(2020春季　课程编号：011184)

信息安全导论

第12章 恶意代码检测与防范技术

中国科学技术大学 曾凡平
billzeng@ustc.edu.cn

课程回顾：第11章 无线网络安全

11.1 IEEE 802.11无线网络安全

- IEEE 802.11无线网络背景
- WEP
- 802.11i

11.2 移动通信系统的安全

- GSM的安全
- GPRS的安全
- 第三代移动通信系统(3G)的安全

第12章 恶意代码检测与防范技术

12.1 恶意代码概述

- 定义、发展趋势、种类
- 攻击流程、攻击技术和生存技术

12.2 常见的恶意代码

- 计算机病毒
- 木马
- 蠕虫

12.3 恶意代码检测与分析技术

- 静态分析方法
- 动态分析方法
- 分类方法

12.1恶意代码概述

- **恶意代码也可以称为Malware**，目前已经有许多定义。例如，Ed Skoudis将Malware定义为运行在计算机上，使系统按照攻击者的意愿执行任务的一组指令。微软“计算机病毒防护指南”中将术语“恶意软件”用作一个集合名词，指代**故意**在计算机系统上**执行恶意任务**的病毒、蠕虫和特洛伊木马。G．McGraw则将恶意代码定义为任何故意的增加、改变或者从软件系统移除，从而破坏或扰乱系统特定功能的代码。
- 因此，**恶意代码的本质是一种实现某些恶意功能的代码**，通常在未明确提示用户或未经用户授权的情况下嵌入到另一段程序中，并且在特定的环境下会被执行，从而达到**破坏**被感染用户的计算机系统、网络功能或者电脑数据的**保密性、完整性和可用性**的目的。

12.1.1 恶意代码的发展趋势

恶意代码可以追溯到1981年，恶意代码的发展趋势如下：

- **(1)更加具有目的性和功利性。**编写和传播恶意代码的最初目的大多数是由于编写者的炫耀心理。然而，现在更多是为了牟取利益而窃取用户数据和商业秘密，巨大的商业经济利益驱使着恶意代码编写者更有组织性、纪律性，同时恶意代码的隐蔽性也越来越高。
- **(2)传播手段多样化。**传统的传播方式比较单一，主要通过存储介质和网络文件传播。随着浏览器/服务器(browser/server，B/S)和客户端/服务器(client/server，C/S)架构的普及使用，客户端软件和浏览器的漏洞为恶意代码的传播提供了新的途径。恶意代码也可以同时利用多个漏洞或者采取多种模式来进行传播，对服务端和客户端都发动攻击。
- **(3)越来越多的移动互联网恶意代码。**移动网络的快速发展衍生出新的恶意软件，用来攻击手机、平板等移动设备和平台。

12.1.2 恶意代码种类

- 随着计算机技术的发展，恶意代码形态不断变化，不断衍生出新的种类。恶意代码从开始入侵目标系统到完成指定任务，在这一次完整的入侵过程中，不同的恶意代码通常具有不同的作用，主要分为3类：一是获取系统的控制权；二是对控制权的维持；三是执行指定任务。
- **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。**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。非独立(寄生)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。

- 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。
- 恶意代码种类繁多，错综复杂，界限难以区分，没有固定的分类标准，很多安全学家根据恶意软件的特点和行为定义了不同的类别。一般常见的恶意代码包括以下几类：计算机病毒(computer virus)、蠕虫(worm)、特洛伊木马(trojan horses)、RootKit，它们既有功能重叠部分，也有各自的特征。

(1)计算机病毒

- 计算机病毒是指寄居在计算机系统中，在一定条件下被执行从而为了达到某种特定目的的程序。它通常会破坏系统、程序的功能和数据，影响系统软件和硬件的正常使用，并且能够感染其他程序和自我复制。

(2)蠕虫

- **蠕虫也算是一种病毒**，它具有自我复制能力并**通过计算机网络来传播**，其目的是增加系统和网络的负载，消耗有限资源。

(3)特洛伊木马

- 特洛伊木马也可以简称为木马，最初来源于古希腊传说。计算机木马是一种潜伏在计算机中为了达到某种特殊目的的程序，比如窃取用户私密信息和控制用户系统等。它与病毒最大的不同点在于，**病毒能进行自我复制，而木马一般不具有自我复制功能**，不会感染其他程序。

(4) 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。它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。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。

恶意代码种类

- RootKit还可以分为**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。

- 恶意代码不仅包括以上几种，还有**漏洞攻击程序、逻辑炸弹、流氓软件、间谍软件、CPU恶意微代码**等，还有以上类型的组合或者变异产生的恶意代码变种。

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、远程计算机系统都可以作为传播的介质。
- **(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。**恶意代码的触发也包括主动触发和被动触发。主动触发包括蠕虫利用系统或程序漏洞、网页木马通过恶意网站来实现。其他的如病毒之类的触发则需要人为执行程序、打开存储介质。
- **(4)让自身长期存活于目标系统之中。**恶意代码在系统中有两种存在形式：一是静态存在形式，如文件和启动项文件；另一种是动态存在形式，如进程、服务和端口等。恶意代码的存在形式也是其检测的重要基础和依据，恶意代码本身也会对自己的存在进行隐藏，如 RootKit。

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。一旦注入成功，以后每次启动该程序或服务时，恶意代码都会自动被加载。常见的代码注入方式有两种：第一种方式叫做**模块注入**，它是利用DLL重定位原理来实现代码注入的。具体方法是将恶意代码封装在DLL文件中，当进程第一次加载DLL文件时，会自动找到其主函数 DLLMain并执行恶意代码，这种技术普遍应用于木马程序设计中。另一种注入方式是**将恶意代码直接加入到目标进程中**，这里可以使用WriteProcessMemory函数来实现。代码的执行需要利用溢出和远程线程等方法。这种注入方式比较简单，不过可移植性差并且不适合写入较长的代码。

