基于监督学习的
病毒检测技术研究

张波云 著

COMPUTER VIRUS DETECTION TECHNIQUE
BASED ON
SUPERVISED MACHINE LEARNING METHODS

WUHAN UNIVERSITY PRESS
武汉大学出版社

本书研究工作得到了国家自然科学基金面上项目（编号：61471169）、湖南省科技计划重大专项（编号：2017SK1040）、湖南省科技计划重点研发项目（2017NK2402）、湖南省社会科学基金项目（编号：12YBB090）、网络侦查技术湖南省重点实验室开放研究基金项目、网络犯罪侦查湖南省普通高等学校重点实验室开放研究基金项目和湖南警察学院学术专著出版项目的大力资助。

基于监督学习的病毒检测技术研究

张波云 著

COMPUTER VIRUS DETECTION TECHNIQUE
BASED ON
SUPERVISED MACHINE LEARNING METHODS

武汉大学出版社

图书在版编目(CIP)数据

基于监督学习的病毒检测技术研究/张波云著. —武汉：武汉大学出版社，2017.12

ISBN 978-7-307-19904-0

Ⅰ.基… Ⅱ.张… Ⅲ.计算机病毒—检测—研究 Ⅳ.TP309.5

中国版本图书馆 CIP 数据核字(2017)第 309191 号

责任编辑：田红恩　　责任校对：李孟潇　　版式设计：韩闻锦

出版发行：**武汉大学出版社**　(430072　武昌　珞珈山)

(电子邮件：cbs22@whu.edu.cn　网址：www.wdp.com.cn)

印刷：虎彩印艺股份有限公司

开本：720×1000　1/16　　印张：9.5　　字数：168 千字　　插页：1

版次：2017 年 12 月第 1 版　　2017 年 12 月第 1 次印刷

ISBN 978-7-307-19904-0　　定价：35.00 元

前　言

日益泛滥的病毒问题已成为信息安全的最严重威胁之一。据 Symantec 公司发布的《2016 互联网安全威胁报告》，2015 年该公司共捕获 4.3 亿个病毒样本，平均每天 117 万个新的病毒代码释放到互联网，较 2014 年增长了 36%。360 互联网安全中心发布的《2016 年中国互联网安全报告》显示，2016 年 360 互联网安全中心共截获 PC 端新增病毒样本 1.9 亿个，Android 平台新增恶意程序样本 1403.3 万个。如此数目庞大的病毒已经成为互联网的最大安全威胁，严重影响了世界各国的信息安全。由于当前病毒检测方法存在较多的局限性，迫切需要研究新的有效的病毒检测方法，以保障网络空间安全。

本书结合当前国内外计算机病毒领域的研究现状，总结作者在病毒研究领域近年来的研究成果，主要介绍基于有监督机器学习理论指导下的病毒自动化和智能化检测技术。重点分析病毒分类过程中的特征抽取和抽象表示方法，并利用机器学习算法提出多种病毒检测方法模型，在真实的病毒样本集上进行实验测试，并详细分析各种检测方法的性能数据指标。

本书内容分为三个部分，第一部分研究以 API 函数调用信息作为病毒行为特征的动态检测技术，与第 3 章、第 4 章和第 5 章对应。第二部分研究以从程序二进制比特流中抽取的 n-gram 信息作为病毒特征的静态检测技术，与第 6 章和第 7 章对应。第三部分是病毒动态检测与静态检测方法相结合的集成检测技术研究，与第 8 章对应。各章的内容安排简要描述如下：

第 1 章绪论，给出了病毒的形式定义，介绍了病毒的功能结构，以及病毒检测经典技术，着重对基于机器学习的病毒智能化和自动化检测技术的研究状况进行了综述，并指出了未来研究趋势。

第 2 章病毒检测基础。病毒理论研究的结论对病毒检测工程工作有重要指导意义，本章首先介绍了病毒理论研究中的一些重要基本定理。病毒对抗查杀的一个重要技术就是变形和多态，研究病毒检测的新技术有必要了解病毒的变形机理，本章第 2 节对此进行了阐述。本书主要研究当前最流行的 Windows 操作系统平台下的 PE 格式病毒，为方便理解后述章节内容，对 WinPE 病毒原理

