



启发式分析

——检测未知病毒

反病毒软件除了要被动地检测出已知病毒外，还更应该主动地扫描未知病毒。

那么，反病毒软件的扫描器到底是如何工作的呢？

David Harley (大卫·哈利)

信息安全研究和咨询顾问



Andrew Lee (安德鲁·李)

首席研究主管

ESET 公司

本文由 NOD32 中国官方论坛组织翻译

<http://www.nod32club.com/forum/>

目录

	页数
简介	5
了解检测	6
病毒	6
蠕虫	7
非复制型恶意软件	7
启发式究竟指什么？	9
特征码扫描	10
启发式的对立面	11
广谱防毒技术	12
我绝对肯定	13
灵敏度与误报	14
测试的争议	16
结论：启发式技术的尴尬	19
参考文献	21
术语表	22

作者简介

大卫·哈利

大卫·哈利从 20 世纪 80 年代末就开始研究并撰写有关恶意软件和其它电脑安全问题的文章。2001 – 2006 年，他是英国国家健康中心 (UK's National Health Service) 的国家基础设施安全管理人员，专门研究恶意软件和各种形式的电子邮件的滥用，负责威胁评估中心 (Threat Assessment Center) 的运作。2006 年 4 月起，他从事独立作家和安全技术顾问的相关工作。

大卫的第一部重要著作是《病毒揭秘 (Viruses Revealed)》(Harley, Slade, Gattiker)，奥斯本的计算机病毒防护全面指南。他参与编写并审核了很多其它有关计算机安全和教育方面的书籍，以及大量的文章和会议论文。最近，他作为技术编辑和主要撰稿人参加了《企业防御恶意软件的 AVIEN 指南 (The AVIEN Guide to Malware Defense in the Enterprise)》的编写。这本书将在 2007 年由 Syngress 出版社出版。

联系地址：8 Clay Hill House, Wey Hill, Haslemere, SURREY GU27 1DA

电话：+44 7813 346129

网址：<http://smallblue-greenworld.co.uk>

安德鲁·李

安德鲁·李，国际信息系统安全协会会员，ESET 公司的首席研究员。他是反病毒信息交换网络 (Anti-Virus Information Exchange Network, AVIEN) 和它的姊妹团体——AVIEN 信息和早期预警系统 (AVIEN Information & Early Warning System, AVIEWS) 的创建者之一。他是亚洲反病毒研究协会 (The Association of Anti-virus Asia Researchers, AVAR) 的会员，也是 WildList 国际组织 (一个负责收集流行性计算机病毒活体样本的组织) 的记者。在加入 ESET 公司之前，他作为一名

资深安全管理人员帮助英国的一个大型政府组织防范恶意软件。

安德鲁写了大量的有关恶意软件方面的文章，并经常在包括 AVAR、EICAR、Virus Bulletin 在内的各种会议和活动中发言。

ESET 公司

地址：610 West Ash Street, Suite 1900, San Diego, California 92101, U.S.A.

电话: +1.619.876.5400

传真: +1.619.876.5845

网址: <http://www.eset.com>

简介

俗话说“不太明白不要紧，错误认识要不得”

计算机领域一些最持久的误解是与病毒和反病毒技术相关的。在早期的反病毒研究中，反病毒软件只能明确检测已知病毒的说法受到了广泛的支持。但当时这种说法并不完全正确，一些反病毒程序并不检测特定的病毒，而是检测并且阻止类病毒行为或者文件中的可疑变化。而这在现在看来是完全错误的。

商业反病毒系统将多种更广谱的方法作为特征码扫描的补充，这些方被统称为启发式分析。此外，大多数当代的反病毒产品能够检测多种恶意软件（malware 是“malicious” and “software”这 2 个单词合并写成），而不仅仅是病毒。这些手段可能和其它的安全技术（如垃圾邮件和钓鱼信息的检测）组合运用。