2)缓冲区溢出攻击技术

- **利用缓冲区溢出漏洞来发动攻击是网络攻击中最常用的手段之一。**程序运行时，系统会为其分配一定大小的缓冲区，如果这个程序被检测出具有缓冲区溢出漏洞的话，攻击者可以设计一段过长的字符串写入缓冲区中就会造成缓冲区溢出。
- 溢出的字符串将会覆盖程序的堆栈数据，同时改变程序执行下一条指令的地址。一般情况下，这个地址会指向一个无效地址或者随机地址，此时将使得程序发生异常、系统不稳定甚至崩溃。更高级的做法是攻击者还可以设置新的地址，将其指向一段植入的可执行代码，这样程序接着就会执行攻击代码从而达到获取程序控制权和其他目的。

3)三线程技术

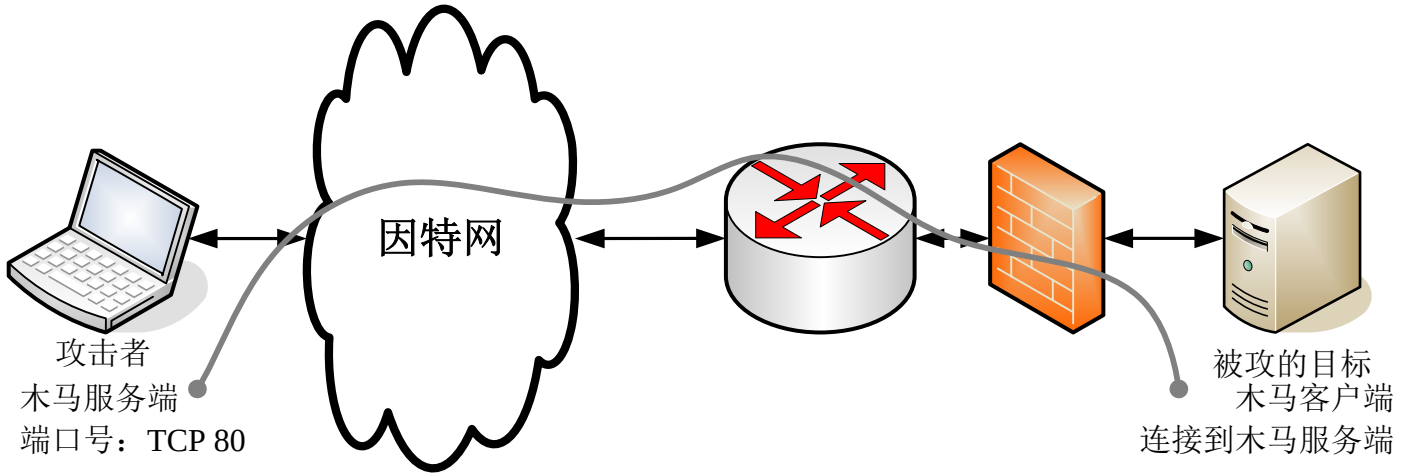
- 在Windows操作系统中，一个进程可以生成多个线程。
- 三线程技术是利用多线程技术来实现的，恶意程序执行时会同时建立三个线程，分别**为主线程、监视线程和守护线程**。其中，主线程用来远程控制目标主机，监视线程主要负责监视进程的状态，检查恶意程序是否被关闭或者删除并向守护线程发送消息。守护线程一般注入目标主机的其他可执行文件内，一旦发现恶意程序进程被关闭，将会强制重新启动该进程。
- 主线程、监视线程和守护线程分工合作，保证了恶意程序的长期运行。

4)端口复用技术

- 端口复用技术是指恶意程序**通过使用网络的常用端口来发送或者接收数据**，这样做的目的是防止新开端口被防火墙等安全机制检测出来。
- 端口复用技术的使用前提是保证该端口其他服务能正常工作，所以这种攻击很难被检测，欺骗性很好。
- 广为所知的**Executor木马**就是采取了端口复用技术来进行远程控制目标主机，它通过80端口来发送控制命令和数据信息。当然，还有很多其他端口可以使用，比如25、135和139。

5)端口反向连接技术

- 端口反向连接是指**被攻击者主动连接攻击者**，通常这里的被攻击者是服务器端，攻击者处于客户端。这种技术利用的是防火墙控制策略，由于防火墙等安全机制一般会对外网流进内网的数据进行严格检查，而对于内部网络流出的数据不作限制。
- Boinet木马是最先采取此技术实现攻击的。



反弹端口型的木马

12.1.5 恶意代码生存技术

1)反跟踪技术

- 恶意代码通过反跟踪技术来增强自身的隐蔽性，使得恶意代码的检测难度变大，减少被系统发现的可能性。

- 根据恶意代码的检测方法可以将反跟踪技术分为两类：
  ① 反**静态**跟踪技术
  ② 反**动态**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。

12.1.5 恶意代码生存技术——2)加密、加壳技术

- 恶意代码大多都会对自身程序代码进行加密，加密技术经常与反跟踪技术配合使用，使得逆向分析者不能得到完整的、正确的恶意代码进行反编译。

- 加壳技术一般对恶意代码文件(exe、dll文件)里面的内存进行加密和压缩，这里需要用到特殊的算法，分析者只有将加过壳的恶意程序脱壳之后才能看到源代码，这样达到反跟踪的目的。

12.1.5 恶意代码生存技术—— 3)变形技术

- 变形技术是指在复制或者传播恶意代码时让其产生变形，防止被基于特征的检测工具识别，使得恶意代码能够长期在目标系统或者网络中存活而不被发现。
- 其**主要是通过对恶意代码片段进行动态地加密和解密**来实现变形的，每一次变形都会将代码进行动态转化，并且和原来的代码不同。
- 这种技术虽然使得代码表面发生了变化，但不会改变代码实际功能。

