计算机实用软件工具系列丛书

胡祥义
田立宪
汪永洲
天　奥　编著

计算机病毒与数据防护

学苑出版社

计算机病毒与数据防护

胡祥义　田立宪
汪永洲　天　奥　编写

熊可宜　审校

学苑出版社
1994

(京)新登字 151 号

内 容 简 介

本书共分两大部分，两部分各自独立成章，又互为补充。

第一部分是“计算机病毒与数据防护”。从计算机病毒的基本概念入手，由浅入深，阐述了计算机病毒的危害性、特征及感染计算机的途径，列举大量病毒实例，给出了病毒源程序，并逐段进行分析，使读者了解和掌握各类常见病毒的特征、攻击文件的方式以及预防措施，从而更好地保护数据，防止病毒的侵害。本部分由胡祥义、田立宪、汪永洲编写。本部分的编写还得到了苏文忠、曹新春等的大力支持。

第二部分是“反病毒软件 CPAV”。本部分是检查和清除已知病毒的实用程序的综合集。它由八章组成，共同构成一个计算机的完整的保护防线。本部分介绍 CPAV 软件及其组成，详细描述了安装过程，并帮助用户学习如何使用 CPAV。本部分由天奥编写。

本书可作为计算机用户了解和预防计算机病毒的一本教材，也可作为计算机病毒预防和检测人员的参考。

需要本书的用户，请直接与北京 8721 信箱联系，邮编：100080，电话：2562329。

计算机实用软件技术系列丛书

计算机病毒与数据防护

编　　写：胡祥义　田立宪　汪永洲　天　奥
审　　校：熊可宜
责任编辑：甄国宪
出版发行：学苑出版社　　邮政编码：100036
社　　址：北京市海淀区万寿路西街 11 号
印　　刷：双青印刷厂
开　　本：787×1092　1/16
印　　张：17.375　　字数：388 千字
印　　数：1～5000 册
版　　次：1994 年 6 月北京第 1 版第 1 次
ISBN 7—5077—0876—4/TP・25
本册定价：21.00 元

学苑版图书印、装错误可随时退换

目　录

第一部分　计算机病毒与数据防护

第二部分 反病毒软件 CPAV

第一部分

计算机病毒与数据防护

第一章 什么是计算机病毒

80年代初，如果有人说病毒能感染计算机，别人听了会觉得好笑。

在此期间，有大量的证据说明计算机病毒是存在的，但是，很多用户传说，计算机病毒起源于生物病毒，据有关方面报道，磁盘细菌对计算机进行腐蚀，破坏ROM。这就误解了计算机病毒的实质，在计算机用户的思想上造成了混乱和惊慌。

上述例子当然不是真的，计算机病毒就是一个计算机程序。至今，人们才重视对计算机病毒进行正确的定义。本书正是为了纠正人们的混乱概念，并作为一本计算机病毒手册奉献给用户。

1.1 计算机病毒与生物病毒类似

虽然人们把计算机病毒和生物病毒混为一谈是错误的，但是，两者恰有许多相同的特征。例如，一台带有病毒的计算机能通过用户向另一台计算机扩散，只要用户不使用被感染的程序和磁盘，就可避免计算机被感染。

生物病毒和计算机病毒都是传染源，每一种病毒都使自己的基质不断增长。具有破坏性的病毒能够清掉文件，重新格式化硬盘，给用户造成错觉，以为键盘出故障了。

下面列表说明有关生物病毒和计算机病毒的相似处。

功　能	生物病毒	计算机病毒
攻击/感染	特殊的体内细胞	特殊的程序(全部*.COM，*.EXE，等等)
破坏目的	改变细胞的原始成份	改变程序中的数据
重复复制	新的病毒在被感染的细胞里繁殖	被感染的程序产生另一个被感染的程序
重复感染	被感染的细胞不再受感染	被感染的程序不再受感染
潜伏期	症状在该生物体内可能长时间潜伏	被感染的程序能在一段时间内正常工作，不出错
其它相似之处	生物体可能发展成对病毒产生免疫性，生物体病毒能变异，所以，显然不能被发现	程序能产生病毒免疫功能，病毒程序通过自我修改掩藏起来

