意代码攻击

12.1 恶意代码概述
12.2 计算机病毒概述
12.3 几种常见恶意代码的实现机理
12.4 网络蠕虫
12.5 木马
12.6 恶意活动代码的防御
12.7 恶意代码检测与分析技术

参考：第14章 恶意代码攻击

- 《网络信息安全》，曾凡平编著，机械工业出版社, 2016.01

12.1 恶意代码概述

- 恶意代码最初是指最传统意义上的病毒和蠕虫。
- 随着攻击方式的增多，恶意代码的种类也逐渐增加。
- 木马、后门、恶意脚本、广告软件、间谍软件等都是恶意代码。

12.1.1 恶意代码的定义

定义一

- 恶意代码是任何的程序或可执行代码，其目的是在用户未授权的情况下更改或控制计算机及网络系统。

定义二

- 恶意代码又称恶意软件。这些软件也可称为**广告软件（adware）**、**间谍软件（spyware）**、**恶意共享软件（malicious shareware）**。是指在未明确提示用户或未经用户许可的情况下，在用户计算机或其他终端上安装运行、侵犯用户合法权益的软件。

恶意代码的定义

定义三

- 恶意代码是指故意编制或设置的、对网络或系统会产生威胁(或潜在威胁)的计算机代码。
- 最常见的恶意代码有计算机病毒（简称病毒）、特洛伊木马（简称木马）、计算机蠕虫（简称蠕虫）、后门、逻辑炸弹等。
- 最近有人把**不需要的代码**（**Unwanted Code**）也归类为恶意代码。Unwanted Code是指没有作用却会带来危险的代码，包括所有可能与某个组织安全策略相冲突的软件。

12.1.2 恶意代码的分类

- 根据其代码是否独立，可以将其分成独立的和寄生的恶意代码。

1. 独立的恶意代码能够独立传播和运行，是一个完整的程序，它不需要寄宿在另一个程序中。
2. 非独立的恶意代码只是一段代码，必须寄生在某个程序(或文档)中，作为该程序的一部分进行传播和运行。

恶意代码的分类

- **根据其是否能自我复制(自动传染)，可以将其分成广义病毒及普通的恶意代码。**
- 对于非独立恶意代码，自我复制过程就是将自身嵌入宿主程序的过程，这个过程也称为感染宿主程序的过程。
- 对于独立恶意代码，自我复制过程就是将自身传播给其他系统的过程。不具有自我复制能力的恶意代码必须借助其他媒介进行传播。
- **传统意义上的病毒是狭义病毒**，指同时具有寄生和传染能力的恶意代码。

(1) 后门

- 后门也被称为陷阱，它是某个正常程序的秘密入口，通过该入口启动程序，可以**绕过正常的访问控制过程**。因此，获悉后门的人员可以绕过访问控制过程，直接对资源进行访问。
- 后门最初的作用是程序员开发具有鉴别或登录过程的应用程序时，为避免每一次调试程序时都需输入大量鉴别或登录过程所需要的信息，通过后门启动程序的方式来绕过鉴别或登录过程。当程序正式发布时，程序员会删除该后门。后来程序员有意在程序中留下后门，以防止非授权用户的盗用。再后来，某些（尤其是免费的共享）软件故意留下后门，以窃取目标系统的敏感信息。

(2) 逻辑炸弹

- 逻辑炸弹是包含在正常应用程序中的一段恶意代码，当某种条件出现，如到达某个特定日期、增加或删除某个特定文件等，将触发这一段恶意代码，执行这一段恶意代码将导致非常严重的后果，如删除系统中的重要文件和数据、使系统崩溃等。

- 逻辑炸弹最初是程序员用于保护版权而采取的手段，一般不破坏目标系统。后来被用于讹诈和报复非授权用户，会对系统造成破坏。

(3) 间谍软件

- 间谍软件（Spyware）与商业软件产品有关。
- 有些商业软件产品在安装到用户机器上的时候，未经用户授权就通过Internet让用户方软件与开发商软件进行通信，这部分通信软件就叫做谍件。
- 用户只有安装了基于主机的防火墙，通过记录网络活动，才可能发现软件产品与其开发商在进行定期通讯。谍件作为商用软件包的一部分，多数是无害的，其目的大多在于扫描系统，取得目标系统的一些状态信息，以更好地改进软件产品。

(4) 特洛伊木马

- 特洛伊木马也是包含在正常应用程序中的一段恶意代码，一旦执行这样的应用程序，将触发恶意代码。**木马的功能主要在于削弱系统的安全控制机制，尤其是访问控制机制。**
- 远程访问特洛伊RAT（Remote Access Trojan）是安装在受害者机器上，实现非授权的网络访问的程序。RAT可以伪装成其他程序，迷惑用户下载安装。
- 目前值得关注的是一些提供免费软件的网站强制用户安装其下载软件才能完成软件的下载，这很可能附带了某些附加功能，有可能侵害用户的隐私。因此，下载免费软件时要特别慎重。

(5) 病毒

- 这里的病毒是狭义病毒，即传统意义上的病毒，指那种既具有自我复制能力，又必须寄生在其他程序(或文件)中的恶意代码。

- 它和陷阱门、逻辑炸弹的最大不同在于自我复制能力。通常情况下，陷阱门、逻辑炸弹不会感染其他实用程序，而病毒会自动将自身添加到其他实用程序中。

(6) 蠕虫

- 蠕虫也是一种病毒，但它和狭义病毒的最大不同在于自我复制过程。病毒的自我复制过程需要人工干预，无论是运行感染病毒的实用程序，还是打开包含宏病毒的电子邮件，都不是由病毒程序自我完成的。蠕虫的传播不需要人工干预，他其实是**能完成特定攻击过程的自治软件**，它自动完成以下任务：

① 查找攻击对象：利用网络侦察技术查找下一个存在漏洞的目标。

② 入侵目标：利用漏洞入侵目标系统。

③ 复制自己：复制自己到被攻击的系统，并运行它。

(7) 僵尸(Zombie)

- Zombie(俗称僵尸)是一种在被入侵者控制的系统上安装的、能对某个特定系统发动攻击的恶意代码。

- Zombie主要用于定义恶意代码的功能，并没有涉及该恶意代码的结构和自我复制过程，因此，分别存在符合狭义病毒的定义和蠕虫定义的Zombie。

(8) P2P系统

- 基于Internet的点到点（peer-to-peer）应用程序已经广泛应用于因特网。然而，从安全性考虑，P2P软件对于企业是十分不利的。
- P2P程序可以通过HTTP或者其他公共端口穿透防火墙，直接连接到企业的内部网。这种连接如果被利用，就会给组织或者企业带来很大的危害。

- 对于企业信息网络的安全性而言，在某种程度上可以将P2P软件看作恶意代码。

(9) RootKit

- RootKit最初是指一组能帮助使用者获取系统root权限的工具包，这里的**RootKit是一种恶意程序，用于获取目标主机root权限之后隐藏攻击者访问痕迹，使得攻击者不被发现，从而能够长期拥有管理员权限**。
- RootKit具有很好的隐蔽性和潜伏性，难以检测。
- 一般情况下，RootKit不负责root权限的获取，只是辅助恶意代码长期存活于目标系统中。所以RootKit通常与木马程序、以太网嗅探器、日志清理工具等配合使用。

(9) RootKit

RootKit的攻击过程

- 攻击者首先通过使用其他远程攻击软件或者安全漏洞来获得系统的最高访问权限，成功侵入系统后，接着需要在目标主机中安装上RootKit，完成远程命令和操作之后，RootKit会将自身和其他攻击进程等相关文件都隐藏起来，并清除系统日志中的有关信息，这样能够保证指定的恶意程序能够长期运行从而持续性窃取机密信息。

用户级RootKit和核心级RootKit

- 用户级RootKit侵入系统通常是通过修改普通用户或者管理员执行的程序来实现的，这种方法比较容易实现，但同时隐蔽性不够高，容易暴露；
- 而核心级RootKit 一般是侵入操作系统内核，伪装成内核的一部分来进行隐藏和创建后门。

12.1.3 恶意代码攻击流程

(1)寻找目标

- 本地文件、移动存储设备、电子邮件、远程系统。

(2)将自身保存在目标之中

- 主动性的恶意代码如病毒、蠕虫程序会靠自身来实现这一步骤，而木马、RootKit等则需要人为植入目标系统或者利用恶意网站欺骗用户下载。

(3)触发目标系统中的恶意代码执行

- 包括主动触发和被动触发。

(4)让自身长期存活于目标系统之中

- 静态存在形式，如文件和启动项文件；
- 动态存在形式，如进程、服务和端口等。

12.1.4 恶意代码攻击技术

1)代码注入技术

- 代码注入技术就是攻击者将一段恶意的可执行代码插入到其他程序地址空间中。大多数代码注入技术是利用目标系统中已知的系统服务和网络服务的漏洞、缺陷来实现的。

2)缓冲区溢出攻击技术

3)三线程技术

- 恶意程序执行时会同时建立三个线程，分别为主线程、监视线程和守护线程

4)端口复用技术

5)端口反向连接技术

12.1.5 恶意代码生存技术

1)反跟踪技术

(1)反静态跟踪技术：

- 为了防止逆向分析者对程序反编译进行静态分析，可以将恶意代码分为多块并对每块进行加密，并且不同时装载到内存中，每执行完一段代码就立刻将其删除，使得分析者得不到完整的攻击代码。还有一种方法是在恶意代码中加入一些伪指令使代码变形，这样分析者进行反编译时得到的指令将会包含大量的无效干扰信息，导致无法顺利地进行静态分析。

(2)反动态跟踪技术：

- 由于分析者主要是使用调试器(如Debug)的中断功能来进行恶意代码的跟踪和调试，所以，反动态跟踪技术的基本原理就是要破坏单步中断、断点中断或者破坏调试工具的运行环境，使得其无法跟踪程序的运行。常见的反动态跟踪方法主要包括修改中断服务程序入口地址、封锁键盘输入和屏幕显示、检测跟踪法等。

2)加密、加壳技术

3)变形技术

12.1.6 恶意代码长期存在的原因

- 利益的驱使是恶意代码泛滥的主要原因，另外软件漏洞也是恶意代码得以传播的主要因素。

(1) 利益驱使

- 如果无利可图，则没有人会冒着触犯法律的风险去散布和利用恶意代码。正是因为盗号木马、网银木马等恶意软件可以获得巨大的物质利益，而间谍软件等可以窃取用户的隐私进而讹诈用户，才使得攻击者乐此不疲、研发出越来越先进的恶意代码。

(2) 系统和应用软件存在漏洞

- 软件漏洞是导致网络信息系统安全的最根本原因。

- 分析与测试是发现软件漏洞的主要技术手段。然而由于软件的复杂性，在现有的资源条件下要对所有软件进行彻底分析与测试是一个尚待解决的世界难题。这就导致现有软件不可避免地存在缺陷和漏洞，这正是恶意代码赖以存在的基础。

12.2 计算机病毒概述

- 在此讨论的**计算机病毒是指狭义病毒**，即同时**具有寄生性和感染性的恶意代码**。计算机病毒将自身的精确拷贝或者可能演化的拷贝放入或链接入其他程序，从而感染其他程序。