12.2 常见恶意代码—12.2.1 计算机病毒

1. 病毒的概念与特征

- **概念：计算机病毒是一种人为编制的、能够对计算机正常程序的执行或数据文件造成破坏，并且能够自我复制的一组指令程序代码。**
- 和生物病毒一样，计算机病毒执行使自身能完美复制的程序代码。因此，通过可信任用户在不同计算机间使用磁盘或借助于网络向他人发送文件，病毒是可能从一台计算机传到另一台计算机的。在网络环境下，访问其他计算机的某个应用或系统服务的功能，给病毒的传播提供了一个完美的条件。
- 病毒程序可以执行其他程序所能执行的一切功能，唯一不同的是它必须将自身附着在其他程序（宿主程序）上，当运行该宿主程序时，病毒也跟着悄悄地执行了。

计算机病毒的特性

1)破坏性

- 病毒一旦被触发而发作就会对系统和应用程序产生不同的影响，造成系统或数据的损伤甚至毁灭。所有的病毒都会降低计算机系统的工作效率，占用系统资源，其侵占程度取决于病毒程序自身。
- 病毒的破坏程度主要取决于病毒设计者的目的，如果病毒设计者的目的在于彻底破坏系统及其数据，那么这种病毒对于计算机系统进行攻击造成的后果是难以想象的，它可以毁掉系统的部分或全部数据并使之无法恢复。

计算机病毒的特性

2)传染性

- **计算机病毒的传染性也叫做自我复制或传播性**。病毒通过各种渠道从已被感染的计算机扩散到未被感染的计算机。所谓“感染”，就是病毒将自身嵌入到合法程序的指令序列中，致使执行合法程序的操作会招致病毒程序的共同执行或以病毒程序的执行取而代之。因此，只要一台计算机染上病毒，如不及时处理，那么病毒会在这台机器上迅速扩散，其中的大量文件（一般是可执行文件）就会被感染。而被感染的文件又成了新的传染源，再与其他机器进行数据交换或通过网络接触，病毒会继续传染。**传染性是病毒的最基本特性。**

计算机病毒的特性

3)隐蔽性

- 病毒一般是具有很高编程技巧的、短小精悍的一段代码，通常附着在正常程序代码中。如果不经过代码分析，病毒程序与正常程序是不容易区别开的。这是病毒程序的隐蔽性。
- 很多情况下，病毒程序取得系统控制权后，计算机系统通常仍能正常运行，用户不会感到任何异常，好像计算机内不曾发生过什么。这是病毒传染的隐蔽性。

计算机病毒的特性

4)潜伏性

- 病毒进入系统之后一般不会马上发作，可以在几周或者几个月甚至几年内隐藏在合法程序中，默默地进行传染扩散而不被人发现。
- 病毒的内部有一种触发机制，不满足触发条件时，病毒除了传染外不做什么破坏。一旦触发条件得到满足，病毒便开始执行破坏工作。
- 触发条件可能是预定时间或日期、特定数据出现、特定事件发生等。

计算机病毒的特性

5)多态性

- 病毒试图在每一次感染时改变它的形态，使对它的检测变得更困难。一个多态病毒还是原来的病毒，但不能通过扫描特征字符串来发现。病毒代码的主要部分相同，但表达方式发生了变化，也就是同一程序由不同的字节序列表示。

6)不可预见性

- 计算机病毒制作技术不断提高，种类不断翻新，而相比之下，反病毒技术通常落后于病毒制作技术。新型操作系统、工具软件的应用，为病毒制作者提供了便利。对未来病毒的类型、特点及其破坏性，很难预测。

在其生命周期中，病毒一般会经历如下4个阶段

- **(1)潜伏阶段。**处于休眠状态，病毒最终会被某些条件(如日期、某特定程序或特定文件的出现或内存的容量超过一定范围)所激活。并不是所有的病毒都会经历此阶段。
- **(2)传染阶段。**病毒程序将自身复制到其他程序或磁盘的某个区域上，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。
- **(3)触发阶段。**在被激活后，会执行某一特定功能从而达到某种既定的目的。和处于潜伏期的病毒一样，触发阶段病毒的触发条件是一些系统事件，包括病毒复制自身的次数。
- **(4)发作阶段。**在触发条件成熟时，即可在系统中发作。由病毒发作体现出来的破坏程度是不同的：有些是无害的，如在屏幕上显示一些干扰信息；有些则会给系统带来巨大的危害，如破坏程序以及文件中的数据。

2．病毒的种类

1)按破坏程度的强弱不同，计算机病毒可以分为良性病毒和恶性病毒

- **良性病毒**是指那些只是为了表现自身，并不彻底破坏系统和数据，但会占用大量CPU时间，增加系统开销，降低系统工作效率的一类计算机病毒。该类病毒制作者的目的不是为了破坏系统和数据，而是为了让使用染有病毒的计算机用户通过显示器看到或体会到病毒设计者的编程技术。
- **恶性病毒**是指那些一旦发作，就会破坏系统或数据，造成计算机系统瘫痪的一类计算机病毒。该类病毒危害极大，有些病毒发作后可能给用户造成不可挽回的损失。

病毒的种类

- 2)按传染方式的不同，计算机病毒可分为文件型病毒和引导性病毒
- **文件型病毒**一般只传染磁盘上的可执行文件（如.com、.exe文件）。这类病毒的特点是附着于正常程序文件中，成为程序文件的一个外壳或附件。这是一种较为常见的传染方式。当该病毒完成了它的工作后，其正常程序才被运行，使人看起来仿佛一切都很正常。
- **引导型病毒**是寄生在磁盘引导区或主引导区的计算机病毒。该类病毒感染的主要方式就是发生在计算机通过已被感染的引导盘（常见的如一个软盘）引导时。引导型病毒利用系统引导时不对主引导区内容的正确性进行判别的缺点，在引导系统时侵入系统，驻留内存，监视系统运行。