本文的目的在于解答一些关于反病毒技术工作原理的困惑，并且阐明我们对病毒防护机制，特别是启发式分析所抱的期望中，哪些是现实的。

关于启发式扫描的特点，本文将做较为详细的讨论。现在我们简单的描述下启发式分析：分析某个未被定义为恶意软件的程序是恶意软件（恶意的或者病毒的）可能性的方法。

了解检测

大多数我们提到的病毒可能被称为恶意软件更为准确

一个反病毒程序检测什么？实际上非常多，包括一些技术上来说并不是病毒的程序。大多数我们提到的病毒可能被称为恶意软件更为准确。讽刺的是许多在市场上销售的专杀产品（比如检测间谍或者木马程序的）都宣称是必备的，他们的理由是反病毒软件只能检测病毒。

事实上，商业的反病毒产品涉及的恶意软件的范围比大多数专杀程序要广。专杀程序可能在它的特定范围内检测到更多的威胁，但是这不仅与程序检测已知威胁及其类型的能力有关，还包括以下几个因素：

- ◆ 程序的广谱检测能力
- ◆ 区分不同恶意软件的标准
- ◆ 样本共享机制（相比其它恶意软件检测领域的厂商，反病毒厂商为此采取了行之有效的交换途径）

下面的章节讨论了 3 个主要的恶意软件类型。完整的恶意软件分类方法将不在本文讨论的范围内。

病毒

期待反病毒软件检测病毒，当然是合情合理的。因为多年来反病毒软件在检测病毒方面如此成功，也正是由于这部分原因，以至于它检测其它类型恶意软件的能力被低估了。

虽然病毒有很多种定义，但一种被恶意软件研究者普遍接受的定义是“病毒是一种计算机程序，它通过将复制自身（或者自身的变体）修改计算机中的其他程序，以达到感染目的” [1, 2]。

多年来反病毒软件在检测病毒方面如此成功，也正是由于这部分原因，以至于它检测其它类型恶意软件的能力被低估了。

这个定义涵盖了很多种类型的病毒，包括：

- ◆ 引导型病毒，硬盘分区病毒
- ◆ 文件型病毒（寄生病毒）
- ◆ 混合型病毒
- ◆ 宏病毒和脚本病毒

尽管有些病毒类型现在已经很少见了（比如引导型病毒和硬盘分区病毒），但是反病毒程序一般都能检测在该平台上（有时包括其它平台）发现的这类已知病毒。通常，反病毒软件也善于通过启发式的方式检测新的或未知的病毒种类。

蠕虫

业界从未在蠕虫的定义上真正地达成一致，如科恩所说“蠕虫是一种病毒的特例” [1]，但不论蠕虫是怎样的特例，反病毒软件通常都能检测它们。

对于蠕虫的不同定义甚至比针对病毒的还要多，但是大部分反病毒研究者将蠕虫定义为一种以非寄生方式复制的程序，也就是说，它不把它自身附加到一个寄主文件上。邮件群发者可以描述为一种特殊类型的蠕虫。多数反病毒厂商将这种通过电子邮件传播的恶意软件视为蠕虫，但是一些邮件群发者具有纯病毒的特点（比如，Melissa 事实上是一种纯病毒，一种能够像蠕虫一样传播的宏病毒，而 W32/Magistr 则是一种文件型病毒）。

对于新型蠕虫变种的检测，厂商同样有很好的手段。比如说，新型邮件群发器几乎在出现的同时，就被通信安全服务供应商及其系统列入了黑名单。

非复制型恶意软件

这个定义是根据上文的定义得出来的，即如果一个恶意软件不具有复制能力，那它就不是病毒或者蠕虫。但是这并不意味着反病毒软件不能检测到它，除非其不具有威胁。

要说明的是，即使当反病毒厂商过去常以非复制型的对象不是病毒为由拒绝对它们的检测时，一些非自我复制的对象（它们当中的一些甚至不是可执行程序，更不要说是有恶意的了）仍然能被检测到并且报毒。例如：