12.2.1 计算机病毒的起源

- 计算机病毒的设想出现于1949年冯.诺依曼（John Von Neumann）的一篇论文《复杂自动装置的理论及组织的进行》，该论文描述了一种能自我繁殖的程序的构想。
- 在当时，绝大多数计算机专家都无法想象这种会自我繁殖的程序能够实现，但仍有少数人在研究这种会自我繁殖的程序。

“磁芯大战”(core war)

- 经过十几年的努力，这种程序以一种叫做**“磁芯大战”(core war)**的电脑游戏的形式产生了。

- 磁芯大战是当时美国电报电话公司(AT&T) 的贝尔实验室中三个年轻程序员发明出来的，他们是道格拉斯.麦耀莱（Douglas Mallory），维特.维索斯基（Victor Vysottsky）以及罗伯特.莫里斯(Robert T. Morris)。其中的莫里斯就是后来制造了“莫里斯蠕虫”的罗特·莫里斯的父亲。

- 磁芯大战(core war or core wars)其实就是汇编程序间的大战。程序在虚拟机中运行，并试图破坏其他程序，生存到最后即为胜者。由于能自我复制的病毒程序很可能对现实世界带来无穷的祸害，长久以来，懂得玩“磁芯大战”游戏的电脑工作者都严守一项不成文的规定：不对大众公开这些程序的内容。然而，这项规定在1983年被科恩.汤普逊（Ken Thompson）打破了。
- 科恩.汤普逊是当年一项杰出电脑奖的得主，在颁奖典礼上，他作了一个演讲，不但公开证实了电脑病毒的存在，而且还告诉所有听众怎样去写自己的病毒程序。潘多拉之盒就此被打开，许多程序员都了解到了病毒的原理，进而开始尝试编制这种具有隐蔽性、攻击性和传染性的特殊程序。

“巴基斯坦智囊”病毒

- 1986年，巴基斯坦的Basit和Amjad为了打击那些盗版软件的使用者，设计出了一个名为“巴基斯坦智囊”的病毒。
- “巴基斯坦智囊”病毒只传染软盘引导区，这就是世界上最早流行的真正意义上的病毒。

- 自此以后，病毒从隐秘走向公开，先是利用磁盘，然后是利用网络，迅速在全世界范围内扩散开来，成为了电脑用户的头号敌人。

12.2.2 病毒的分类

(1) 引导型病毒

- 主要感染计算机系统的引导部分，在系统启动时就运行了病毒。它感染软盘、硬盘的引导扇区或主引导扇区，在用户对软盘、硬盘进行读写操作时进行传播。

- 这种病毒在早期很流行，然而这种病毒技术过于简单，无法抵挡反病毒软件的查杀，现在已经基本绝迹了。

(2)文件型病毒

- 它主要感染可执行文件（如感染Windows系统的PE病毒和感染Linux系统的ELF病毒）。只要运行被感染的可执行文件，病毒就被加载并感染其它未中病毒的可执行文件。实现这种病毒要较高的编程技巧，难于编制，然而这种病毒也容易被查杀，因此现在也较少出现。

(3)混合型病毒

- 就是既能感染引导区，又能感染文件的病毒。混合型病毒的目的是为了综合利用以上二种病毒的传染渠道进行破坏。这种病毒也不多见了。

(4) 变形病毒

- 病毒传染到目标后，病毒自身代码和结构会发生变化，使得杀毒软件难于发现其特征，以抵抗杀毒软件。这一类病毒使用一个复杂的编解码算法，使自己每传播一份都具有不同的内容和长度。它们一般是由一段混有无关指令的解码算法和被改变过的病毒体组成。

- 以上四类病毒因为要改变可执行文件或引导区的内容，而正常程序一般是不会修改另一个可执行程序的，所以只要杀毒软件监控是否有可执行程序被修改就可以发现病毒。故此这类病毒的存活率在目前看来是很低的。

(5) 脚本病毒

- 利用脚本语言如Javascript、VBscript编写的病毒。这些病毒一般嵌入在html等文本文件中，作为文本文件的一部分而存在。

(6) 宏病毒

- 是利用高级语言——**宏语言**编制的病毒，与前几种病毒存在很大的区别。宏病毒充分利用宏命令的强大系统调用功能，实现某些涉及系统底层操作的破坏。宏病毒仅向 WORD 、 EXCEL 和 ACCESS 、 POWER POINT 、 PROJECT等**办公自动化程序编制的文档**进行传染，而不会传染给可执行文件。
- **脚本病毒和宏病毒保存在文档中**，而**文档是允许经常被修改的**，故反病毒软件难于区分是正常修改还是病毒感染文件。也就是说，较难查杀脚本病毒和宏病毒。

12.2.3 病毒的特性

(1) 感染性

- **感染性是指病毒具有把自身的拷贝放入其他程序(或文档)的特性**，这是计算机病毒最根本的属性，是判断某些可疑程序是否是病毒的主要标准。
- 感染计算机病毒的程序一旦被执行，则病毒代码首先被执行，然后根据触发条件决定是执行原来的程序还是感染其他文件。病毒会搜寻其他符合其感染条件的程序或存储介质，确定目标后再将自身代码插入其中，达到自我繁殖的目的。

(2) 非授权性

- 正常程序的执行先是由用户启动（通过鼠标点击或通过命令行输入命令），再由系统分配资源，最后完成某些给定的任务。
- 正常程序的目的对用户是可见的、透明的。
- 而病毒隐藏在正常程序中，当用户启动正常程序时窃取到系统的控制权，先于正常程序执行，病毒的动作、目的对用户是未知的，是未经用户允许的。

(3) 潜伏性

- 为了提高病毒的存活率，病毒入侵系统后并不是马上发作，而是尽量感染更多的计算机和文件，这段时间的计算机并未表现异常，也就难以被用户察觉。
- 另外，病毒一般是用短小精悍的汇编代码编写的，通常附在正常程序中或磁盘较隐蔽的地方，也有个别的以隐含文件形式出现，如果不经过代码分析，病毒程序与正常程序不易区分。
- **计算机病毒的这种隐蔽自己、使人难以发现的特性称为潜伏性。**

(4) 可触发性

- 不能被激活的病毒是没什么危害的。一般的说，计算机病毒会因某个特定的条件（如时间、数值、指令）而激活，激活后的病毒会攻击目标系统。病毒的这种能被激活的特性被称为可触发性。

(5) 破坏性

- **破坏文件或数据，扰乱系统正常工作的特性称为破坏性。**侵入系统的病毒都会影响目标系统的运行，轻者会降低计算机的工作效率，占用系统资源，重者可导致系统崩溃。早期的计算机病毒以删除文件、格式化磁盘、破坏分区表**等极端的恶意攻击**为主，其影响极为恶劣，往往会遭致全人类的强烈谴责。现在的病毒很少再做这种极端的破坏，而是和一些黑客技术相结合，窥探、修改、窃取系统的某些敏感信息，**最终目的是获取经济利益**。

12.2.4 病毒的结构

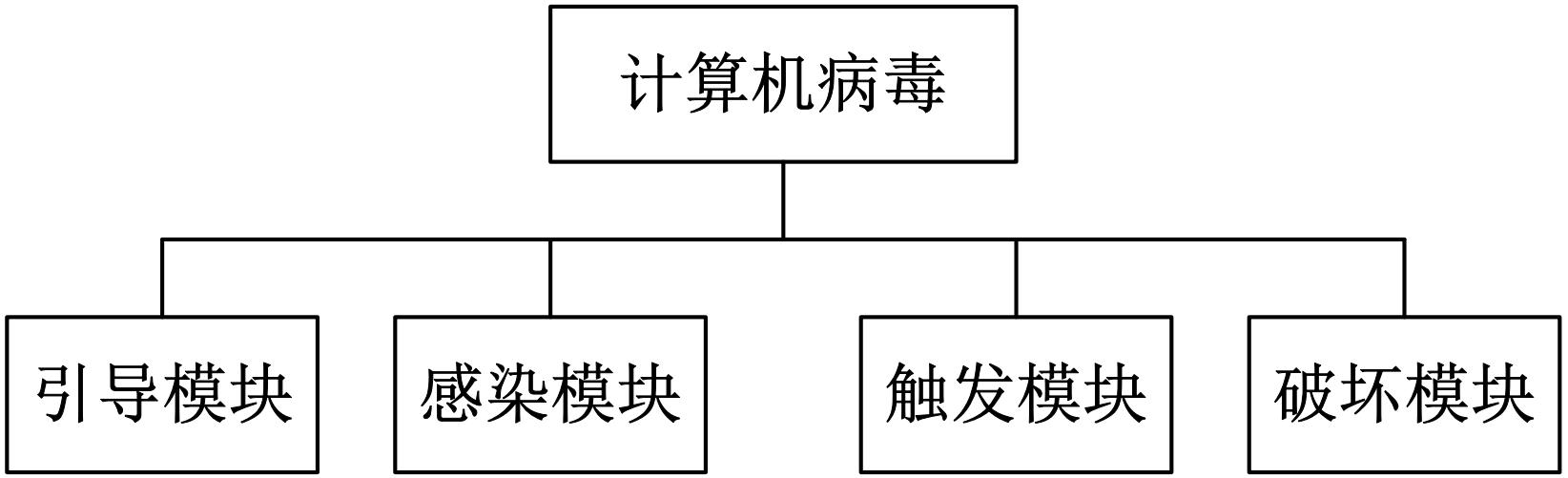
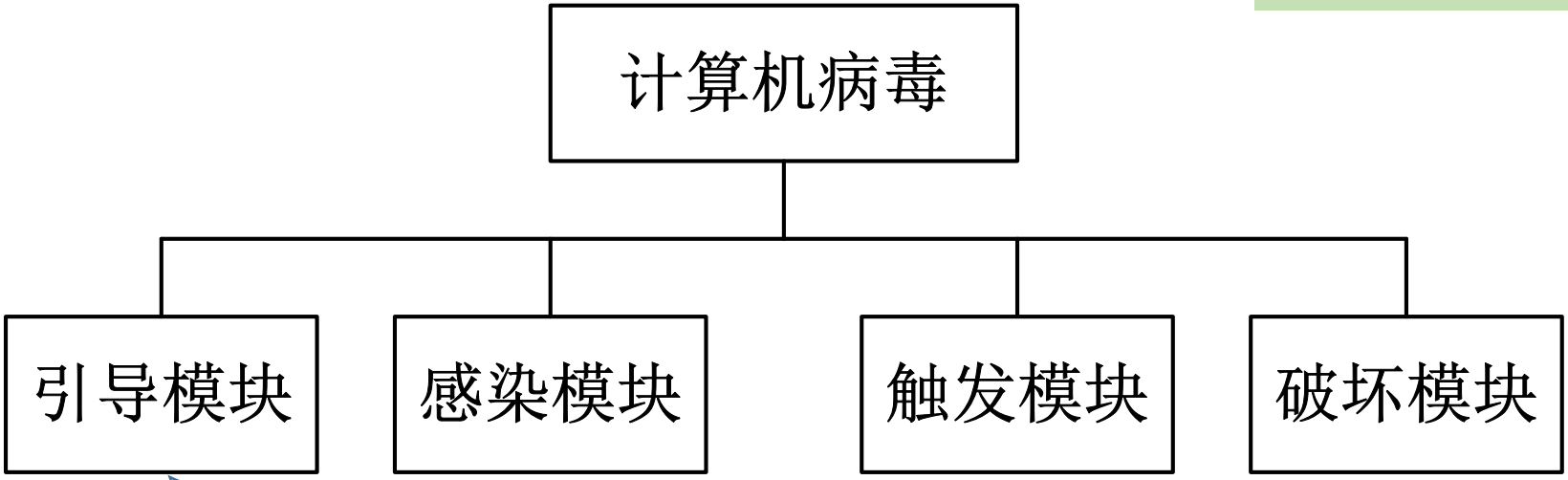
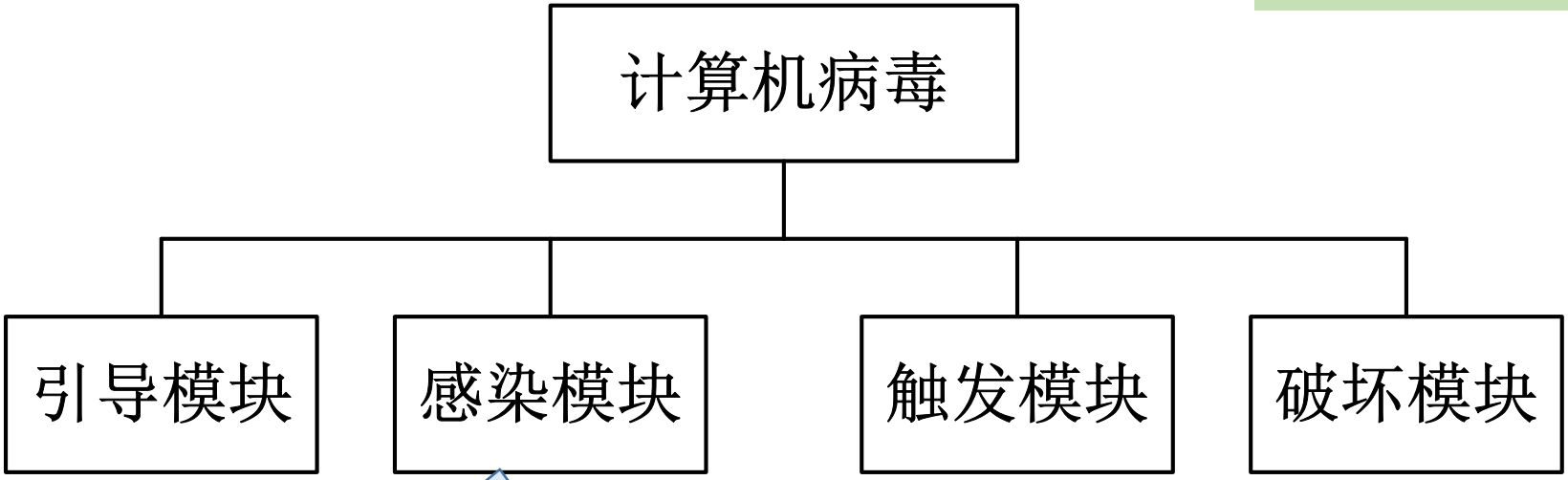