病毒的种类

- 3)按连接方式的不同，分为源码型病毒、嵌入型病毒、操作系统型病毒和外壳性病毒
- **源码型病毒**较为少见，亦难以编写。它要攻击高级语言编写的源程序，在源程序编译之前插入其中，并随源程序一起编译、连接成可执行的文件，这样刚刚生成的可执行文件便已经带毒了。
- **嵌入型病毒**可用自身替代正常程序中的部分模块，因此，它只攻击某些特定程序，针对性强。一般情况下也难以被发现，清除起来也较困难。
- **操作系统型病毒**可用其自身部分加入或替代或操作系统的部分功能。因其直接感染操作系统，因此病毒的危害性也较大，可能导致整个系统瘫痪。
- **外壳型病毒**将自身附着在正常程序的开头或结尾，相当于给正常程序加了一个外壳。

病毒的种类

- 除了上述几种基本分类方法，还有**隐蔽性病毒、多态性病毒、变形病毒**等概念。
- 隐蔽性病毒目的就是为了躲避反病毒软件的检测；多态性病毒每次感染时，放入宿主程序的代码互不相同，不断变化，因此采用特征代码法的检测工具是不能识别它们的；变形病毒像多态性病毒一样，它在每次感染时都会发生变异，但不同之处在于，它在每次感染的时候会将自己的代码完全重写一遍，增加了检测的困难，并且其行为也可能发生变化。

3．病毒预防技术

- 病毒预防是指**在病毒尚未入侵或刚刚入侵还未发作时，就进行拦截阻击或立即报警**。
- 要做到这一点，首先要清楚病毒的传播途径和寄生场所，然后对可能的传播途径严加防守，对可能的寄生场所实时监控，达到封锁病毒入口，杜绝病毒载体的目的。
- 不管是传播途径的防守还是寄生场所的监控，都需要一定的检测技术手段来识别病毒。

病毒预防技术

1)病毒的传播途径及其预防措施

- 常见措施包括：对不可移动的计算机硬件设备（包括ROM芯片、专用ASIC芯片和硬盘），以及可移动的存储介质设备（包括软盘、磁带、光盘以及可移动式硬盘等），经常用检测病毒软件或其他病毒检测手段（包括人工检测方法）检查；建立封闭的使用环境，即做到专机、专人、专盘和专用，如果通过移动存储设备与外界交互，要进行病毒检测；采取各种措施保证网络服务器上的系统、应用程序和用户数据没有染毒，如坚持用硬盘引导启动系统，经常对服务器进行病毒检查等；对网络内的共享区域，如电子邮件系统、共享存储区和用户数据卷进行病毒扫描监控，发现异常及时处理，防止在网上扩散；一旦某台计算机出现染毒迹象，应立即隔离并进行排毒处理，防止它通过网络传染给其他计算机；等等。

2)病毒的寄生场所及其预防措施

- 常见措施包括：对计算机系统干净的引导扇区进行备份；检查.com和.exe可执行文件的内容、长度、属性等，判断是否感染了病毒；采用一些内存检测工具，检查内存的大小和内存中的数据来判断是否有病毒进入；等等。

4. 病毒检测技术

- 理想的解决病毒攻击的方法是对病毒进行预防，即在第一时间阻止病毒进入系统。尽管预防可以降低病毒攻击成功的概率，但一般说来，上面的目标是不可能实现的。因此，实际应用中主要采取检测、鉴别和清除的方法。

① **检测：**一旦系统被感染，就立即断定病毒的存在并对其进行定位。

② **鉴别：**对病毒进行检测后，辨别该病毒的类型。

③ **清除：**在确定病毒的类型后，从感染文件中删除所有的病毒并恢复程序的正常状态。

- 病毒检测就是采用各种检测方法将病毒识别出来。识别病毒包括对已知病毒的识别和对未知病毒的识别。目前，对已知病毒的识别主要采用特征判定技术，即静态判定技术；对未知病毒的识别除了特征判定技术外，还有行为判定技术，即动态判定技术。

1)特征判定技术

- 特征判定技术是根据病毒程序的特征，如感染标记、特征程序段内容、文件长度变化、文件校验和变化等，对病毒进行分类处理，而后在程序运行中凡有类似的特征点出现，则认定是病毒。

2)行为判定技术

- 识别病毒要以病毒的机理为基础，不仅识别现有病毒，而且基于现有病毒的机理设计出对一类病毒（包括基于已知病毒机理的未来新病毒或变种病毒）的识别方法，其关键是对病毒行为的判断。行为判定技术就是要解决如何有效辨别病毒行为与正常程序行为，其难点在于如何快速、准确、有效地判断病毒行为。如果处理不当，就会带来虚假报警。
- 行为监测法是常用的行为判定技术，其工作原理是利用病毒的特有行为特征进行检测，一旦发现病毒行为则立即警报。经过对病毒多年的观察和研究，人们发现病毒的一些行为是病毒的共同行为，而且比较特殊。在正常程序中，这些行为比较罕见。

12.2.2 木马

1.木马概述

- **木马是有隐藏性的、可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恶意程序**。因此，木马是可被用来进行恶意行为的程序，但这些恶意行为一般不是直接对计算机系统的软硬件产生危害的行为，而是以控制为主的行为。
- 木马通常不像传统病毒那样感染文件。木马一般是以寻找后门、窃取密码和重要文件为主，还可以对计算机进行跟踪监视、控制、查看、修改资料等操作，具有很强的隐蔽性、突发性和攻击性。而且，木马也不像病毒那样重视复制自身。