- ◆ 未遂病毒(复制自身失败的病毒)和损坏的病毒
- ◆ 垃圾文件
- ◆ 与病毒相关的非病毒程序，如病毒原体，释放器，病毒生成器
- ◆ 合法的测试程序如 EICAR 测试文件[4]

许多已传播多年的非复制型病毒，由于管理不善，仍然被一些评测机构作为样本

许多已传播多年的非复制型病毒，由于管理不善，仍然被一些评测机构作为样本，用于针对杀毒软

件的测试中。绝大部分厂商早已不再拒绝将这些病毒的特征码添加到其产品的病毒库中，避免因没有检测出这些病毒样本，而取得不理想的测试成绩。遗憾的是，日趋复杂的启发式扫描引擎，也只是刚好跟上了杀毒软件测试者不断更新的测试手段，有时新的测试手段也并非恰当。本文的后面部分会简略地分析测试产品启发式扫描能力时，技术上可接受的方式。

人们了解最多的非复制型恶意软件是木马程序（或简称为木马）。木马程序声称能执行一些有用的操作或提供一些必要的功能，可能也确实如此。但同时它也会执行一些用户并不希望或不需要的操作。这包括一系列特定的恶意软件：

- ◆ 病毒释放器；
- ◆ 键盘记录器；
- ◆ 破坏性木马程序；
- ◆ 下载者；
- ◆ 间谍程序；
- ◆ 广告程序；
- ◆ 内核后门与 **stealthkits**
- ◆ 玩笑程序；
- ◆ 僵尸程序（机器人程序，远程控制木马，DDoS 客户端等）

可自我复制的恶意软件（如病毒）有时也被认为是木马程序（或称为木马型病毒，这表示这种木马通过将一个以前合法的程序损坏，修改或替换而引起破坏），大部分人可能会发现这样的分类所带来的困惑多于帮助。检测出所有非复制型恶意软件要比检测出所有形式的病毒更加困难，因为不仅要测试程序的复制能力，还要测试程序的一系列作用。

争论的焦点在于，判断一个程序是不是木马程序（或恶意软件），是否应更多根据其目的来界定，而非功能来界定。例如，一个键盘记录程序如果是经过用户授权或用户自愿安装的，即使它的功能与木马程序是相同的，那么它也不算是木马程序。这会给检测带来问题，因为电脑在判断目的方面的能力弱于人类。

间谍软件和广告软件（可能由于媒体的过多关注，以及大量针对性产品的存在）已经被划分为了不同恶意软件的子类。尽管人们经常争论，广告软件不一定是恶意软件，但这种区别是大多情况下没有意义的。而同样的争论也适用于该类别的几乎所有其他项目，因为一个程序的行为并不能作为判定恶意软件的标准，而是在于程序员的不良意图与用户期望值之间存在着差别。

启发式究竟指什么？

“启发式”（Heuristic）是指探索和发现的行为或过程。牛津英文词典将启发式定义为“让某人能够自主的探索和学习”或者（在计算领域）“仅依靠宽泛的定义，或者依靠不断尝试和汲取失败经验教训的方法来解决问题[6]。”韦氏词典将其定义为“依靠实验尤其是反复试验，通过尝试和汲取失败经验教训的方法，来帮助学习、探索或者解决问题”或者（在计算领域）“一种利用自主学习的手段（通过反馈的评估）来提高表现，以探索解决问题的技术[7]。”

启发式程序通常被认为是一种具有人工智能的应用程序，而且也是一种解决问题的工具。启发式程序的编写，例如用于专门系统的程序，建立于一些从经验中提取的规则，它通过不断积累经验产生更好的解决方法，并且增加自己的知识库。