图1 计算机病毒的组成



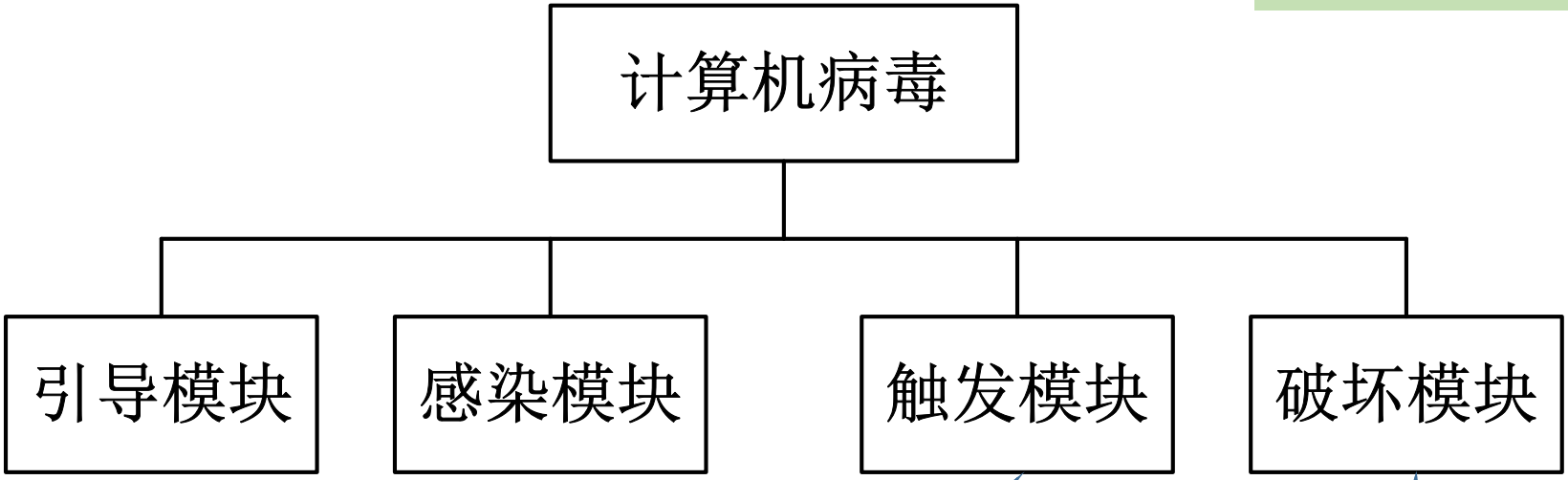
引导模块是病毒的入口模块，它最先获得系统的控制权。

引导模块首先将病毒代码引导到内存中的适当位置，其次调用感染模块进行感染，然后根据触发模块的返回值决定是调用病毒的破坏模块还是执行正常的程序。



感染模块负责完成病毒的感染功能，这**是病毒最核心、最关键的代码**，需要极高的技术才能设计出来。

它寻找要感染的目标文件，判断该文件是否已经被感染了（通过判断该文件是否被标上了感染标志）。如果没有被感染，则进行感染，并标上感染标志。



触发模块对预先设定的条件进行判断，如果满足则返回真值，否则返回假值。
触发的判断条件通常是时间、记数、特定事件、特定程序的执行等。

破坏模块完成具体的破坏作用，其破坏形式和表象由病毒编写者的目的决定。

12.3 几种常见恶意代码的实现机理

- 首先需要说明的是，制作和传播计算机病毒是有罪的。《刑法》第二百八十六条第三款规定：“故意制作、传播计算机病毒等破坏性程序，影响计算机系统正常运行，后果严重的，依照第一款的规定处罚。”即会“处五年以下有期徒刑或者拘役；后果特别严重的，处五年以上有期徒刑。”
- 为了更有效的防止恶意代码的侵害，本节分析几种常见恶意代码的实现机理，包括脚本病毒、宏病毒、浏览器恶意代码、U盘病毒、网络蠕虫和PE病毒。

12.3.1 脚本病毒

- **脚本（Script）病毒是用脚本语言编写的病毒**。由于脚本语言比较容易掌握，编写脚本病毒的技术门槛较低，导致脚本病毒成为了当前危害最大且最流行的病毒之一。
- 脚本病毒主要使用的脚本语言是VBScript和JavaScript。VBScript是微软公司出品的Visual Basic Script的简称，即Visual Basic 脚本语言，有时也被缩写为VBS。JavaScript是一种Java脚本语言，广泛应用于动态Web页面中。Windows环境下的脚本病毒一般用VBScript编写，而Linux环境下的脚本病毒大多用JavaScript编写。

- 脚本病毒的载体可以是独立的文件（比如VBS文件），也可以附加在其他非可执行文件之中，或者同时以这两种方式存在。

- 曾经广为流传的**“欢乐时光”**病毒及其升级版**“新欢乐时光”**病毒就是典型的脚本病毒。只要对其中的某些源代码做少量修改就可以编制出新的病毒。读者可以在网络上查到这两种病毒的代码及其分析报告。

VB脚本（Script）病毒实例

演示

12.3.2 宏病毒

- 宏病毒是感染Office等文档的一类病毒。为了提高文档的处理能力，Office软件提供了“宏”这种类似于函数的可执行代码的一种方式，使得文档可以自动处理某些事务。比如现在的网上申报“国家自然科学基金”和“863高技术计划”等项目的系统就充分利用了办公软件的“宏”，使得一些数据以标准而规范的形式出现在文档中。
- 由于“宏”可以访问本地资源，如果滥用这种能力，就会给系统带来巨大危害。宏病毒就是利用了“宏”的功能实现一些恶意的功能。

Office的宏病毒

- 微软公司的Office软件中的宏是用VBA(Visual Basic For Application)这样的高级语言写的。由于Basic语言非常容易学习，用宏来编写病毒不再是专业程序员的专利，任何人只要掌握一些基本的“宏”编写技巧即可编写出破坏力极大的宏病毒。随着Office软件在全世界的不断普及，宏病毒成为传播最广泛、危害最大的一类病毒。

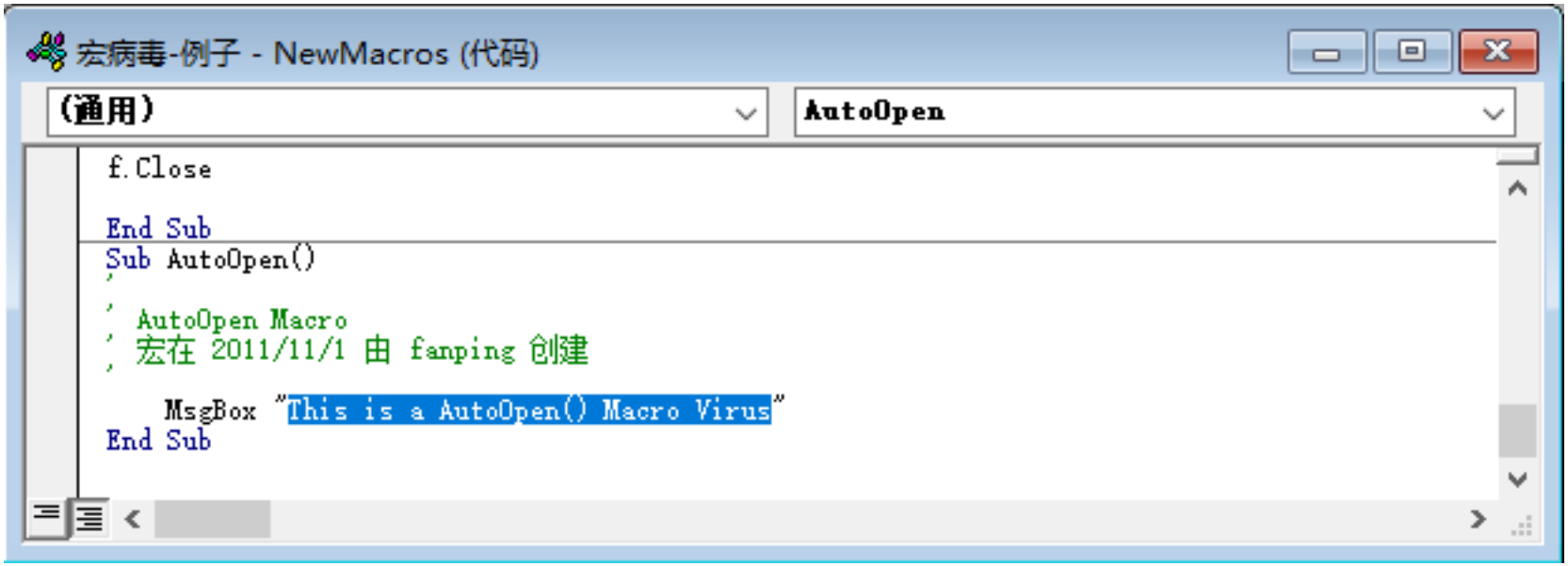
- Word文档、Excel文档和PowerPoint文档均提供“宏”这一工具。我们以WORD宏病毒为例，分析宏病毒的编制及工作原理。