2．木马工作原理

- 木马程序一般采用客户机／服务器工作模式，包括客户端(client)部分和服务器端(server)部分。客户端也叫控制端，运行在木马控制者的计算机中；服务器端运行在被入侵计算机中，打开一个端口以监听并响应客户端的请求。
- 典型地，攻击者利用一种称为绑定程序的工具将木马服务器端绑定到某个合法软件或者邮件上，诱使用户运行合法软件。只要用户一运行该软件，特洛伊木马的服务器端部分就在用户毫无知觉的情况下完成了安装过程。

木马工作原理

- 通常，特洛伊木马的服务器端部分都是可以定制的，攻击者可以定制的项目一般包括：服务器端运行的IP端口号、程序启动时机、如何发出调用、如何隐身、是否加密等。另外，攻击者还可以设置登录服务器端的密码，确定通信方式。木马控制者通过客户端与被入侵计算机的服务器端建立远程连接。一旦连接建立，木马控制者就可以通过对被入侵计算机发送指令来控制它。
- 不管特洛伊木马的服务器端和控制端如何建立联系，有一点是不变的，就是攻击者总是利用控制端向服务器端发送命令，达到操控用户机器的目的。用木马这种工具控制其他计算机系统，从过程上看大致可分为六步。

木马的工作过程

1)配置木马

- 一般来说，一个设计成熟的木马都有木马配置程序，从具体的配置内容看，主要是为了实现以下两方面功能：①木马伪装：为了让服务端在侵入的主机上尽可能好地隐藏，木马配置程序会采用多种手段对服务器端进行伪装，如修改图标、捆绑文件、定制端口、自我销毁等。②信息反馈：木马配置程序就信息反馈的方式或地址进行设置，如设置信息反馈的邮件地址、IRC号、QQ号等。

木马的工作过程

2)传播木马

- 当前，木马的传播途径主要有两种：一种是通过电子邮件，木马服务器端以附件形式附着在邮件上发送出去，收件人只要打开附件就会感染木马。为了安全起见，现在很多公司或用户通过电子邮件给用户发送安全公告时，都不携带附件。第二种是软件下载，一些非正式的网站以提供软件下载的名义，将木马捆绑在软件安装程序上，程序下载后只要一运行这些程序，木马就会自动安装。因此，用户从互联网上下载了免费软件以后，在运行之前一定要进行安全检查。对于安全要求较高的计算机，则应禁止安装从互联网上下载的软件。

木马的工作过程

3)运行木马

- 服务端用户运行木马或捆绑木马的程序后，木马就会自动进行安装，并设置好木马的触发条件，条件满足时将自动运行木马的服务器端。木马被激活后，进入内存，并开启事先定义的木马端口，准备与控制端建立连接。

木马的工作过程

4)信息收集与反馈

- 一般来说，设计成熟的木马都有一个信息反馈机制。所谓信息反馈机制是指木马成功安装后会收集一些服务端所在计算机系统的软硬件信息，并通过E-mail、IRC或QQ的方式告知控制端。
- 从反馈信息中控制端可以知道服务端的一些软硬件信息，包括使用的操作系统、系统目录、硬盘分区情况、系统口令等。在这些信息中，最重要的是服务端IP，因为只有得到这个参数，控制端才能与服务端建立连接。

木马的工作过程

5)建立连接

- 一个木马连接的建立首先必须满足两个条件：一是服务端已运行在被入侵的计算机中；二是控制端要在线。在此基础上控制端可以通过木马端口与服务端建立连接。

6)远程控制

- 木马连接建立后，控制端端口和木马端口之间将会出现一条通道。控制端程序可借助这条通道与服务器端取得联系，并通过服务器端对被入侵主机进行远程控制。比如通过击键记录来窃取密码、对服务端上的文件进行操作、修改服务器端配置、断开服务端网络连接、控制服务端的鼠标与键盘、监视服务端桌面操作、查看服务端进程等。

3．木马常用技术

- 现代木马采用了很多先进的技术，以提高自身隐藏能力和生存能力。这些技术包括进程注入技术、三线程技术、端口复用技术、超级管理技术、端口反向连接技术等。

1)进程注入技术

- 当前操作系统中都有系统服务和网络服务，它们都在系统启动时自动加载。进程注入技术就是将这些与服务相关的可执行代码作为载体，木马将自身嵌入到这些可执行代码中，实现自动隐藏和启动的目的。
- 这种形式的木马只需安装一次，以后就会被自动加载到可执行文件的进程中，并且会被多个服务加载。只有系统关闭，服务才会结束，因此木马在系统运行时始终保持激活状态。

2)三线程技术

- 三线程技术就是一个木马进程同时开启了三个线程，其中一个为主线程，负责接收控制端的命令，完成远程控制功能。另外两个是监视线程和守护线程，监视线程负责检查木马是否被删除或被停止自动运行。守护线程则注入其他可执行文件内，与木马进程同步，一旦进程被终止，它就会重新启动木马进程，并向主线程提供必要的数据，这样就可以保持木马运行的可持续性。

3)端口复用技术

- 端口复用技术是指重复利用系统网络打开的端口，如25、80、135等常用端口，来进行数据传送，这样可以达到欺骗防火墙的目的。
- 端口复用是在保证端口默认服务正常工作的条件下复用，具有很强的隐蔽性和欺骗性。比如，木马“Executor”利用80端口来传送控制信息和数据，实现远程控制的目的。

木马常用技术

4)超级管理技术

- 一些木马还具有攻击反恶意代码软件的能力。为了对抗反恶意代码软件，一些木马采用超级管理技术对反恶意代码软件进行拒绝服务攻击，使反恶意代码软件无法正常工作。比如，国产木马“广外女生”就采用超级管理技术对金山毒霸和天网防火墙进行拒绝服务攻击，以使其无法正常工作。