启发式分析使用基于规则的方法来诊断一个有潜在威胁的文件(或信息，如果是分析垃圾邮件的话)。

当启发式分析被用于处理恶意软件时（当然还包括垃圾邮件和流氓软件），尽管涉及反复试验和从经验中学习的原理，却又有更多限定的含义。启发式分析使用基于规则的方法来诊断一个有潜在威胁的文件(或信息，如果是分析垃圾邮件的话)。分析引擎的工作是基于自身规则库的，它依据规则检查恶意软件存在的可能性，当找到一个匹配的规则时就为其分配一个分值。如果这些分值达到或超过了一个阈值[8]，这个文件就会被标记为可疑文件（或者潜在的恶意软件、垃圾邮件）并进行处理。

某种意义上来说，针对反恶意软件的启发式技术，尝试的是模拟人类的智能分析方法。与恶意软件分析人员设法判定某个软件的行为和动作相同，启发式分析执行相同的智能决策过程，有效地担当虚拟分析师的角色。恶意软件分析人员从出现的新生威胁中不断学习，并将他的知识通过编程应用于启发式分析器，以提高今后的检测率。

启发式编程在反病毒软件性能中有着双重的任务：速度和检测。事实上，“启发式”这个术语在其他科学领域也有类似的含义[9]；即专注于通过达到“足够好”的结果(尤其指数据吞吐速度)而非“完美的”结果来提高性能。当已知病毒的数量与日俱增，就需要提高检测速度。否则，增长的恶意软件数量所带来额外扫描时间，会降低系统的使用效率。

当已知病毒的数量与日俱增，就需要提高检测速度。

尽管当今的启发式引擎性能已经大幅提升，但在用户看来，启发式（甚至非启发式）扫描所带来负

面影响，可能会超过其检测率提高所带来的优势，这是危险的。人们普遍相信启发式扫描器通常比静态扫描器慢，但在特定的情况下这一点不再正确。

早期的启发式扫描使用经过优化的简单检测模式，它只搜索目标中可能存在特定病毒的部分文件。

（一个简单的例子：如果一个病毒只将核心代码存储在被感染文件的头部和尾部，就没有必要扫描整个文件）。这样就减少了扫描的资源占用也降低了误报的风险。

对于在正常情况下，病毒不可能存在的位置检测出了病毒，体现的不仅是劣质检测方法的副作用，同时也是劣质检测设计程序的表现。例如，一些测试者尝试将病毒代码随机的插入一个文件或者其他可被感染的目标，以此来测试杀毒软件能力。同样的，对于一些特殊类型的目标，如文件或者引导区，可以选择性的扫描其中针对性的恶意软件类型，这种方法有时被描述为“过滤”。毕竟，没有理由在引导区上去扫描宏病毒代码。

然而，一种文件类型被正确识别，不能成为该文件未被感染的实际证据。举例来说，将 Microsoft Word 文档植入恶意的可执行文件，长期以来就是商业间谍和信息盗窃者的主要攻击手段。同样的，恶意软件编写者经常寻找一些正常情况下不可能包含可执行代码的文件，通过修改运行环境等方法使其能够包含可执行代码进而作为攻击目标。例如 W32/Perrun 病毒，将自身添加到 JPG 和 TXT 类型的文件中，一般不会运行，一旦操作环境做了特殊的改变之后，它就能释放出代码并运行。

特征码扫描

特征码检测指一种精确地检索一组匹配字符串的直接模式，每个病毒及其变种的特征都包含在扫描器的定义库中，而这些特征是不会出现在未被感染的文件中的。一些反病毒研究者并不赞成[2]将特征码扫描的称作“检索字符串”或者“扫描字符串”，但这似乎并没有影响反病毒软件公司例行的使用这个称谓。