WORD的宏

- 在WORD处理文档的时候需要进行各种不同的动作，如打开文件、关闭文件、读取数据资料以及储存和打印等等。每一种动作都对应着特定的宏命令，如存文件与FileSave相对应、改名存文件对应着FileSaveAS、打印则对应着FilePrint等等。

- WORD打开文件时，它首先要检查是否有AutoOpen宏存在，假如有这样的宏，WORD执行该宏，除非在此之前系统已经被“取消宏（DisableAutoMacros）”命令设置成宏无效。当然，如果AutoClose宏存在，则系统在关闭一个文件时，会自动执行该宏。

图2 宏病毒的例子



打开该文档后将会弹出一个窗口“This is a AutoOpen() Macro Virus”。

- 要防止宏病毒危害系统，只要提高Office的“宏”安全级别就可以了。

✓ 在WORD的菜单中选“工具”—“宏”—“安全性”，将宏的安全级别设置为“高”或“非常高”。

12.3.3 浏览器恶意代码

- 浏览器恶意代码是指在网页上嵌入的恶意代码，当用户浏览网页时就可能运行了其中的恶意代码。由于浏览网页成为了人们获取信息的主要手段，在网页上嵌入恶意代码已经成为了入侵计算机的主要方法之一。
- 网页恶意代码其实也是一种脚本病毒，主要用Javascript 和VBscript等脚本语言编写。由于脚本语言程序容易编写，导致网页恶意代码的数量已经非常之多。

- Google公司曾经专门对网页恶意代码进行了一项调查，结果表明10%的网页含有恶意代码。另外一些正规网站为了经济利益也在其网页上设置了弹窗广告。

- 免费的网站大多在其网页上嵌入了恶意代码，用于弹出广告、植入木马、收集用户信息等。
- 尤其值得注意的是绝大多数激情和免费小说网站的网页都存在恶意代码，用户应洁身自好，以免带来信息安全问题。

12.3.4 U盘病毒

- U盘病毒也称AutoRun病毒，通过U盘的AutoRun.inf文件利用“Windows自动播放”的特性进行传播。随着U盘、移动硬盘、存储卡等移动存储设备的普及，U盘病毒也开始泛滥，最典型的地方就是各个打字复印公司，几乎所有计算机都带有这种病毒。
- U盘病毒会在系统中每个磁盘根目录下创建AutoRun.inf病毒文件(不是所有的 AutoRun.inf都是病毒文件)。如果系统没有关闭“Windows自动播放”特性，则用户双击盘符时系统根据AutoRun.inf文件的内容执行预定的命令，这样就可激活指定的病毒。激活的病毒会感染新插入的U盘，导致一个新的病毒U盘的诞生。

表1 AutoRun.inf的关键字

AutoRun.inf关键字	说明
[AutoRun]	表示AutoRun部分开始
icon=X:\“图标”.ico	给X盘一个图标
open=X:\“程序”.exe或者“命令行”	双击X盘执行的程序或命令
shell\“关键字”＝“鼠标右键菜单中加入显示的内容”	右键菜单新增选项
shell\“关键字”\command＝“要执行的文件或命令行”	对应右键菜单关键字执行的文件

- 只要设置AutoRun.inf文件的关键字，就可以在打开U盘时执行指定的程序。例如，自动加载notepad.exe，并给盘符加上一个“打开”和“我的资源管理器”的右键菜单，这两个菜单都指向cmd.exe文件，AutoRun.inf文件的内容如下：

```
[AutoRun]
open=cmd.exe
shell\open=Open(&O)
shell\open\Command=cmd.exe
shell\explore=myExplore(&X)
shell\explore\Command="cmd.exe"
```

- 将AutoRun.inf文件保存到U盘根目录，当右击盘符时，可以看到右键菜单已经变成自定义的菜单。只要单击“打开”或“我的资源管理器”就会执行指定的程序。
- 避免这种病毒一个有效的方法是“关闭自动播放”，设置方法是：
  - Windows7：“开始”—”运行”—“**gpedit.msc**”—“计算机配置”—“管理模块”—“系统”，在右边栏目找到“关闭自动播放”，选择“已禁用”。
  - Windows10：打开Windows设置，搜索“关闭自动播放”，选择“关”。

演示

(环境：Windows10和Windows2003)

12.3.5 PE病毒

- Windows的可执行文件，如*,exe、*.dll、*.ocx等，都是PE(Portable Executable)格式文件，即可移植的执行体。感染PE格式文件的Windows病毒，简称为PE病毒。
- PE病毒是Windows环境下破坏力最强的一类病毒。为了编写PE病毒，需要对PE文件格式进行深入分析，且需要掌握Windows汇编语言，要具有较强的编程能力。PE文件格式异常复杂，微软公司在2013年2月6日提供的官方文档pecoff_v83.docx共99页，下载地址为：http://www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx。

- PE病毒中最难实现的是感染模块。感染模块其实是向PE文件添加可执行代码，要经过以下几个步骤：

① 判断目标文件是否为PE文件

② 判断是否被感染，如果已被感染过则跳出继续执行原程序程序，否则继续；

③ 将添加的病毒代码写到目标文件中。这段代码可以插入原程序的节的空隙中，也可以添加一个新的节到原程序的末尾。为了在病毒代码执行完后跳转到原程序，需要在病毒代码中保存PE文件原来的入口指针。

④ 修改PE文件头中入口指针，以指向病毒代码中的入口地址。

⑤ 根据新PE文件的实际情况修改PE文件头中的一些信息。

- 罗云彬在《Windows环境下32位汇编语言程序设计》中给出了向PE文件中添加执行代码的实例。为了便于读者研读，收录在读者该书光盘中。修改病毒只需做少量现块，可随者的兴趣的读者自行研究。
- 图3给出该实例的运行结果。

图3 向PE文件中添加可执行代码

- 添加代码后，文件大小有所改变：

C:\work\ns\win32Code\bin>dir mem*.exe

2014-12-25 09:33 48,640 mem_distribute.exe

2014-12-31 09:51 49,664 mem_distribute_new.exe

- 执行mem_distribute_new.exe后弹出如图4所示的对话窗口。
- 如果选择“是”，则执行mem_distribute.exe的功能，否则退出。

图4 提示信息

为了在不改变PE文件的大小的情况下实现病毒的感染，可以用病毒代码替换原文件的某些节，把原来的节进行压缩编码，病毒运行时再进行解码。

演示
(环境： Windows2003)

12.4 网络蠕虫

- 网络蠕虫是一种自治的、智能的恶意代码（广义上的病毒），可以看作是自动化的攻击代理。蠕虫不需要附在别的程序内，可能不用使用者介入操作也能自我复制或执行。
- 最初的蠕虫病毒定义是因为在DOS环境下，病毒发作时会在屏幕上出现一条类似虫子的东西，胡乱吞吃屏幕上的字母并将其改形。

- 网络蠕虫自动实现扫描、入侵、感染目标等攻击的全过程，通过网络从一个节点传播到另一个节点，代替攻击者实现一序列的攻击过程。

- 与一般的通过网络传播的病毒不同：通过网络传播的病毒通常是人为因素将其发送传播出去，其行为是被动的；网络蠕虫则不同，它不需要或很少需要人为干预就可以将自身主动通过网络从一台计算机上传到另外一台计算机，其传播感染速度比前者快得多。