5)端口反向连接技术

- 一般来说，防火墙对外部网络进入内部网络的数据流有严格的过滤策略，但是对内部网到外部网的数据流控制力度相对较小一些。端口反向连接技术指的是利用了防火墙的这个特点。控制端口反向连接，从而使得数据流的流向从被侵入方来看是从内到外。国外的“Boint”是最早实现端口反向连接的木马，国内的“灰鸽子”木马则是这项技术的集大成者。

4．木马的预防

- 目前木马已对电脑用户信息安全构成了极大的隐患，做好木马的防范已经刻不容缓。用户要提高对木马的警惕，尤其是网络游戏玩家、电子商务参与者更应该提高对木马的关注。
- 网络中比较流行的木马程序，传播速度比较快，影响也比较严重，因此尽管我们掌握了很多木马的检测和清除方法及软件工具，但这些也只是在木马出现后被动的应对措施。这就要求我们平时要有对木马的预防意识和措施，做到防患于未然。以下是几种简单适用的木马预防方法和措施。

木马的预防

(1)不随意打开来历不明的邮件，拒绝可疑邮件。
(2)不随意下载来历不明的软件。
(3)及时修补漏洞和关闭可疑的端口。
(4)尽量少用共享文件夹。
(5)运行实时监控程序。
(6)经常升级系统和更新病毒库。
(7)限制使用不必要的具有传输能力的文件。

12.2.3 蠕虫

1. 蠕虫概念与特征

- **蠕虫是一种结合黑客技术和计算机病毒技术，利用系统漏洞和应用软件的漏洞，通过复制自身进行传播的、完全独立的程序代码**。蠕虫的恶意行为主要体现在消耗系统资源和网络资源上。
- **蠕虫是一种通过网络传播的恶意代码**，它具有普通病毒的传播性、隐蔽性和破坏性，但与普通病毒也有很大差别。
- 就存在形式而言，蠕虫不需要寄生到宿主文件中，它是一个独立的程序，而普通病毒需要宿主文件的介入。传统病毒是需要寄生的，它可以通过自己指令的执行，将自己的指令代码写到其他程序内，而被感染的文件就被称为“宿主”。

1．蠕虫概念与特征

- 就**触发机制**而言，蠕虫代码不需要计算机用户的干预就能自动执行。一旦蠕虫程序成功入侵一台主机，它就会按预先设定好的程序自动执行。而传统病毒代码的运行，一般需要用户的激活。只有用户进行了某个操作，才会触发病毒的执行。
- 就**复制方式**而言，蠕虫完全依靠自身来传播，它通过自身的复制将蠕虫代码传播给扫描到的目标对象。而普通病毒需要将自身嵌入到宿主程序中，等待用户的激活。
- 就**搜索机制**而言，蠕虫搜索的是网络中存在某种漏洞的主机。普通病毒则只会针对本地上的文件进行搜索并传染，其破坏力相当有限。也正是由于蠕虫的这种搜索机制导致了蠕虫的破坏范围远远大于普通病毒。

1．蠕虫概念与特征

- 就**破坏对象**而言，蠕虫的破坏对象主要是整个网络。蠕虫造成的最显著破坏就是造成网络的拥塞。而普通病毒的攻击对象则是主机的文件系统，删除或修改攻击对象的文件信息，其破坏力是局部的、个体的。
- 最后，蠕虫的可怕之处也就在于它不需要计算机用户的参与就能悄无声息地传播，直至造成了严重的影响甚至是网络拥塞才会被人们所意识到。而此时，蠕虫的传播范围已非常广泛。
- 任何蠕虫在传播过程中都要经历如下三个过程：首先，探测存在漏洞的主机；其次，攻击探测到的脆弱主机；最后，获取蠕虫副本，并在本机上激活它。因此，蠕虫代码的功能模块至少需包含扫描模块、攻击模块和复制模块三个部分。

2. 蠕虫的分析和防范

- 对于个人用户而言，威胁大的蠕虫一般采取电子邮件和恶意网页传播方式。这些蠕虫对个人用户的威胁最大，同时也最难以根除，造成的损失也很大。
- 防范蠕虫需要从以下几点入手：

(1)提高防杀恶意代码的意识。

(2)保持各种操作系统和应用软件的更新，解决漏洞问题。

(3)购买正版的防病毒（蠕虫）软件。

(4)经常升级病毒库。

(5)不随意查看陌生邮件，尤其是带有附件的邮件。

(6)建立备份和容灾系统。对于数据库和数据系统，必须采用定期备份、多机备份和容灾等措施，防止意外灾难下的数据丢失。

12.3 恶意代码检测与分析技术

- 一个完整的恶意代码检测与分析系统通常由两个重要部分组成：**技术模块和分析模块**。
- 首先，**技术模块的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。
- 而**分析模块是用于分析**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

- 技术模块需要搜集、提取的信息主要有三种：
- **(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。

分析模块采用的技术

- 不同的数据类型所采取的提取方法也不同，所以根据技术模块使用的不同技术可以将恶意代码检测与分析技术分为两种：一种是静态分析方法，另一种是动态分析方法。
- **静态分析方法**主要包括：基于特征码的检测、基于代码语义检测和启发式扫描方法。
- **动态分析方法**主要包括：系统监控法和动态跟踪法。

12.3.1 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

- 基于特征码的检测方法是现在广泛使用的恶意代码检测方法之一，目前大多数商业杀毒软件都是采取这种检测机制。
- 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。
- 检测过程如下：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。现阶段，特征码的生成技术主要是利用噪声引导和自动产生分发。

- 由于早期的恶意代码种类少，形态单一，这种检测方法取得了比较好的效果，只要特征库中存在该恶意代码的特征码，那么就能检测出来。但是随着技术的发展，变形技术的使用出现了许多变种恶意代码，这使得很多已知的恶意代码也无法检测出来。
- 总的来说，由于只有已知的恶意代码特征码才会加入到特征库中，所以基于特征码的检测方法主要优点有：误报率低，容易实现。然而该方法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。**自动检测是其发展方向。**