你大概会感到疑惑，计算机中的程序怎么会像生物病毒一样活动？要回答这个问题，不但要精通计算机系统，而且要精通计算机病毒的特征，我们在本章以下几节中将讨论这个问题。

1.2　病毒程序的特征

一个病毒程序必须同时具备几个特征才被称为病毒(详见 1.3 节)。

作为控制其它程序的病毒程序,是因为它在运行过程中修改别的程序,并自我复制增长(长度增加),在本章我们将讨论计算机病毒是如何完成这些操作的。

1.2.1　哪种病毒能感染计算机系统

对于计算机病毒来说,在没有人(即:用户)"帮助"下传播是不可能的。除非用户不慎或偶然地执行一个被感染的程序,否则,病毒是不能感染计算机系统的。

下面是计算机病毒通常攻击的目标。

1. 硬驱动器的文件分配目录表。

2. 有些病毒把自己与其它文件连接,或与 COM 文件连接(例:COMMAND. COM)、与 EXE 文件连接,或者,与其它隐含的系统文件相连。

3. 另有一些病毒会感染磁盘扇区。

当你考虑到计算机病毒时,就没有"安全文件",任何在系统中的可执行文件都可能感染上病毒。

1.2.2　病毒必须进行复制才起作用

计算机病毒通过修改文件、驱动器设备或系统中的其它数据等方式进行复制。当你执行一个被感染程序时,病毒程序段一般在改变控制主程序之前首先运行。

病毒能够完成很多"任务"。

1. 它能立刻感染其它程序。

2. 它能在一段时间以后感染其它程序(大概通过系统时钟激活病毒进行感染)。

3. 它可能进入 RAM 里,也可能继续扩散。

当一个感染上病毒的文件在其他系统上使用时,病毒就扩散到其他系统中去。

1.2.3　病毒是怎样感染程序的

病毒必须通过两方面发生作用:

. 操作

. 复制

操作方面必须通过某些变化激发出来,这些变化过程较常见的一种是特殊的数据或根据系统时钟的时间;另一种常见的变化是使病毒在一个定点时间开始执行。

病毒程序通常不是一开始执行就破坏你的系统,病毒程序首先必须增长、复制并扩散。

当被病毒感染的程序开始执行时,它先寻找当前的驱动器中的可变换程序,如果病毒程序找到这样的程序,它对其进行检查,看看是否已经被感染了。我们很快说明介绍以上这些内容为什么是很重要的。