在第 3 节中做了简介。

第 3 章基于多重朴素贝叶斯算法的病毒动态检测方法。介绍了多重朴素贝叶斯方法的工作原理，设计了病毒检测系统的框架，描述了基于 API 调用信息的特征提取与选择方法，最后详述了应用多重 Bayes 分类器进行病毒自动检测的算法。同时还说明了本书使用的实验数据集，并对评价检测器性能的指标参数进行了介绍。

第 4 章基于模糊模式识别的病毒动态检测方法。介绍了模糊模式识别数学模型，提出基于模糊模式识别的病毒检测方法新思路。检测系统将病毒与正常程序分别用模糊集方法来表示，然后应用“择近原则”进行模式分类。

第 5 章基于支持向量机的病毒动态检测方法。介绍在小样本集情况下具有良好泛化能力的支持向量机在病毒动态检测中的应用。检测系统以 API 函数调用短序为特征空间，应用 ReliefF 算法进行特征降维，通过实验确定了相关的核函数和错分惩罚因子，然后在数据集上用支持向量机进行学习和泛化。

第 6 章基于粗糙集属性约简的病毒静态检测方法。在借鉴传统的特征扫描技术的基础上，提出一种病毒静态分析和自动检测方法。检测引擎静态抽取程序的 n-gram 信息，根据特征的信息增益值选择特征，并采用粗糙集理论对所抽取的特征进行约简，消除冗余特征，最后将优化后的特征向量输入分类器，实现病毒的自动检测。本章重点研究了基于核的属性约简方法，并对其作了优化处理。

第 7 章基于集成神经网络的病毒静态检测方法。研究了集成神经网络作为模式识别器在计算机病毒自动检测中的应用。在 Bagging 算法的基础上，将基于信息增益的特征选择技术引入集成神经网络的构建中，提出了 IG-Bagging 集成方法。

第 8 章基于 D-S 证据理论的病毒动态与静态检测方法融合。研究了病毒动态检测与静态检测相结合的方法。注意到计算机病毒的动态检测与静态检测各有其优缺点，由此考虑可将二者结合起来，扬长避短，以提高病毒检测准确率。

本书研究工作得到了国家自然科学基金面上项目(编号：61471169)、湖南省科技计划科技重大专项(编号：2017SK1042)、湖南省科技计划重点研发项目(2017NK2402)、湖南省社会科学基金项目(编号：12YBB090)、网络侦查技术湖南省重点实验室开放研究基金、网络犯罪侦查湖南省普通高等学校重点实验室开放研究基金和湖南警察学院学术专著出版的大力资助。课题组成员鄢喜爱博士、张明键博士、段丹青博士和范强教授、周建华副教授、肖自红副教

授、唐德权博士生在项目研究和书稿撰写中付出了大量心血，在此一并致谢。

病毒技术的发展将不断促进反病毒技术的发展，反之亦然。只有不断结合一些最富吸引力的思想和发明，才能在这场永不停歇的反病毒斗争中坚持到底。由于计算机病毒的复杂性，在反病毒方面仍然需要进行大量的研究，加上作者知识和研究水平有限，书中难免会有疏漏或不当之处，恳请专家、学者不吝赐教。

目　录

第1章 绪 论

1.1 病毒基本概念

1.1.1 病毒定义

“计算机病毒”这个术语是由 L. M. Adleman 引入的，F. Cohen 博士第一次给出了关于计算机病毒的抽象定义。Cohen 关于计算机病毒的形式化定义很难准确易懂地翻译成自然语言，但他自己给出了一个著名的关于计算机病毒的非形式化定义：“一个计算机病毒是一个程序，它能够通过把自身(或自己的一个变体)包含在其他程序中来进行传染。通过这种传染性质，一个病毒能够传播到整个的计算机系统或网络(通过用户的授权感染他们的程序)。每个被病毒传染的程序表现得像一个病毒，因此传染不断扩散。”