事实上，许多病毒无法通过检索静态的字符串而被识别

这个术语的缺陷在于它延续了扫描器工作原理的一种陈旧观念，虽然其他可供选择术语也会引起争论。

使用“特征码扫描”这个术语的真正的困难在于：

- ◆ 它延续了“特征码扫描”是杀毒软件唯一的检测方式的误解。事实上，许多病毒无法通过检索静态的字符串而被识别。

- ◆ 它暗示每个病毒都有一个不同的字符串，所有扫描器均可统一识别。事实上，不同的扫描器可能会检索不同的字符串（使用不同的算法）来检测同一个病毒。

一些资料给人的印象是扫描器查找的是简单的文本字符串而非字节序列，这就更加混淆了问题。查找简单文本字符串的方法通常是不可靠的，不仅编程效率低下而且对于很多恶意软件类型完全无效。这也很容易被病毒作者，或者实际上，任何能够编辑文件的人所利用，并且有极大的误报风险。

通配符和类 UNIX 系统的规则表达，使字符串检索有了更多的灵活多样性。扫描器可以避免只查找一个静态的字符串（一个固定的字节序列），而是在其它字节或字节序列（噪音字节）夹杂其中的情况下，也能辨认出一组与病毒有关的字符串。噪音字节的一个简单的例子是插入一个 NOP（空操作）指令，这个指令的功能只是占用处理器时间而无任何实际的操作。

这一系列基本字符串检索方面的改进，能够检测出一些加密病毒和变形病毒种类[8]。可是，即使经过了这样的改进，字符串扫描还是无法特别有效地扫描混合型病毒。而复杂的变形病毒的出现，实际上使得一些缺乏高级检测技术的扫描器淡出市场[8,11]。

在当今反病毒技术领域，针对特定病毒的算法扫描，通常基于在虚拟机内模拟执行代码并进行解析的方法。虚拟化和模拟技术，可以被用来应对无意或有意的欺骗方法，如加壳、压缩或加密。一旦一个文件的欺骗方法被识破，它就能通过杀毒软件的扫描过程进行算法或者启发式分析。

虚拟机也在启发式分析中担当重要角色，而且取得了很大的成功，尽管要使模拟环境与当今 Windows™ 环境具有同样复杂性仍有许多难度。（然而我们需要理解的是模拟不可能是完美的，潜在的延迟影响（如处理时间的增加）也是明显的，随测试文件的不同有相应的变化。）

复杂的变形病毒的出现，实际上使得一些缺乏高级检测技术的扫描器淡出市场

启发式的对立面

关于商业反病毒软件只能检测已知的恶意软件及其变种和亚种的说法可能不再像以前那样普遍了。然而，部分取而代之的是另一种不很流行的说法，即病毒特征码扫描器和启发式扫描器是两种完全不同类型的扫描器。

事实上，我们知道启发式分析的使用已经有十年之久，而“已知病毒”扫描器利用启发式技术，优化病毒处理的历史则更久。启发式分析也在相关的应对措施，如行为拦截器和监控器，以及完整性检查

中占有一席之地。

在某种意义上，与反病毒软件启发式分析的相对的不是特征码扫描而是算法扫描，特征码扫描是其中的一种特殊的情况。

算法扫描，与其他形式的算法汇编一样，以数学公理[13]为基础。业界所说的算法扫描，通常认为是基于一种算法(而非简单的搜索一个静态字符串，或一种固定的字符串)针对特定目标病毒的检测。

当然，在日常生活中，上面提到的启发式分析也被视为一种更为普遍意义上的算法。但是将算法一词和病毒特征关联使用(因此有一些误导)，在业界已经非常广泛，所以不容忽视[12]。启发式通常指使用分值算法判断被扫描对象恶意与否，而非明确识别界定恶意软件类别的方法。

在某种意义上，与反病毒软件启发式分析的相对的不是特征码扫描而是算法扫描，特征码扫描是其中的一种特殊的情况

广谱防毒技术

启发式分析通常被视为一种针对同类病毒的广谱检测机制，而不是针对特定病毒的检测机制。人们常常想不到的是，这句话反过来理解也是正确的，广谱解决方案的部分诊断过程同样包含了启发式的规则。比如说：