- M.Christodorescu及其同事就使用了基于代码语义的检测方法来分析恶意程序代码，并在此方法的基础上做出了改进和完善。
- 对于一般的恶意代码程序，他采用了切片技术对恶意程序的汇编代码进行分析，成功为恶意程序建立模型并得到控制流程图。在此基础上，分析得到程序的功能和意图从而来判断该程序是否为恶意程序。对于某类恶意代码或变种代码，他采取的方法是将恶意代码的反汇编代码中的语义信息抽象为一个模板，用来建立中间语言层的模型，然后将其与其他恶意代码的模板进行模式比较。
- 这里为了能在中间语言层进一步将程序的行为模式描述出来，还可以使用基于语义探测器的方法来进行分析。

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。

3)启发式扫描方法

- 目前大多数商业杀毒软件都是使用基于特征码检测的方法，不过现在也有不少反恶意软件开始加入启发式扫描功能。

- 启发式扫描技术虽然增强了对变种、变形和未知恶意代码的检测效果，但是该方法仍然离不开特征的提取，所以恶意代码编写者只要能改变恶意代码的特征就能轻易地避开启发式扫描技术的检测。

12.3.2 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。常见的动态分析方法主要可以分为两类：系统监控法和动态跟踪法。

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。

1)系统监控法

- 根据监控的对象不同，又可以将系统监控方法细致划分为以下几点。
- **(1)文件监控**。大多数恶意代码在执行过程中不仅对内存进行读写操作，而且还会与文件系统发生交互，所以需要监控文件系统，可以根据文件的状态变化来分析恶意代码。恶意代码对目标主机的文件操作经常有添加、修改或删除各种类型文件，修改其他程序包括应用程序和系统程序，还有可能将恶意代码嵌入到其他文件中。FileMon是一种常用的文件监控工具，它能记录与文件相关的许多操作行为（如打开、读写、删除和保存等）。还有能检测文件完整性的程序，如Tripwier，AIDE(advanced intrusion detection environment)等。

1)系统监控法

- **(2)进程监控**。由于恶意代码在传播或者攻击的过程中，一定会产生新的进程或者嵌入系统进程中，所以需要监控主机进程，可以根据被攻击的主机上所有进程的变化信息来分析恶意代码。Windows系统通常自带了一个进程展示工具，它能列出当前系统内所有正在运行的进程名字、CPU占用率和占用内存大小等信息。由于其功能有限，而分析恶意进程需要了解进程更多的活动情况，所以一般使用Process Explorer，还能看到进程的优先级和环境变量。Process Explorer是一个专业的进程监控程序，它还能监控进程装载过程和注册表键值的变化情况。这样分析者可以对进程的整个结构具有一定的了解，从而能更好地分析恶意代码。

- **(3)网络监控**。恶意代码的传播手段多种多样，不仅可以通过存储介质（U盘、光盘等），现在更多的是通过网络传播，如电子邮件、网页木马等。所以需要监控恶意代码的网络行为来分析恶意代码。常见的网络监控工具有TCPview、Fport、Nmap和Nessus。其中，TCPview和Fport可以查看本机开放端口和进程的关系，用来检测是否有新的TCP和UDP连接系统。TCPview和Fport最大的不同是，TCPview能用于Windows和Linux平台上，而Fport适用于Linux。Nmap和Nessus的功能是用来查看系统是否存在脆弱性信息，这两种扫描工具都可以用在Windows和Linux系统上。
- **(4)注册表监控**。恶意代码通常需要改变系统注册表项来控制操作系统的操作，或者是为了实现其他目的，所以注册表监控也是恶意代码分析过程中的重要一项内容。常用的注册表监控工具有Regmon，它能够为恶意代码分析者实时提供所有读写注册表项的操作。

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**。这里使用了用户态的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，结合输入输出参数和指令特征等信息来分析得到恶意代码的行为。这种技术的缺点是：对于直接调用系统内核服务的行为，使用API Hooking技术无法获取用来分析。基于用户态行为检测技术的分析工具有很多，常见的有Joebox，它能在真实的计算机网络环境中者实时地分析恶意代码，不需要预先配置虚拟机或者沙箱，能够有效地检测出具有抗虚拟机检测功能的恶意代码。还有 CWSandBox和Norman Sandbox两种工具只能运行于虚拟操作系统环境检测。

- **(2)基于内核态的行为监测技术**。这种检测技术是指使用内核态的API Hooking技术来跟踪恶意程序的行为。这里还使用到了设备驱动过滤技术，就是在Windows设备驱动上再加一层过滤驱动，监视并截取I/O请求用来分析程序对文件系统的操作行为。与基于用户态的行为监测技术相比，这种方法比较难以实现，但是由于其工作于系统内核态，具有很强的监测能力。现有的基于此技术实现的工具有Capture、MwDAS等。
- **(3)基于指令模拟器的行为监测技术**。基于指令模拟器的行为监测是指，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。这种方法能很有效地分析出恶意代码的行为，但是其实现的技术难度比较大。现有的基于此方法实现的工具有TTAnalyze、Panorama等。

12.3.3 恶意代码分类方法

- 恶意代码的检测离不开分类技术的支持，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。
- 快速划分恶意代码的种类能有效促进恶意代码的检测和清除，并促进恶意代码解决方案的提出。随着变种恶意代码数量的剧增，恶意代码分类技术已经成为安全学者们重点研究领域之一。
- 根据是否需要计算代码之间的相似性，可以将常见的恶意代码分类方法分为如下两种。

1) 基于相似性计算的分类方法

- 通常情况下，同一家族的恶意程序性相似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。
- Marius Gheorghescu采用了计算程序控制流程图相似性的方法来对恶意代码进行分类。首先中流的离代某代码家还使用了倒实度过脱代能意将图之需索结果的静壳码够代给一内要引果表明，使用编辑距离匹配算法效果最为理想。