定义 1.1 满足如下公式的序偶 (M, V) 称为图灵机 M 上的一个病毒集，所有病毒集的集合记为 $\overline{V}$：

$$\forall M \forall V(M, V) \in \overline{V} \Leftrightarrow \tag{1.1}$$

$$[V \in I^*] \wedge [M \in \overline{M}] \wedge (\forall v \in V \forall H_M \forall t, j \in \mathbf{N}) \tag{1.2}$$

$$[[P_M(t)=j] \wedge [\Xi_M(t)=\Xi_M(0)] \wedge [W_M(t, j+|v'|-1))=v]] \Rightarrow \tag{1.3}$$

$$(\exists v' \in V \exists t'>t \exists j' \in N) \tag{1.4}$$

$$[[(j'+|v'|) \leqslant j] \vee j+|v|) \leqslant j']] \tag{1.5}$$

$$[W_M(t', j'), \cdots, W_M(t', j'+|v'|-1)=v') \wedge \tag{1.6}$$

$$[(\exists t'')[t<t''<t'] \wedge [P_M(t'') \in \{j', \cdots, j'+|v'|-1\}]] \tag{1.7}$$

对应上面的形式化定义，每一行的直观解释如下：

(1.1)对所有的 M 和 V，序偶(M, V)称为一个病毒集当且仅当

(1.2)V 是一个 M 上带符号串的非空集，M 是一个图灵机，并且对 V 中的每个病毒 v，对 M 的所有历史 H_M，对所有的时刻 t 和带上单元 j，

(1.3)如果带的读写头在时刻 t 位于单元 j 上，并且机器 M 在时刻 t 处于开始状态，带上单元从 j 开始包含病毒 v，那么

(1.4)存在 V 中的一个病毒 v'，时刻 $t'>t$，和单元 j' 使得

(1.5)单元 j' 远离病毒 v

(1.6)带上单元从 j' 开始包含病毒 v'，

(1.7)并且存在一个 t 和 t' 之间的时刻 t''，病毒 v' 被 M 写在带上。

为了简洁，表达式 $a\overset{B}{\Rightarrow}C$ 用来表示以上定义中从第(1.2)行开始的部分，其中 a，B，C 分别是 v，M，V 的特定的实例，即

$$\forall(B,\ C)[(B,\ C)\in\overline{V}]\Rightarrow$$

$$[C\subset I^{*}]\wedge[B\in M]\wedge(\forall a\in C)[a\overset{B}{\Rightarrow}C]$$

以上的定义在所有图灵机上定义了谓词 $\overline{V}$。这个定义使得病毒集的一个给定元素可以产生任意数目的该集合的其他元素(依赖于带上的其他部分)。这提供了额外的一般性而没有不合理的复杂性或限制。最后，这里没有所谓的“条件病毒”，因为每一个病毒集中的元素必须总是产生该集中的另一个元素。如果愿意要“条件病毒”，那么可以不修改定义而增加一些条件，它导致或者避免病毒被当做其他部分的一个函数来执行。

一般来说，如果 $v\in\overline{V}$，则称 v 是一个相对于图灵机 M 的病毒，即 $\{v\in V\mid(M,\ V)\in\overline{V}\}$。

“v 对于 M 进化到 v'”当且仅当

$$[[(M,\ V)\in\overline{V}]\wedge[v\in V]\wedge[v'\in V]\wedge[v\overset{M}{\Rightarrow}\{v'\}]];$$

“v'对于 M 从 v 进化而来”当且仅当“v 对于 M 进化到 v'”；

“对于 M，v'是 v 的一个进化”当且仅当

$$(M,\ V)\in\overline{V}\wedge(\exists i\in N\exists v'\in V)$$
$$[[v\in V]\wedge[v'\in V]]\wedge$$
$$[(\forall v_k\in V')[v_k\overset{M}{\Rightarrow}v_{k+1}]]\wedge$$
$$[(\exists l,\ m\in\mathbf{N})[[l<m]\wedge[v_l=v]\wedge[v_m=v]]]。$$

和 F. Cohen 的途径不同，L. M. Aldeman 发展了一个基于递归函数论的计算机病毒理论，在他的定义中，一个计算机病毒是一个递归全函数 v，作用于所有程序 x 上(程序 x 的 Godel 编码)，使得 $v(x)$ 具有计算机病毒的典型特征：致损(injury)、传染(infection)和模仿(imitation)。在此基础上，Aldeman 证明