- ◆ 邮件网关过滤器是使用预定的规则，来判断哪些文件类型和文件名称可以作为附件。一些过滤技术能够很好地应对明显的病毒威胁，例如以.LNK，.JPG，.EXE 为扩展名的文件，但在拒收整批可执行文件时显得缺乏灵活性。另一些过滤方法使用更先进的技术，例如扫描的文件头并核对是否符合特定的文件类型。这样一来，就能大大降低误报或者漏报的风险。
- ◆ 变化检测使用的规则是，一旦目标文件发生变化，即视为可疑文件，由于二进制文件改变文件校验码的情况很多(比如自我修改代码、重新合并代码、重新配置、压缩成可执行文件，补丁或程序升级等)，这种原始的依赖于文件变化的检测技术(就是说，只要文件发生变化，则认为遭受感染)会带来较高的误报率。然而，变化检测对于针对病毒本身的扫描是有效果的。业界公认的一项技术就是将目标文件与其校验码进行比对，只有在校验码发生变化时才对其进行完全扫描。这样就避免了对没有发生变化的文件进行扫描，从而节省大量的时间。这就是一些杀毒软件首次扫描所需的时间要远远大于后续扫描时间的原因。

◆ 行为监控与拦截，是就应用程序的行为进行判断并做出相应反应的一种技术，是早期杀毒软件就开始采用的技术之一。这种技术能够和启发式很好地衔接，进一步加强行为阻止能力并减少误报率。传统杀毒软件的行为监控倾向于检测两种类型的代码行为：自身复制和潜在危害。

——复制型代码从定义来看，极有可能是病毒(或者蠕虫，取决于代码类型和您倾向的定义)。反复制方法有它的优点，就是依靠编程能够比较容易甄别系统发出的代码复制请求，尤其是比较浅显的代码。一比一复制自身的病毒(即非多态性病毒)，对比释放变体的病毒而言，识别起来要容易很多。

——潜在的破坏性代码很可能反映恶意活性载体的存在。如果没有活性载体，或者活性载体不具备明显的破坏性，这种方法的效力也就随之降低了。一些破坏性行为，比如说删除文件，会比较容易通过编程来检测而另一些种类，比如说广告程序，以及令人尴尬或反感的色情信息或图片，屏闭掉会比较困难一些。另一方面，成功的检测出活性载体，也给非复制型恶意软件(比如木马和其他非病毒程序)的检测带来一定的益处。但这里需要留意的是，通过删除文件的行为来判断是恶意的方法常常并不可靠，因为很多程序在日常使用中，会经常合法地删除或者覆盖过时的配置文件和数据。

我绝对肯定

病毒的识别是介于两种要素的平衡：避免漏报（没有检测出存在的病毒感染）和误报（错误地报告不存在的病毒感染）。2006年初的几个月中，数款主流杀毒软件都出现了一系列的误报现象，这说明扫描技术的优化和进步并没有消除误报的风险。

使用启发式是不太可能消除误报的，启发式的定义说明了其在一定程度上需要进行错误的尝试。前面谈到，启发式编程的目的在于，能够达到“足够好”的检测结果，而并非完美的结果。那么，问

题出在哪里呢？

病毒的识别是介于两种要素的平衡：避免漏报（没有检测出存在的病毒感染）和误报（错误地报告不存在的病毒感染）

检测已知病毒最安全的手段，是扫描文件的每一个字节，并与病毒体中共同存在的校验码进行比对，从而确定是否遭受感染。这一过程常被称作“精确识别”。

识别率是一种衡量标准，用来考验杀毒软件检测并认定病毒样本为某个病毒或其变种的能力。因此，精确识别首先是精确度，病毒代码的每一个恒定字

节都要考虑到。尽管人们都希望这种精确度能够应用到针对每个病毒的扫描当中，在现实世界中却很少这样做，因为这意味着对扫描时间和系统资源带来潜在的负面影响，而且通常来说，这种精确程度也是不必要的。