病毒程序的上述做法是通过读程序的首部,核实是否出现病毒标志的"S"字样。

病毒注册了内存字节,说明感染了程序。因病毒不能再感染一个已经受感染的程序,则病

毒就继续寻找下去，直至找到一个没被感染的程序为止，这是一个不包括病毒标志“S”的程序。

实行保护，避免各种病毒感染是很必要的，可是，病毒仍要花费大部分时间感染一个已经被感染的程序。

下面举例并画表进行说明：

S 表示病毒注册字样，这表明已经实现感染，并避免程序多次感染。

VIR 代表病毒核，它包括一些程序库和函数，这些程序和函数对病毒进行复制。

我们假定找到的第一个被感染的用户程序，并包括病毒的标志。

S	VIR	第一个用户程序

病毒跳过被感染的程序，寻找第二个未被感染的用户程序，若这第二个程序是能够被感染的用户程序，则病毒就通过拷贝操作，把自已写到该程序的首部。从而进入用户程序。

第二个用户程序

这时，病毒正在扩散，用户可以注意到病毒进行扩散时向磁盘驱动器写(病毒)的过程。

S	VIR	第二个用户程序

当病毒感染完毕第二个用户程序时，病毒程序也执行完毕，这是因为病毒已写进第二个用户程序中去了。于是，病毒重复这个过程，把自已复制到第三个用户程序中。

病毒拷贝过程完成之后，一些严重的程序错误在第二个用户程序中出现。第二个用户程序的一部分丢失，因为，其所需空间被病毒码占领。

病毒再继续搜索第三个未被感染的程序。

第三个用户程序

如果这第三个程序是能被进行感染的用户程序，则病毒再次通过拷贝把自已写在程序首部。

S	VIR	第三个用户程序

当感染完毕第三个用户程序后，病毒程序则再次运行，于是，病毒重复以上过程，把自已复制到第四个用户程序中去。

以上过程不断进行，直到没有可感染的程序为止，直至病毒被发现或者离开为止。

1.3 病毒程序的定义

在此，我们给出计算机病毒的定义方法和种类。提供各种病毒程序的操作过程。

我们认为，正确定义计算机病毒是困难的。

许多专家对如何确切定义计算机病毒的意见不一致，部分专家主张 Christmas 病毒(参见 5.1 节和 9.3 节)不是病毒，因为 Christmas 不感染其它程序。

另一部分专家确信特洛伊木马(Trojan horse)(参见 1.4 节)是病毒，因为它需要用户“帮

助"扩散。

一般来说,计算机病毒的定义标准是,看设计的程序是否用于起破坏作用,造成恶作剧;是否能完成插入,并把程序拷到其它程序中(也包括系统程序);甚至,要看是否被感染的程序能依次把病毒的复制品放到其它程序中去。

人们常常忽视计算机病毒,正像我们在第 1.1 节讨论的那样,计算机病毒与生物病毒有相似之处。举例来说:用户也许在几天或几周里都没有察觉到计算机病毒的影响,在此期间,任何插到被感染的计算机驱动器上的软盘,都可能被隐藏的病毒感染,被感染的软盘能感染其它计算机系统。

科学的定义计算机病毒是件困难的事,已经发表的确定病毒的特征是正确的,举例来说,病毒不必进行整个程序的自我拷贝,仅需要复制程序的某一部分。有些病毒仅限定有效程序段的复制,且不与其它程序联接在一起。

专家们对于计算机病毒的定义方法意见不同。如果你的 PC 机感染上特洛伊木马(Trojan Horse)病毒,你要做到的事就是去清掉它。我们在 6.1 节讨论"疫苗"程序,包括:检测、识别以及在某些情况下破坏计算机病毒。

1.3.1 病毒程序的属性

在本章中,我们通过对确定各种计算机病毒方法的讨论,很容易使我们弄混,可是鉴定一个程序是否为计算机病毒最好的方法就是看如下特征。

程序具备如下特征,我们称其为计算机病毒。

1. 该程序是否通过与其他程序接在一起,来修改这些程序。
2. 该程序是否完成对很多程序的修改。
3. 该程序是否能辨认已修改过的程序。
4. 该程序是否有能力防止同一个程序再被修改。
5. 已修改过的程序是否也具有以上 4 点特征。

如果一个程序不同时具备以上五点特征,则严格地说该程序就不能称为病毒程序。

1.4 类似病毒的程序

正像我们在前节中讨论的那样,病毒程序没有十分明确的定义,有些程序常常被人们混成病毒,据专家们说,某些具有病毒的个别特征的程序,也不一定是病毒。在本节,我们讨论三个有代表的例子。