了更强的计算机病毒不可解定理，即：所有计算机病毒的集合是 Π_2 完备集。但其认为将任何具有破坏作用的程序都认定为病毒，这种定义有将病毒内涵扩大化的倾向。

在国内，专家和研究者对计算机病毒也做过不尽相同的定义，但一直没有公认的明确定义。直至 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》，第 28 条中明确指出："计算机病毒，是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。"该定义具有法律性、权威性，本书也使用此定义。

1.1.2 病毒功能结构

一个病毒主要由感染、载荷和触发等机制组成，病毒程序是一种特殊程序，其最大的特点是具有感染能力。病毒的感染动作受到触发机制的控制，病毒触发机制还控制了病毒的破坏动作。病毒程序一般由感染标记、感染模块、破坏模块、触发模块、主控模块等构成。

感染标记又称病毒签名。病毒程序感染宿主程序时，要把感染标记写入宿主程序，作为该程序已被感染的标记。感染标记是一些数字或字符串，通常以 ASCII 方式存放在程序里。病毒在感染健康程序以前，先要对感染对象进行搜索，查看它是否带有感染标记。如果有，说明它被感染过，就不再进行感染；如果没有，病毒就感染该程序。不同的病毒感染标记位置不同，内容不同。例如：巴基斯坦病毒感染标记在 BOOT 扇区的 04H 处，内容为 1234H；大麻病毒在主引导扇区或 BOOT 扇区的 0H 处，内容为 EA 05 00 C0 07；耶路撒冷病毒在感染文件的尾部，内容是 MsDos。

感染模块是病毒进行感染时的动作部分，感染模块主要做三件事：(1)寻找一个可执行文件；(2)检查该文件是否有感染标记；(3)如果没有感染标记，进行感染，将病毒代码放入宿主程序。

破坏模块负责实现病毒的破坏动作，其内部是实现病毒编写者预定的破坏动作的代码。这些破坏动作可能是破坏文件、数据，破坏计算机的时间效率和空间效率或者使机器崩溃。

触发模块根据预定条件满足与否，控制病毒的感染或破坏动作。依据触发条件的情况，可以控制病毒的感染或破坏动作的频率，使病毒在隐蔽的情况下，进行感染或破坏动作。病毒的触发条件有多种形式，如日期、时间、发现特定程序、感染的次数、特定中断的调用次数。

主控模块在总体上控制病毒程序的运行。其基本动作如下：(1)调用感染模块，进行感染；(2)调用触发模块，接受其返回值；(3)如果返回真值，执行破坏模块；(4)如果返回假值，执行后续程序。

感染了病毒的程序运行时，首先运行的是病毒的主控模块。实际上病毒的主控模块除上述基本动作外，一般还做下述工作：(1)调查运行的环境；(2)常驻内存的病毒要做包括请求内存区，传送病毒代码，修改中断矢量表等动作，这些动作都是由主控模块进行的；(3)病毒在遇到意外情况时，必须能流畅运行，不应死锁。

为精确起见，用伪代码对病毒的结构详细描述，如表1.1所示。

表1.1 **病毒的功能结构**

```
1. program virus: =
2. {1234567;
3.
4. subroutine infect-executable: =
5. {   loop:    file = get-random-executable-file;
6.             if first-line-of-file = 1234567 then goto loop;
7.         prepend virus to file;
8. }
9.
10. subroutine do-damage: =
11. {   whatever damage is to be done}
12.
13. subroutine trigger-pulled: =
14. {   return true if some condition holds}
15.
16. main-program: =
17. {   infect-executable;
18.     if trigger-pulled then do-damage;
19. goto next;}
20.
21. next:}
```