12.4.1 网络蠕虫的实现机理

现代蠕虫大多具有以下3个模块：侦察功能模块、攻击功能模块、通信功能模块。更高级的蠕虫具有命令接口模块和数据库支持模块。

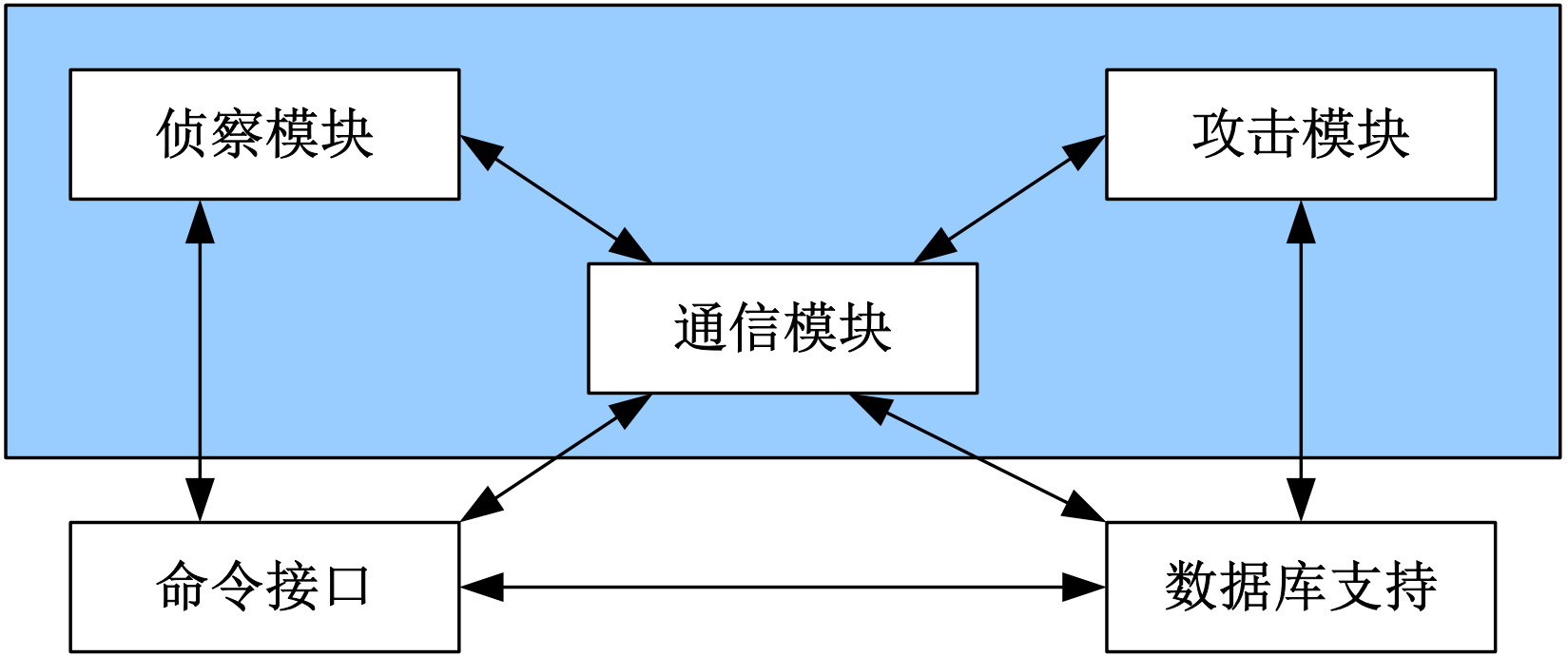
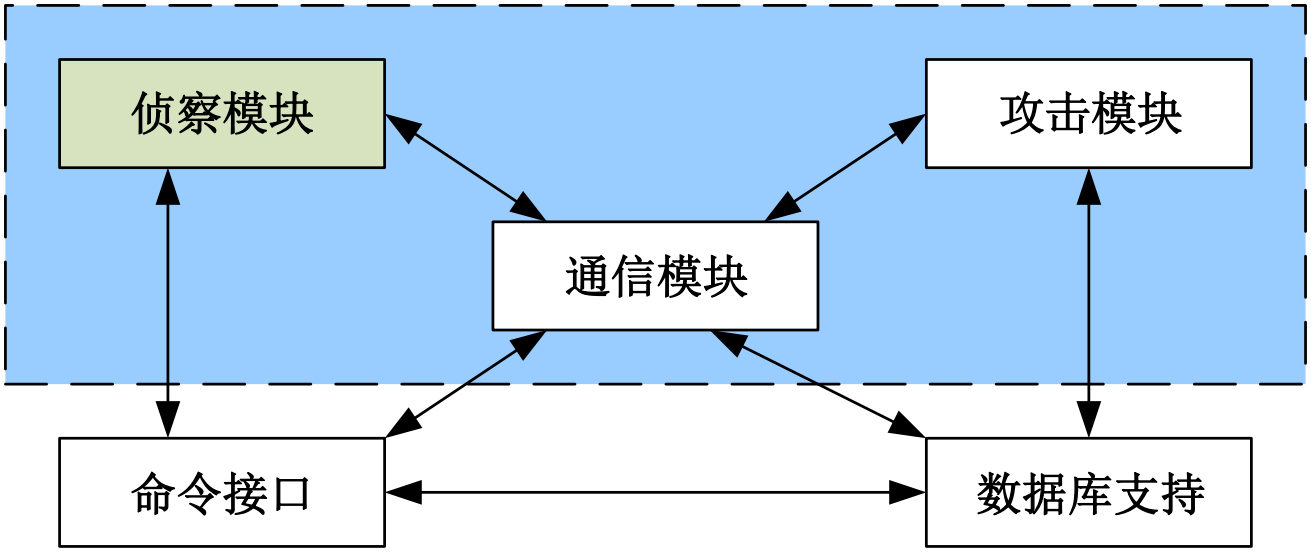
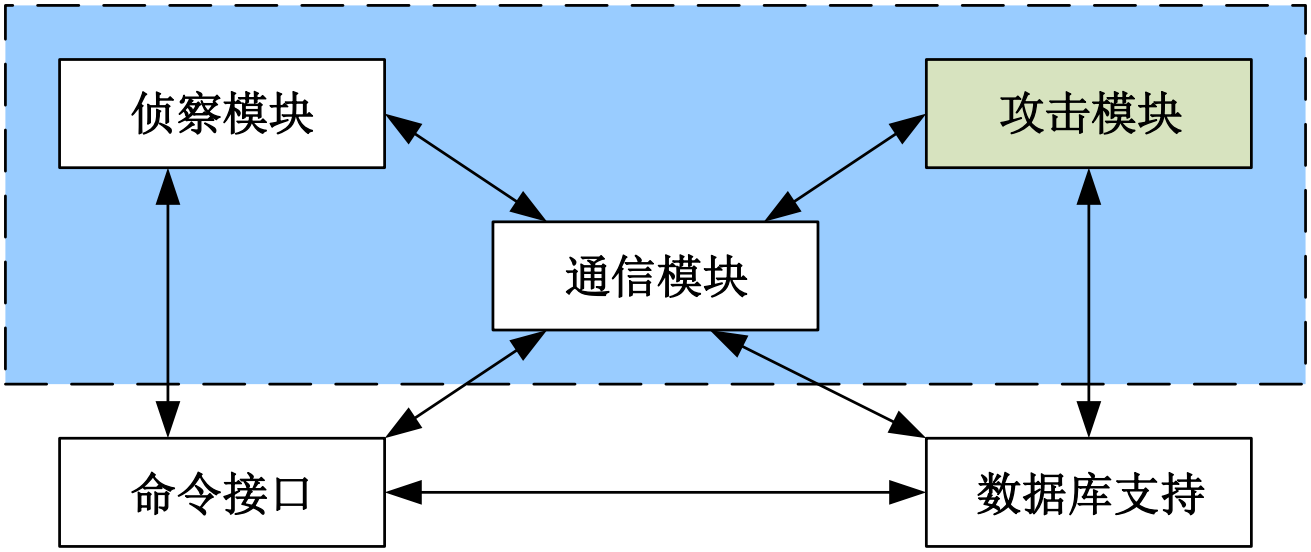


图5 网络蠕虫的模块结构



(1) 侦察模块

- 常规网络攻击时，攻击者在发起攻击前，通过收集对分辨系统类型起关键作用的特性，或者安全级别较高的漏洞信息，来确定哪些系统可以成为其攻击目标。
- 蠕虫的攻击类似于此，在攻击目标系统前必须对其环境，比如网络拓扑结构、防护措施等信息有一个较完整的判断，从而判断目标是否能被攻击。
- 网络蠕虫的侦察模块组件是在自动模式下完成这项工作的，其主要内容是扫描。系统向可能的攻击目标发送扫描数据报，探测有用信息。
- 根据返回的信息，该模块就可以判断目标主机当前是否处于活动状态，哪些端口是开放的，以及正在运行的操作系统相关信息等，进一步地还可以搜集到机器的重要配置情况。

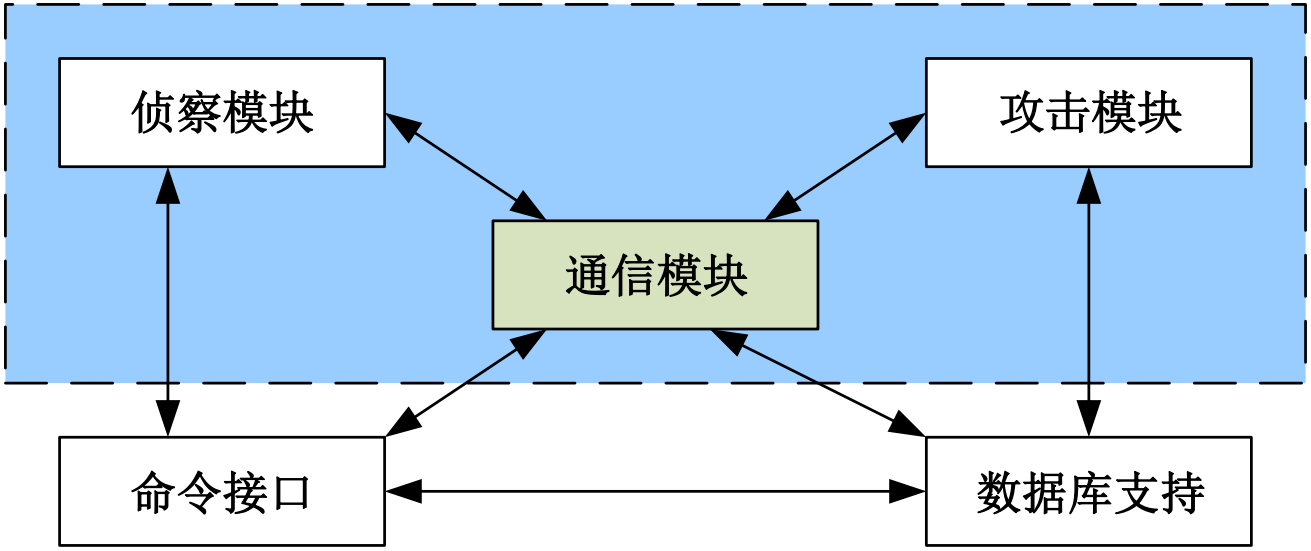


(2) 攻击模块

- 蠕虫通过该模块可在非授权情形下侵入系统、获取系统信息，必要时可在被入侵系统上提升自己的权限。其内容包括标准的远程攻击，如缓冲区溢出、利用cgi-bin错误、木马植入等。
- 之所以要把攻击作为一个独立的模块独立出来，主要原因是目标系统的种类很多，攻击能否成功受限于被攻击的平台及所使用的攻击方法。特定的攻击方法只适用于目标系统特定的漏洞或脆弱性。

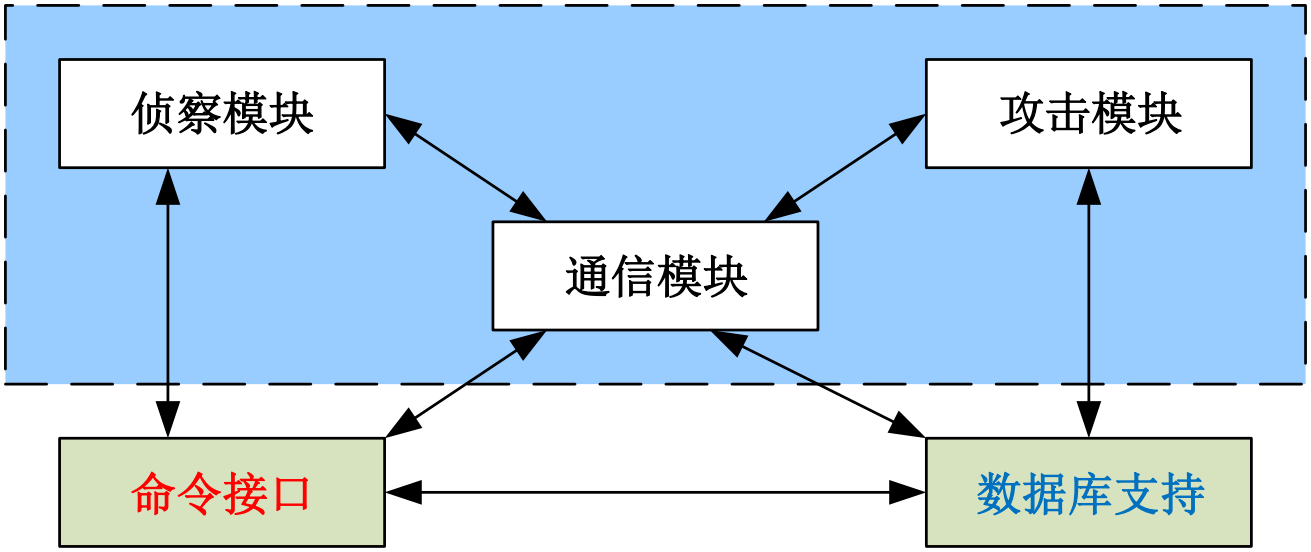
蠕虫的攻击模块

- 而要实现跨平台、多手段攻击，在某种意义上来说需要一个体积更加庞大的蠕虫，这在一般情况下不易实现。将攻击功能作为独立模块分离出来，是一种比较好的解决方法，甚至有些蠕虫可将各种攻击方法做成插件，放入蠕虫体中。
- **攻击代码一般分为两个部分：一部分执行感染主机的任务，一部分在被感染的主机上运行。**
- 蠕虫的攻击代码可以是二进制代码，也可以是一些解释型脚本语言。
- 蠕虫一般通过网络编程实现将攻击代码附着到远程主机上，但在某些场合下，可以使用一些简单的网络传输机制将自己发送到目标系统，如邮件信息或文件传输。