1.4.1 寄生虫程序

寄生虫程序通过建立自我拷贝,来自我复制。尽管该程序具有病毒的一个特征,但是,它不需要基质程序来复制。

1.4.2 逻辑病毒程序

某些专家认为逻辑病毒是病毒程序,这些程序通过将文件更名的方法,不仅能修改其基质程序,而且能删除和恢复基质程序。

举例来说，如果A是一个病毒程序，B是一个用户程序，于是，B命名为A，使B作为病毒出现。

1.4.3 特洛伊马程序

这些程序包括一些具有破坏性的指令码的函数，在执行中，使用户在上机处理工作时，被搞的糊里糊涂，当你感到情况不妙时，特洛伊木马程序可能会对其硬盘重新格式化了。

特洛伊木马程序有较大的破坏性，只有你把带有该程序的软盘插到其它计算机上工作时，该硬盘的全部数据就会被删除。

1.5 确定病毒有利于用户

许多用户经常讨论计算机上的病毒影响问题。他们认为，最典型的例子是，Fred Cohen 压缩病毒(详见第 2.1 节)。

在系统中的这种病毒，首先对所有可执行程序进行感染。病毒为减少对驱动器上已感染程序的内存开支，通过 huffman 编码压缩数据。从而达到压缩文件长度之目的。

程序文件的存储空间依其结构减少 50%～80%，我们发现 Huffman 编码能够大幅度压缩文本文件和图表文件的内存空间占有量。但是，你不能直接运行经过压缩的程序，在拟运行经过压缩的程序之前，必须把这些程序还原成压缩前代码，这项工作在程序装入内存之后，由病毒立即完成，而病毒自已本身不能被压缩。

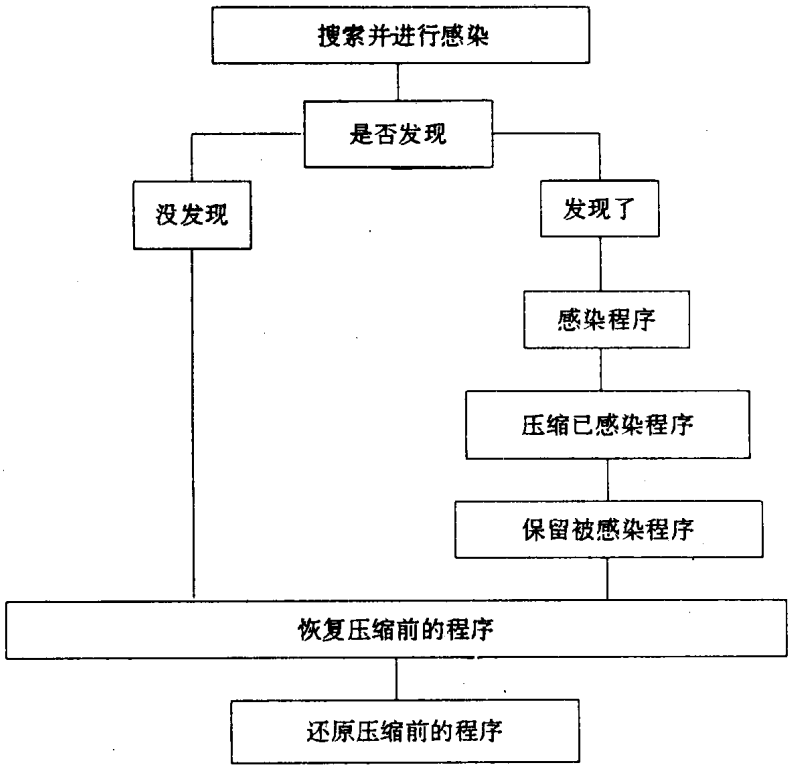

图1—1

图 1—1 的图解说明，病毒是怎样感染和压缩程序的过程流程图。

从理论上看，这是一个非常确切的病毒程序，可是，仔细分析这个程序的结构，你会注意到几个问题：

1. 因为在每个过程之前，都需要时间来还原被压缩的程序，所以，程序的运行需要额外增加时间。

2. 病毒始终不断搜索新的程序，一旦发现就进行感染并把其压缩。

3. 被压缩的程序至少大于病毒程序本身的 50%。另外，被压缩和感染的程序需要比原始程序占更多的存储空间。

4. 对于大部分用户来说，硬盘空间是负担得起的。

我们对于压缩病毒看法：

个人或家庭计算机用户在碰到压缩病毒时，会发现程序执行速度太慢，但是微型计算机和主机用户有足够的存储空间可用，所以，他们不必担心压缩病毒。

第二章 历史的回顾

现在很难精确地说出第一个病毒程序是在哪一天出现的。但大多数 PC 专家认为 MS—DOS 上的病毒始于 1986 年。大约 1989 年传入我国。

至今所知关于传染扩散的数学模式的著作有：N. T. J. 贝利所著《传染的数学理论》(1957)。美国在 70 年代及 80 年代初关于"蠕虫程序"和病毒方面的文章有：《计算机协会利用病毒功能在 User control 下给出病毒性的 APL 语言编译程序》(1974)和《蠕虫程序—分布式计算初探》(1982)。