表中相关符号含义约定如下：

: = 表示定义

: 表示语句标号

; 表示语句分隔

~ 表示非

{ }表示一组语句序列

… 表示一组省略的无关紧要的代码。

病毒从主程序开始，先执行 infect-executable 子程序，病毒程序(*V*)搜索一个未被病毒感染的可执行程序(*E*)。根据程序开始行有无“1234567”判定程序是否被病毒感染。如果开始行为 1234567，则表示程序已被病毒感染，不再进行传染；如果开始不是 1234567，则表示程序没有被病毒感染，需要运行 random-executable，并把病毒(*V*)放到可执行程序(*E*)的前面，使之成为感染的文件(*I*)，prefend 语句的作用就是将(*V*)放到(*E*)的前面。

接着，病毒程序(*V*)检查激发条件是否为真。如果为真，则执行 do-damage 子程序，即进行破坏，最后(*V*)执行它所附着的程序；如果激发条件不满足，则执行 next 其他的子程序。

当用户要运行可执行程序(*E*)时，实际上是(*I*)被运行，它传染其他的文件，然后再像(*E*)一样运行，当(*I*)的激发条件得到满足时，就去执行破坏活动，否则除了要传染其他的文件要占用一定的系统开销外，(*I*)和(*E*)都具有相同的功能。

一个病毒程序的作用，其关键在于动态执行过程中具有病毒传递性。需要指出，病毒并不一定要把自身附加到其他程序前面，也不一定每次运行只感染一个程序。如果修改病毒程序(*V*)，指定激发的日期和时间，并控制感染的多次进行，则有可能造成病毒扩散到整个计算机系统，从而使系统处于瘫痪状态。

1.1.3 病毒与恶意代码

值得注意的是，有人很宽泛地定义病毒，将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，此定义将病毒等同为恶意代码。而实际情况中，病毒、蠕虫和特洛伊木马等常常融为一体，到 2017 年为止商用的 Antivirus 工具也同时防范不同形式的恶意代码。

Grimes 将恶意代码定义为，经过存储介质和网络进行传播，从一台计算机系统到另外一台计算机系统，未经授权认证破坏计算机系统完整性的程序或代码。它包括计算机病毒(Computer Virus)、蠕虫(Worms)、特洛伊木马(Trojan Horse)、逻辑炸弹(Logic Bombs)、病菌(Bacteria)、用户级 RootKit、核心级 RootKit、脚本恶意代码((Malicious Scripts)和恶意 Activex 控件等。

蠕虫是经常和计算机病毒一起出现的术语。实际情况中也如此，不仅有些计算机病毒的定义涵盖了蠕虫，而且在真实环境中，一些恶意程序的确既具有病毒的特征，也具有蠕虫的特征。

RFC1135 将蠕虫定义为“一段独立运行的程序，通过消耗宿主的资源来维

护自身，并能够向其他机器传播自身的完全工作版本”。与计算机病毒不同，蠕虫不需要把自身附加在宿主程序上。

特洛伊术马(简称木马)是一个程序，它执行一些无授权的、木马编制者有意设置的操作，这些操作如果用户知道通常不会批准。一些人认为病毒是木马的一个特例，即能够传播到其他程序上的木马。另一些人认为，一个病毒如果除了复制自身以外，不做其他有意地破坏，则不是一个木马。尽管存在争论，许多人使用术语“木马”指称那些没有自我复制功能的恶意软件，因此木马的集合和病毒的集合是完全分离的。

表1.2列举几种主要的恶意代码类型及其相关的说明。从中可以看出恶意代码具有非授权性和破坏性的共性。

表1.2 主要恶意代码

类 型	说 明	特 点
计算机蠕虫	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
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、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

1.1.4 经典病毒检测方法

计算机病毒的检测方法主要有长度检测法、病毒签名检测法、特征代码检测法、校验和法、行为监测法、感染实验法、生物免疫法、人工智能方法等。这些方法依据的原理不同，实现时所需开销不同，检测范围不同，每种方法均

有其自身的优缺点。

1. 长度检测法

病毒最基本特征是感染性，感染后的最明显症状是引起宿主程序增长，一般增长几百字节。在现今的计算机中，这一变化常常不易引起注意，文件长度莫名其妙地增长，是病毒感染的常见症状。