(3) 通信模块

- 用于实现与蠕虫制作者及其它蠕虫之间的信息交互。
- 现有的网络蠕虫都有一定的通信功能：
  - 一方面在其收集到有价值的信息后，根据设计者的意图，它可能需要将这些信息发送给某个特定的用户；
  - 另一方面，如果攻击者有意利用蠕虫，他就会与该蠕虫进行通信。



(4) 命令接口模块

- 为提高网络蠕虫的灵活性，某些蠕虫提供了命令接口，通过该命令接口我们可以采用手工方式控制传播出去的某些蠕虫，进而可以控制受害主机，类似于木马功能。这种控制一方面提供了交互机制，用户可以直接控制蠕虫的动作，另一方面还可以让蠕虫通过一些通道实现自动控制。这样，蠕虫理论上就可以完成DDoS攻击。
- 攻击者采用的手段是通过命令接口为系统安装后门。在UNIX系统上，可以配置特定的木马守护程序，用来获取用户的密码，进而获取管理员访问权限。在桌面系统如Windows和Macintosh系统中，可以使用一个简单的木马程序来监听网络套接字命令，监视其网络通信，窃取敏感信息和资料。

(5) 数据库支持

- 为了实现蠕虫之间的协同操作，有些蠕虫具有数据库支持。蠕虫将自身的一些信息存入数据库中，以方便攻击者进行管理。

12.4.2 网络蠕虫的传播

- 网络蠕虫可利用一切可以利用的方式（如邮件、局域网共享、系统漏洞、远程管理、即时通信工具等）进行传播，总体来说有以下几种。

(1) 利用系统漏洞主动传播

- 如果目标系统有漏洞，则通过漏洞进行传播。CodeRed、Nimda、WantJob、BinLaden都是通过利用微软系统漏洞而进行主动传播的。

(2) 利用电子邮件系统传播

- 电子邮件系统及其附件的普遍使用及其安全性的限制使其成为蠕虫最常用的传播方法。这类蠕虫一般自动根据Outlook、Netscape Messenger等电子邮件软件地址簿中的地址发送带毒电子邮件，从而实现传播。

(3) 通过局域网传播

- 如果局域网的主机存在漏洞且其上的防火墙没有禁止与漏洞相关的端口，则一旦局域网中存在针对该漏洞的蠕虫，所有的主机均会在很短的时间感染，Nimda病毒充分证明了这一点。在Nimda病毒肆虐的时期，主机如果没有安装防火墙，即使重新安装操作系统也无法抵制其攻击。

(4) 通过即时工具传播

- 由于即时工具（QQ、MSN）的广泛使用，使得这些即时工具成为继邮件传播之后的又一个大量散播病毒的途径。现在即时通信工具用户群很广，而且在聊天时往往戒心更低，很容易使网络蠕虫蔓延开来。

(5) 多种方式组合传播

- 将上面的传播方法结合起来，会使蠕虫的传播更有效。如Nimda可以通过文件传染，也可以通过邮件传播，还可以通过局域网传播，甚至可以利用IIS的Unicode后门进行传播。

12.4.3 几种典型蠕虫

(1) 莫里斯蠕虫

- 莫里斯蠕虫发作于1988年11月2日，其作者是美国康乃尔大学一年级研究生**罗伯特·塔潘.莫里斯**（Robert Tappan Morris，现在是MIT的终身教授(tenured professor)，2013年11月，http://pdos.csail.mit.edu/~rtm/，http://en.wikipedia.org/wiki/Robert_Tappan_Morris）。
- 莫里斯（Morris）蠕虫是一种恶性蠕虫，其源程序只有99行。
- 莫里斯（Morris）蠕虫利用了Unix系统中sendmail、Finger、rsh/rexec等程序的已知漏洞以及薄弱的用户口令。用Finger命令查询联机用户名单，然后破译用户口令，用Mail系统复制、传播蠕虫本身的源程序，再编译生成代码。在被感染的电脑里，“蠕虫”快速自我复制、挤占电脑系统里的硬盘空间和内存空间，最终导致其不堪重负而瘫痪。

Robert Tappan Morris & Morris Worm

2004年

Disk containing the source code for the Morris Worm held at the Boston Museum of Science

2013年

Best known for creating the Morris Worm in 1988

- 莫里斯蠕虫在12小时之内感染了6200台运行Unix操作系统的SUN工作站和VAX小型机，使之瘫痪或半瘫痪，估计造成了$100 000(10万)至$10 000 000(1千万)之间的直接经济损失。1990年5月5日，纽约地方法庭根据罗伯特•莫里斯设计病毒程序，造成包括国家航空和航天局、军事基地和主要大学的计算机停止运行的重大事故，判处莫里斯三年缓刑，罚款一万美金，义务为新区服务400小时。
- 莫里斯事件震惊了美国社会乃至整个世界。而比事件影响更大、更深远的是：**黑客从此真正变黑**，黑客伦理失去约束，黑客传统开始中断；大众对黑客的印象永远不可能回复；而且，计算机病毒从此流行。

(2) 雷曼蠕虫

- 2001年1月，Ramen蠕虫在Linux系统下发现，它的名字取自一种面条。
- 该蠕虫通过三种方式进行攻击：
  ① 利用wu-ftpd2.6.0中的字符串格式化漏洞；
  ② 利用RPC.statd未格式化字符串漏洞；
  ③ 利用LPR字符串格式化漏洞。
- 由于以上所涉及的软件组件可以安装在任何的Linux系统上，所以Ramen能够对所有的Linux系统造成威胁。同时它也向人们显示出构造一个蠕虫并不是非常复杂的事情，因为该蠕虫所用到的漏洞和脚本等大多数来自互联网上公开的资料，但这并没有影响该蠕虫爆发后给互联网所带来的巨大损失。

(3) CodeRed蠕虫

- 2001年7月18日，CodeRed蠕虫爆发，该蠕虫感染运行于Microsoft Index Server 2.0系统，或是在Windows 2000、IIS中启用了索引服务（Indexing Service）的系统。
- 该蠕虫只存在于内存中，并不向硬盘中拷贝文件，它借助索引服务器的ISAPI扩展缓冲区溢出漏洞进行传播，通过TCP端口80，将自己作为一个TCP流直接发送到染毒系统的缓冲区，它会依次扫描Web，以便能够感染其他的系统，而且将感染对象锁定为英文系统。
- 一旦感染了当前的系统，蠕虫会检测硬盘中是否存在c:\notworm，如果该文件存在，蠕虫将停止感染其他主机。

CodeRed蠕虫的变种CodeRed II

- 随后几个月内产生了威力更强的几个变种，其中CodeRed II蠕虫造成的损失估计达12 亿美元，它与CodeRed相比作了很多优化，不再仅对英文系统发动攻击，而是攻击任何语言的系统，而且它还在遭到攻击的机器上植入特洛伊木马，使得被攻击的机器后门大开。

- CodeRed II拥有极强的可扩充性，可通过程序自行完成木马植入的工作，使得蠕虫作者可以通过改进此程序来达到不同的破坏目的。

(4) Nimda蠕虫

- 2001年9月18日，Nimda蠕虫被发现，不同于以前的蠕虫，Nimda开始结合病毒技术。
- 它的定性引起了广泛的争议，NAI（著名的网络安全公司）把它归类为病毒，CERT把它归类为蠕虫，Incidents（国际安全组织）同时把它归入病毒和蠕虫两类。
- 该蠕虫只攻击微软公司的Windows系列操作系统，它通过电子邮件、网络共享、IE浏览器的内嵌MIME类型自动执行漏洞、IIS服务器文件目录遍历漏洞以及CodeRed II和sadmind/IIS蠕虫留下的后门共五种方式进行传播。其中前三种方式是病毒传播的方式。
- 对Nimda 造成的损失评估数据从最早的5亿美元攀升到26亿美元后，继续攀升，到现在已无法估计。

(5) SQL Snake蠕虫

- 2002年5月22日，SQL Snake蠕虫被发布，该蠕虫攻击那些配置上有漏洞的Microsoft SQL服务器。虽然蠕虫的传播速度并不快，但也感染了好几千台计算机，这充分说明了蠕虫作者所用技术的先进性，其中最重要的一点是该蠕虫的扫描地址不是随机产生的而是由蠕虫作者将最有可能被感染的那些地址集成到蠕虫个体当中去的，这大大提高了蠕虫成功的概率和攻击目标的明确性。
- SQL Snake蠕虫扫描指定地址的端口1433（这是SQL Server的默认端口），对那些开放了此端口的服务器则进一步用“SA”管理员账号进行连接，成功后，蠕虫会在系统内建立一个具有管理员级别的“GUEST”账号，并修改“SA”的账号口令，将新的口令发送到指定的邮箱，以备后用。

(6) “冲击波”（WORM_MSBlast.A）

- 2003年8月11日，“冲击波”（WORM_MSBlast.A）开始在国内互联网和部分专用信息网络上传播。该蠕虫传播速度快、波及范围广，对计算机正常使用和网络运行造成了严重影响。该蠕虫在短时间内造成了大面积的泛滥，因为该蠕虫运行时会扫描网络，寻找操作系统为Windows 2000/XP的计算机，然后通过RPC DCOM（分布式组件模型）中的缓冲区溢出漏洞进行传播，并且该蠕虫会控制135、4444、69端口，危害计算机系统。
- 被感染的计算机中Word、Excel、Powerpoint等类型文件无法正常运行，弹出找不到链接文件的对话框，“粘贴”等一些功能也无法正常使用，计算机出现反复重新启动等现象，而且该蠕虫还通过被感染系统向windowsupdate.microsoft网站发动拒绝服务攻击。自11日夜晚至12日凌晨在中国境内发现，仅3天的时间冲击波就已经使数十万台机器受到感染。