1) 基于相似性计算的分类方法

- 使用相似性计算的分类技术通常需要为恶意代码家族行为事件定义框架，该框架必须能够精确地反映恶意代码的行为信息，包含丰富的语义，具有很强的结构性。代码的相似度可以由不同的距离度量算法得到，但是只能使用聚类分析来实现恶意代码的分类。然而聚类缺少来自外部信息的监督学习，不具备自适应特征，没有准确的衡量标准，所以其分析结果经常不会很理想。比如，Michael Bailey在实验中使用聚类分析将3698个恶意代码样本分成了403个簇，但实际上其中有206个簇是属于同一个家族。

2) 基于数据挖掘的分类方法

- 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。
- 使用基于数据挖掘的分类方法来对恶意代码进行分类，分类流程一般如图12-1所示。该流程通常包括4个步骤。

2) 基于数据挖掘的分类方法

- (1)收集实验所需的恶意代码样本。
- (2)提取恶意代码家族的特征。
- (3)训练学习生成分类器。
- (4)选取测试集进行分类测试。
- 基于数据挖掘的分类方法具有自学习的能力，所以准确率非常高。Batten提取恶意代码中包含的一些特殊字符集作为特征集，在WEKA(waikato environment for knowledge analysis)上对来自11个恶意代码家族的样本进行分类测试，采取了5次交叉验证的形式，测试结果正确率高达97%。

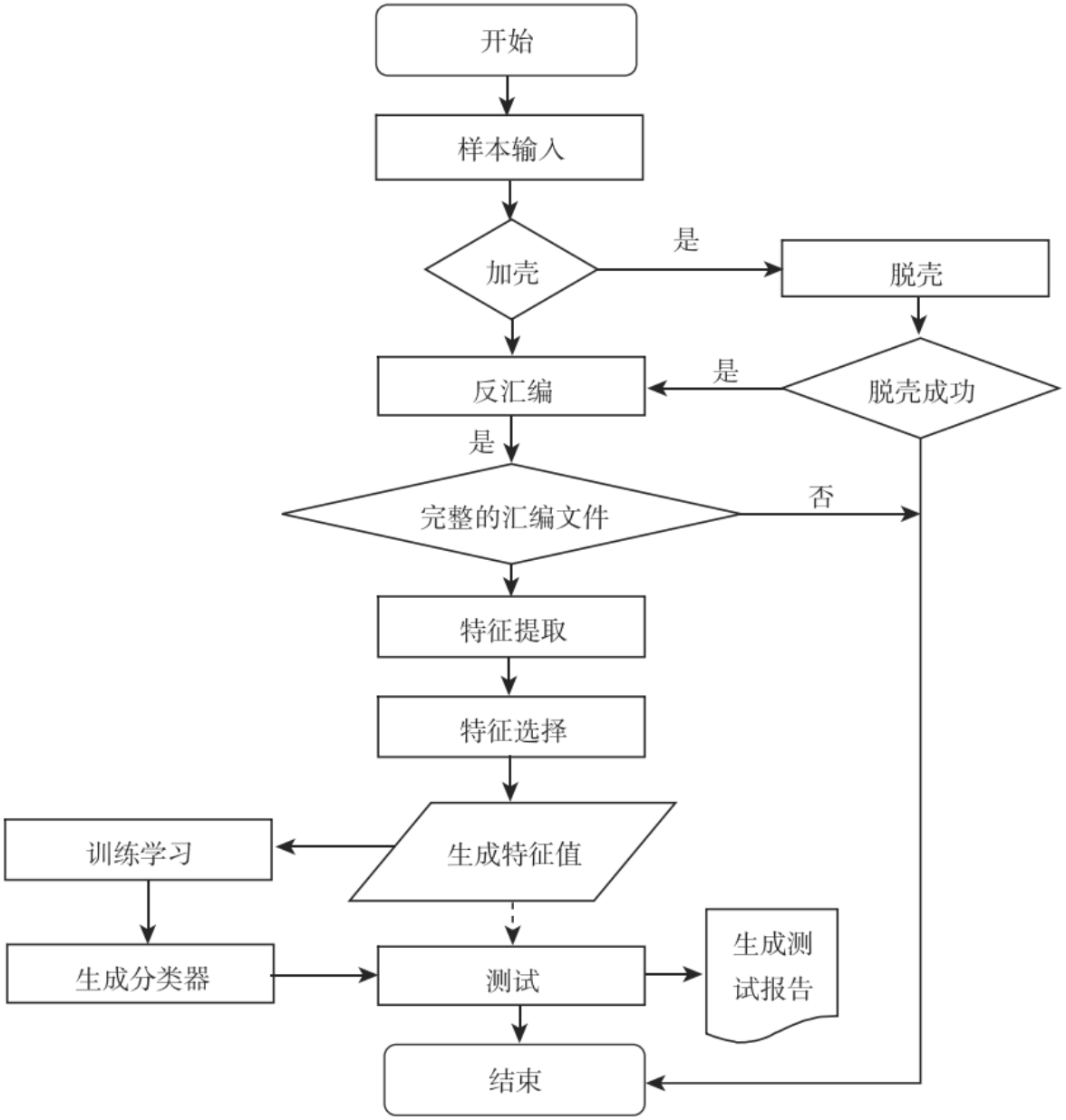


图12-1 基于数据挖掘的恶意代码分类流程图

No! Don’t do that!

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

第12章 作业

- 作业
  1. 恶意代码包括哪些种类？主要特点是什么？
  3. 计算机病毒有哪些特征？
  4. 病毒检测有哪些典型技术？各自的工作原理是什么？
  5. 木马的工作原理是什么？
  6. 蠕虫有哪些独特的特点？

- 实践（自己研究，不考核）
  - 调研“熊猫烧香病毒”