长度检测法，就是记录文件的长度，运行中定期监视文件长度，从文件长度的非法增长现象发现病毒。

知道不同病毒使文件增长长度的准确数字后，由染毒文件长度增加大致可断定该程序已受感染，从文件增长的字节数可以大致断定文件感染了何种病毒。以文件长度是否增长作为检测病毒的依据，在许多场合是有效的。但是，众所周知，现在还没有一种方法可以检测所有的病毒。长度检测法有其局限性，只检查可疑程序的长度是不充分的，因为：

(1)文件长度的变化可能是合法的，有些普通的命令可以引起文件长度变化；

(2)经常进行的不知不觉地对程序的修改可能引起长度变化；

(3)不同版本操作系统也可能造成此类变化；

(4)某些病毒感染文件时，宿主文件长度可保持不变。

上述情况下，长度检测法不能区别程序的正常变化和病毒攻击引起的变化，不能识别保持宿主程序长度不变的病毒。

2. 病毒签名检测法

病毒签名(病毒感染标记)是宿主程序已被感染的标记。不同病毒感染宿主程序时，在宿主程序的不同位置放入特殊的感染标记。这些标记是一些数字串或字符串，例如 1357、1234、MSDOS、FLU 等。不同病毒的病毒签名内容不同、位置不同。经过剖析病毒样本，掌握了病毒签名的内容和位置之后，可以在可疑程序的特定位置搜索病毒签名。如果找到了病毒签名，那么可以断定可疑程序中有病毒，是何种病毒。这种方法称为病毒签名检测方法。该方法的特点是：

(1)必须预先知道病毒签名的内容和位置，要把握各种病毒的签名，必须剖析病毒。剖析一个病毒样本要花费很多时间，每一种病毒签名的获得意味着耗费分析者的大量劳动，是一笔很大开销。掌握大量的病毒签名，将有很大的开销。剖析必须是细致、准确的，否则不能把握病毒签名。

(2)可能虚假报警。一个正常程序在特定位置具有和病毒签名完全相同的代码，这种巧合的概率是很低的，但是不能说绝对没有。如果遇到这种情况，

病毒签名检测法不能正确判断，会错误报警。由于这种误报概率很低，病毒签名法可以相当准确地判断出病毒的种属。

3. 特征代码检测法

病毒签名是一个特殊的识别标记，它不是可执行代码，并非所有病毒都具备病毒签名。某些病毒判断宿主程序是否受到感染是以宿主程序中是否含有某些可执行代码段做判据，因此，反病毒专家也采用了类似的方法检测病毒，在可疑程序中搜索某些特殊代码，称为特征代码检测法。

特征代码法被普遍用于各商业反病毒工具软件中。一般认为特征代码法是检测已知病毒的最简单、开销最小的方法。

特征代码法的实现步骤如下：

(1)采集已知病毒样本。

(2)在病毒样本中，抽取特征代码。

在抽取特征代码时应依据如下原则：

①抽取的代码比较特殊。不大可能与普通正常程序代码吻合。

②抽取的代码要有适当长度。一方面维持特征代码的唯一性，另一方面又不要有太大的空间与时间的开销。

③在感染多种文件的病毒样本中，要抽取不同种样本共有的代码。

(3)将特征代码纳入病毒数据库。

(4)打开被检测文件，在文件中搜索、检查文件中是否含有病毒数据库中的病毒特征代码。

(5)如果发现病毒特征代码，由于特征代码与病毒一一对应，便可以断定被查文件中患有何种病毒。

采用病毒特征代码法的检测工具，面对不断出现的新病毒，必须不断更新版本，否则检测工具便会老化，逐渐失去实用价值。

特征代码法的优点是：检测准确、快速；可识别病毒的名称；误报警率低；依据检测结果，可做杀毒处理。

其缺点是：不能检测未知病毒；搜集已知病毒的特征代码，费用开销大；在网络上效率低(在网络服务器上，因长时间检索会使整个网络性能变坏)。

该种检测法有如下特点：

(1)依赖于对病毒精确特征的了解。必须事先对病毒样本做大量剖析。

(2)剖析病毒样本要花费很多时间。从病毒出现到找出检测方法有时间滞后。

(3)如果病毒中作为检测依据的特殊代码段的位置或代码被改动，将使原