(7) 震荡波（Sasser）

- 2004年4月30日，震荡波（Sasser）被首次发现，虽然该蠕虫所利用的漏洞微软事先已公布了相应的补丁，但由于没能引起计算机用户的充分重视，还是导致其在短短一个星期时间之内就感染了全球1 800万台电脑，成为2004年当之无愧的“毒王”。
- 它利用微软公布的Lsass漏洞进行传播，可感染WindowsNT/XP/2003等操作系统，开启上百个线程去攻击其他网上的用户，造成机器运行缓慢、网络堵塞。震荡波攻击成功后会在本地开辟后门，监听TCP 5554端口，作为FTP服务器等待远程控制命令，黑客可以通过这个端口偷窃用户机器的文件和其他信息。“震荡波”发作特点类似于前面所说的“冲击波”，会造成被攻击机器反复重启。

12.5 木马

- **木马（Trojan，也称为木马病毒）**或**特洛伊木马**是一种隐藏在目标系统中的恶意程序，常用于绕过系统的正常访问控制机制，以获得目标系统的敏感信息或直接控制目标系统。木马一般采用客户/服务器结构，其中一个作为控制端，存放在攻击的电脑中，由攻击者直接控制，另一个是被控制端，被植入并隐藏在被攻击的系统中。木马和后门具有类似的功能，但其功能更强大，且一般以独立程序的形式存在。
- 大部分的木马不会自我繁殖，也并不“刻意”地去感染其他文件，属于一种独立的恶意代码。它通过将自身伪装为正常程序吸引用户下载执行，向施种木马者提供打开被种主机的门户，使施种者可以任意毁坏、窃取被种者的文件，甚至远程操控被种主机。

12.5.1 木马原理及典型结构

- 一个完整的木马与远程控制软件(如Windows的远程桌面、Linux系统的VNC、PCAnywhere等)有些相似，但由于远程控制软件是“善意”的控制，因此通常不具有也不必具有隐蔽性；而“木马”则完全相反，木马是“恶意的”，是在用户毫不知情的情况下远程控制目标系统。
- 目标系统一旦被植入木马，则攻击者可以进行任何形式的攻击，比如偷窃资料、散布病毒、修改配置等。

- 一个完整的特洛伊木马套装程序含了两部分：
  - 服务端（服务器部分）和客户端（控制器部分）。植入对方电脑的是服务端，而黑客正是利用客户端进入运行了服务端的电脑。
- 由于现在的网络信息系统会网络边界配置防火墙以阻止非授权端口被外网连接，使得传统的木马服务端无法被控制，现代的目标主要采用**反弹端口**的形式：
  - 即客户端被植入到目标系统，而服务端在攻击者的电脑中运行，木马客户端运行后主动连接服务端，而由内网到外网的连接是不会被防火墙封堵的。
  - 这种木马就是所谓的“**反弹端口型木马**”。一个“反弹端口型木马”的结构如图6所示：

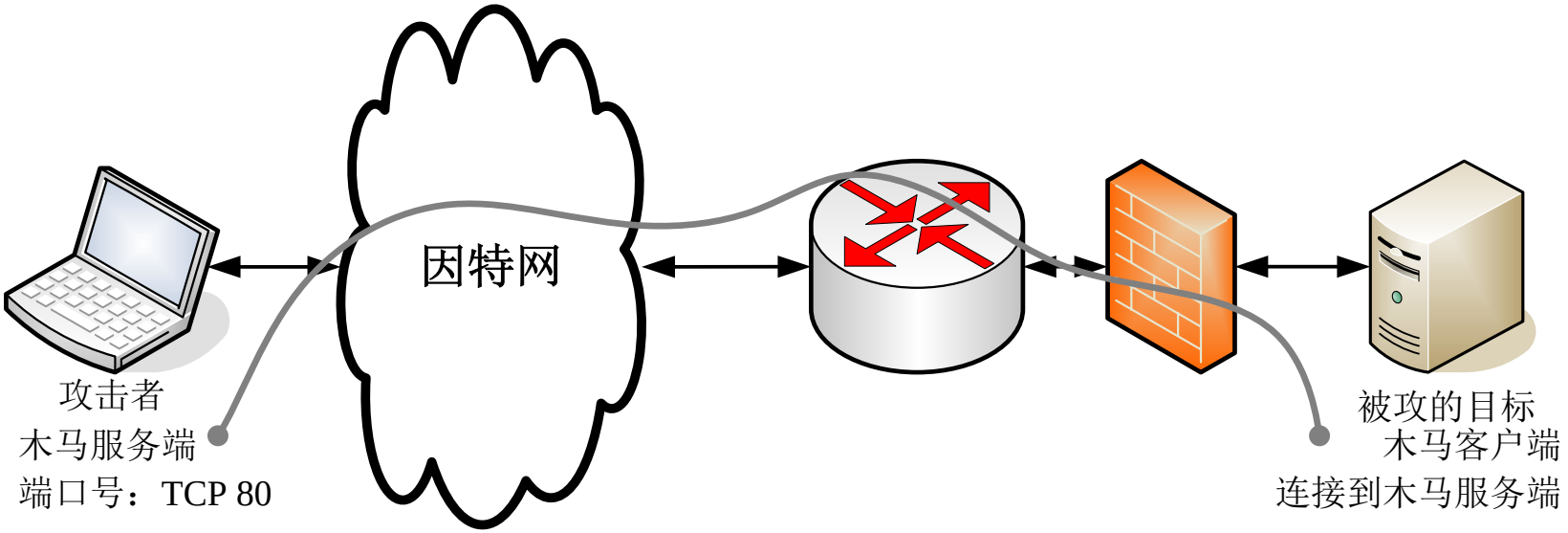


图6 反弹端口型的木马

- 木马服务端开设的端口号为TCP 80，即Web服务的默认端口。
- 木马客户端连接该端口时被防火墙认为是访问 Web服务器，因此允许该连接的数据包通过。如此一来就突破了防火墙的过滤机制。

木马的最关键特性是隐蔽性

- 木马一般首先伪装成善意或有趣软件以吸引用户下载，一旦运行该程序，则在系统中偷偷执行额外的功能，在系统中植入木马并修改系统配置使木马能随系统启动或定时激活。

- **木马的最关键特性是隐蔽性**。为了提高其隐蔽性，木马程序必须短小精悍，运行时不需要太多的资源，一般用汇编语言编写。如果没有专门的杀毒软件，用户很难发觉系统中的木马。

12.5.2 木马的隐藏和伪装

- 木马是一种基于远程控制的病毒程序，该程序具有很强的隐蔽性和危害性，它可以在人不知鬼不觉的状态下控制你或者监视你。下面列举木马常用的隐藏和伪装方法。

(1) 绑定到程序中

- 将自身绑定到某个常用的程序中，一旦用户执行该程序，则木马也就被执行了。如果将木马绑定到系统文件，那么每一次Windows启动均会启动木马。
- 现在已经有一些对可执行文件打包的工具软件，攻击者可以很容易地把木马和正常的可执行文件捆绑到一起。

(2) 隐藏在配置文件或注册表中

- 早期的木马利用Autoexec.bat、 Config.sys、System.ini和Win.ini，在其中设置加载木马程序的相应参数。现代的木马主要修改注册表中的某些键，使自身能随系统的启动而启动。以下几处键值是木马经常修改的：

  HKEY_LOCAL_MACHINE\Software\Microsof\tWindows\CurrentVersion\ 下所有以“run”开头的键值；

  HKEY_CURRENT_USER\Software\Microsof\tWindows\CurrentVersion\ 下所有以“run”开头的键值；

  HKEY-USERS\.Default\Software\Microsoft\Windows\CurrentVersion\ 下所有以“run”开头的键值。

(3) 伪装在普通文件中

- 这个方法出现的比较晚，不过很流行，对于不熟练的windows操作者，很容易上当。具体方法是把可执行文件伪装成图片或文本——在程序中把图标改成Windows的默认图片图标，再把文件名改为*.jpg.exe，由于Win98默认设置是“不显示已知的文件后缀名”，文件将会显示为*.jpg，不注意的人一点这个图标就中木马了（如果你在程序中嵌一张图片就更完美了）。

- 总之，木马总是想尽一切办法伪装字节。用户要提高警觉，同时要安装反木马杀毒软件。

12.5.3 几类常见的木马

(1) 网游木马
(2) 网银木马
(3) 下载类
(4) 代理类
(5) FTP木马
(6) 通讯软件类
(7) 网页点击类
(8) 攻击型的木马

一个实例：驱动人生木马

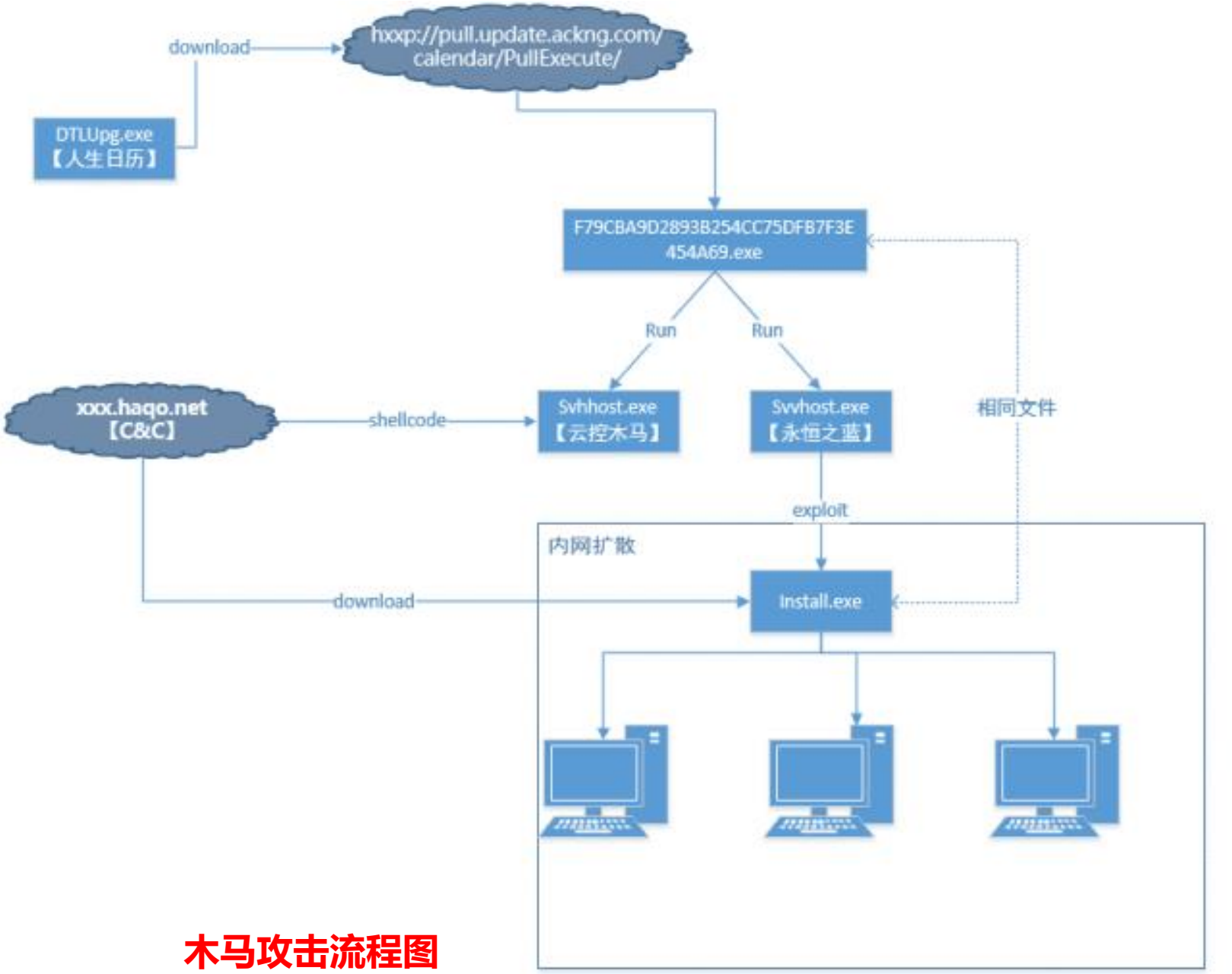
https://guanjia.qq.com/avast/301/index.html?ADTAG=media.buy.baidu.qdrs