由 Fred Cohen 编写的关于计算机病毒的《计算机病毒：理论与实验》首次在世界范围内引起了广泛的注意。

这本书畅销的原因是 Cohen 透彻且详细地对病毒进行了介绍；并写出了在计算机系统上的实验报告。在本章的第一节我们将讨论 Cohen 在实验中所得出的重要观点。

2.1 Fred Cohen 的实验

在《计算机病毒：理论与实验》一书的介绍中，Fred Cohen 试图向读者们解释病毒程序的规律：

"我们定义一个计算机病毒为：它可以通过即使是很小程序修改来传染给其它的程序，病毒通过对每个用户授权的方式来传染一个机器或网络系统，每个被传染上病毒的程序可以扮演病毒的角色来对其它的程序传染使病毒继续蔓延。"虽然这段描述给人们一个关于病毒如何工作的一般印象，但它没有包括传染识别的功能，这一点在以后所举的病毒例子中看到。

2.1.1 V—病毒

下面是用伪代码描述的一个很简单的 V—病毒数字。12345678 是这种病毒的标志字节。

```
program virus:=
{12345678;
subroutine infect-executable:=
        {loop:file = get-random-executable-file;
          if first-line-of-file=12345678 then goto loop;
            prepend virus to file;}
            subroutine do-damage:=
            {whatever damage is to be done}
            subroutine trigger-pulled:=
            {return true if some condition is satisfied}
          main program:=
            {infect-executable;
              if trigger-pulled then do-damage;
                goto next;}
                next;}
```

程序描述

infect_executable 子程序搜寻可执行文件，并检查这个文件是否包含病毒标志字节“12345678”。如果有，则已经被传染过，程序继续搜寻。如果未找到标志字节，病毒将自身放在文件的前面。

do_damage 子程序则带有任何可能的控制使命。

trigger_pulled 子程序查看是否满足某种条件，如果满足，则 trigger_pulled 为真。

main_program 首先传染一个正常的程序，再检测当前条件。如果条件满足则会进行某些操作。

2.1.2 休眠病毒

Cohen 描述了一种特别有欺骗性的计算机病毒变种，叫休眠病毒。这种病毒一般由系统时钟上的时间日期而激活。

在写关于病毒文章的时候，很多杂志的编辑及作者明显地爱写这种类型的病毒。几乎每种物刊都提及了在 4 月 1 日删除用户文件及数据的病毒例子。

Cohen 认为休眠病毒的主要威胁在于对多用户系统的侵害，他在书中写到：

“如果 V 传染给用户 A 的可执行程序 E，而用户 B 启动了这个程序，则 V 病毒也会传给 B 的文件。”

2.1.3 压缩病毒

Cohen 也确实开创了真正病毒的时代。压缩病毒就是一个真正病毒。在第 1.4 节中，我们论述了这种真病毒对用户并没有危害的原因。

```
program Compress_virus:=
{01234567;
subroutine infect_executable:=
        {loop:file = get_random_executable_file;
         if first_line_of_file=01234567 then goto loop;
         compress file;
         prepend compression_virus to file;}
main_program:=
         {if ask_permission then infect_executable;
         uncompress the_rest_of_this_file into tmpfile;
         run tmpfile;} }
```

根据 Cohen 的定义，这个程序具有感染其它程序的特性。因此，压缩病毒程序使被传染的程序减少了存储空间。这个例子说明 Cohen 是成功的，他的实验也是可以被接受的。

2.1.4 Cohen 的实验

Cohen 第一个实验是在 1983 年 8 月 10 日进行的，在南加利福尼亚大学，硬件环境为运行 UNIX VAX－11/750。

为运行程序进行了八个小时的专门工作。为了防止失控，实验包括几个安全因素——其内部跟踪及编码。