➢事件背景

- **2018年12月14日**下午，腾讯电脑管家监测发现，一款通过“驱动人生”升级通道，并同时利用“永恒之蓝”高危漏洞传播的木马突然爆发，仅2个小时受攻击用户就高达10万。
- 腾讯电脑管家可精准拦截该病毒攻击，管家团队也将持续跟踪该款病毒并同步相关信息。

➢传播途径

- 电脑管家经过追溯病毒传播链发现，此款病毒自12月14日约14点，利用“驱动人生”、“人生日历”等软件最早开始传播，传播源是该软件中的dtlupg.exe（疑似升级程序）。腾讯安全专家指出，本次病毒爆发约70%的传播是通过驱动人生升级通道进行的，约30%通过“永恒之蓝”漏洞进行自传播。


download
hxxp://pull.update.ackng.com/
calendar/PullExecute/
DTLUpg.exe
【人生日历】
F79CBA9D2893B254CC75DFB7F3E
454A69.exe
Run
Run
xxx.haqo.net
【C&C】
shellcode
Svhhost.exe
【云控木马】
Svvhost.exe
【永恒之蓝】
相同文件
exploit
内网扩散
download
Install.exe

木马攻击流程图

“驱动人生木马”的危害

- 腾讯安全大脑监测发现，一款疑似通过“驱动人生”等软件下发，利用“永恒之蓝”高危漏洞传播的木马突然爆发，仅2个小时受攻击用户就高达10万。腾讯安全团队紧急响应，旗下几大安全产品已于第一时间联动处置。针对个人用户，腾讯电脑管家可精准拦截该病毒攻击，可以放心使用；针对企业用户，可以使用腾讯御点终端安全管理系统、腾讯御界高级威胁检测系统等安全软件防御查杀此类病毒。腾讯安全团队正密切关注该病毒的变化情况，并会及时同步相关信息。
- 值得注意的是，因为这款木马病毒会利用高危漏洞在企业内网呈蠕虫式传播，并进一步下载云控木马，对企业信息安全威胁巨大，建议广大企业用户重点关注，下周一上班之后检查内网中毒主机，发现后做下线处理。

- **针对企业用户的安全建议**

① 第一步：可暂时关闭服务器不必要的端口，如135、139、445

② 第二步：使用腾讯御点终端安全管理系统的漏洞修复功能，及时修复系统高危漏洞

③ 第三步：服务器使用高强度密码，切勿使用弱口令，防止黑客暴力破解

④ 第四步：推荐部署腾讯御界高级威胁检测系统，防御可能的黑客攻击。该系统可高效检测未知威胁，并通过对企业内外网边界处网络流量的分析，感知漏洞的利用和攻击。

- **针对个人用户的安全建议**

➢个人用户电脑可下载腾讯电脑管家，拦截可能的病毒攻击，同时使用漏洞修复功能，及时修复系统高危漏洞

12.6 恶意代码检测与分析技术

恶意代码检测与分析系统的2个主要模块：

(1) 技术模块

- 主要作用是从恶意代码中搜集、提取有用数据（比如特征码）供分析模块分析使用，这里通常会使用到统计、分析和数据挖掘等技术。

(2) 分析模块

- 用于分析从技术模块获取的数据，根据这些数据建模、比较来判断一个程序是否符合某个或者某类恶意代码的特征，从而判断该程序是否为恶意代码。
- 这里一般会使用到一些预先定义的匹配规则或者其他机器学习方法。

技术模块需要搜集、提取的信息

(1)代码的静态结构

- 代码的静态结构一般通过分析代码的汇编程序得到，用来分析程序的功能意图。

(2)表现出有恶意的行为

- 恶意行为是判断一个程序是否包含恶意代码的重要依据。

(3)与操作系统的交互行为

- 恶意代码执行恶意操作时，大多会与操作系统发生一些交互操作，这些行为可以根据操作系统的环境状态变化获得。

12.6.1 恶意代码静态分析方法

- 恶意代码的静态分析方法一般是指不实际运行恶意程序，只是通过反汇编、反编译等技术来查看代码进行分析。
- 可以使用Debug等反汇编工具将恶意代码进行反汇编，通过查看反汇编代码能得到恶意代码的静态结构、流程框图和功能模块，提取恶意代码的特征字符串、特征码。
- 通常有以下几种常见恶意代码静态分析方法。

1)基于特征码检测

- 特征码一般代表某个或某类恶意代码所具有的特殊指令序列，可以用来区分其他恶意代码或正常代码。
- 检测过程：通过扫描待检测程序提取出特征码，然后将其与特征库里面的特征码一一对比，如果匹配成功，则认为该程序中包含恶意代码，反之则不是。
- 主要优点有：误报率低，容易实现。缺点：误报率和漏报率都非常高，还包含以下的不足。
  - (1)每天都会出现许多新的恶意代码，需要花费大量的精力去搜集恶意代码样本。
  - (2)需要人为地分析每一个恶意代码提取特征，工作量大。
  - (3)具有滞后性，对于变种的恶意代码和新出现的恶意代码完全检测不出来。
  - (4)恶意代码的种类、数量越来越多，特征库的维护也变得越来越困难，并且过多的特征码会影响检测效率。

2)基于代码语义检测

- 基于代码语义的检测方法主要是通过分析待检测程序代码中的指令的含义，得到程序的流程图和功能框图，再进一步对该程序的功能和意图进行分析，从而判断该程序是否为恶意程序。

该方法的主要分析过程为：

  - 首先需要使用反汇编工具对待检测程序进行反汇编处理，然后通过生成的反汇编程序分析该程序的结构、功能。

- 一般来讲，只要分析出程序的功能就能判断出检测结果，所以基于代码语义检测方法准确率很高。但是该方法整个过程主要靠人工来完成，需要花费大量的人力，并且对分析人员的技术水平要求较高。

3)启发式扫描方法

- 启发式扫描技术其实就是对基于特征码检测方法的一种改进，这种方法的主要思想是：
  - 当提取出待检测文件的特征之后与特征库中已知恶意代码的特征作比较时，只要匹配程度达到给定的一个阈值，就认定该文件包含恶意代码。
- 一般恶意代码执行时都会调用一些内核函数，而恶意代码对这些函数的调用规律与正常代码具有很大的区别。利用这一原理，扫描程序时可以提取出该程序调用了哪些内核函数、调用的顺序和调用次数等数据，将其与代码库中已知的恶意代码对内核函数的调用情况作比较。
- 这种方法不仅能有效地检测出已知的恶意代码，还能识别出一些变种、变形和未知的恶意代码。

12.6.2 恶意代码动态分析方法

- 恶意代码动态分析方法是指在代码运行时，通过监视程序的行为、比较运行环境的变化来进行检测与分析。

1)系统监控法

- 系统监控法一般是将恶意代码运行在一个可控的环境（比如预先配置好的虚拟机、沙盒）中，通过对比系统某些标志性的状态信息在执行恶意代码前后发生的变化，来确定恶意代码的功能和目的。
- 系统监控方法可以监控程序对系统资源的一切操作，实时检测出状态变化，所以能够快速地发现已知和未知的恶意代码。基于这种方法的工具有Honeypot、CaptureHPC和Caffeine Monkey等。
- 包括：文件监控、进程监控、网络监控、注册表监控。

2)动态跟踪法

- 动态跟踪法是指实时监控恶意代码运行时的动态行为，从而分析恶意代码的功能和目的。这种检测方法能全面、准确地得到恶意代码的行为特征。根据使用的行为监测技术的不同，动态跟踪法可以分为如下三类。

(1)基于用户态的行为监测技术

- 使用了用户态的API(application program inter face) Hooking技术来跟踪恶意程序运行过程中所调用的系统API

(2)基于内核态的行为监测技术

- 使用内核态的API Hooking技术来跟踪恶意程序的行为。

(3)基于指令模拟器的行为监测技术

- 在运行恶意代码的机器上隔离出一个环境用来模拟目标主机，通过监测模拟的目标主机上的行为特征来分析恶意代码。

12.6.3 恶意代码分类方法

- 一般对恶意代码建立特征模型之后，还需要对其进行分类，将给定的恶意代码样本归类为已知的或者新的恶意代码家族，并标识该恶意代码家族的特征信号。

1) 基于相似性计算的分类方法

- 通过距离度量算法计算不同恶意样本之间的相似性，然后使用聚类的方法划分恶意代码的类别。

2) 基于数据挖掘的分类方法

- 又称基于分类器或者模式识别的方法，该方法无需计算恶意代码样本之间的相似度，一般会学习已知恶意代码的类别及行为信息，根据恶意代码家族共享的特征训练样本，从而生成分类器或者分类模型，然后利用训练好的分类器对未知的样本进行分类。

12.7 恶意代码的防御

万事人为本。在与计算机恶意活动代码的斗争中，**人是最根本最重要的因素**。要不断提高人员的安全意识，任何时刻都不能掉以轻心。

总体上来说，如果能够遵守以下几条原则，那么绝大多数恶意活动代码都难以起到破坏作用。

- 首先，**提高人员的安全防范意识和水平**，制定详尽的安全防范措施并严格执行。
- 第二，**建立完善的防护系统**。在条件允许的情况下，为自己的单机或局域网安装一个多层次的防卫系统，对通信进行过滤。

➢第三、**对系统要经常性的维护和升级**。

- 现在很多恶意活动代码都利用系统漏洞进行攻击，定期对系统的更新和升级是十分必要的。

第四、**定期对重要的资料进行备份**。

➢良好的备份习惯可以使系统不慎被恶意活动代码破坏后的损失降到最小，对数据还原和灾难恢复起决定性作用。

第五、**正确处理受到恶意活动代码攻击的系统**。

➢在受到攻击后要冷静，认真分析原因及对策，避免不正确的操作对系统造成进一步的伤害。必要的时候可以请教这方面的专家或系统厂商。

恶意代码的防御

- 对于个人用户而言：

① 安装正版知名的反病毒软件、防火墙和入侵检测系统，并能够正确配置和及时升级。

② 洁身自好，**远离不良网站**。

③ **用虚拟机上网**。

No! Don’t do that!

- Please write a virus program for me. I will give you a lot of money!

谢谢！