“半精确识别”这一术语，特指识别方法在应用过程中，只能保证不会因为使用了不当的清除方法，造成目标文件的损坏[2]。检测和清除带来的问题不尽相同。一些杀毒软件厂商长期以来，提倡替换程序被感染的二进制字节，而不是清除染毒部分，主张将精力更多的集中在检测上。对于有些情况，比如内核后门和 **stealthkits** 就是很好的例子，将木马程序替换为合法程序就意味着安全软件只能删除文件，而不能清除文件。这时候，通常管理员或者用户就必须还原合法程序，因为自动还原是不现实的，甚至也是不安全的。

近年来，恶意软件的发展已经从单一感染文件，演变成对用户操作系统的控制（比如修改注册表）。这使得恶意软件一旦得逞，彻底清除起来要困难的多。清除不干净（或不正确）的话，可能会使操作系统受损，甚至无法使用，有时不得不采取极端的做法，比如重新安装操作系统或应用软件，并恢复数据备份。

然而，当恶意软件可以采用启发式或广谱特征法主动检测到时（即有机会感染目标系统以前），这个问题不会大量涌现，除非恶意目标（病毒或木马病毒）需要用非感染的形式保留（比如说，目标文件含有用数据）。

“广谱检测”是指通过扫描，寻找一系列已知变种的方法，使用的是可以检测到所有变种的搜索字符串。找到相同的字符串就可以检测出已知变种，如果需要按照分值机制来认定的话就是启发式检测。否则就是一种特殊的特征码检测。一些检测系统使用的是综合的方法，在广谱检测的基础上添加了分值认定系统，以确定病毒变种或族群的类属关系，并按照不同近似等级排比。例如，近似度达到一定程度时，扫描会报告“某个的变种”，如果低于这个程度，就会报告“可能是某个的变种”。

“广谱检测”是指通过扫描，寻找一系列已知变种的方法，使用的是可以检测到所有变种的搜索字符串

灵敏度与误报

启发式分析的精确度依赖于预设规则的强度。对于扫描器来说，目标恶意软件如果是新的，那么分析器输出的结果将不是一个简单的二元判定（“此为已知病毒 XXX”或“这不是已知恶意软件”）。（启发式扫描器的）最有说服力的扫描结果取决于能够在一个由高（尽可能地保持最低限度的误报率）到低（检测尽可能多的新病毒）的连续区间找到一个阈值。高强度的响应会在高误报的风险下尽可

能多地检测到新病毒，而低强度的响应则适合于误报不可接受的场合。

最有说服力的扫描结果取决于能够在—个由高(尽可能地保持最低限度的误报率)到低(检测尽可能多的新病毒)的连续区间找到一个阈值

对于一款杀毒软件来说，在缺省设置(启发式关闭)与设置开启启发式之间提供选项是常见的(因为我们已经指出所有的扫描引擎都采用了某种程度的启发式，或许按照基本启发式作为缺省设置会更准确一些)。某些厂商还会将被动启发式和主动启发式区分开来。虽然在这两种情况下，扫描器都会在代码中扫描可疑特征，但是在主动模式下，扫描器会使用一个模拟环境来动态执行并跟踪代码的行为。在被动模式下，它就只简单地静态检查代码。

如下我们可以了解杀毒软件如何设置启发式阈值的：

阈值级别	相应的启发式级别
最高	仅严格(或近严格)检查；启发式不采用或保持在最低水平。
普通	已知病毒检测采用算法扫描并适时使用严格(或近严格)级别的仿真分析。可能使用广谱特征来识别近似变种。
启发式模式	中级启发水平，增强检测水平；误报很少，采用被动分析多于采用基于仿真的启发式。
最低	最高级(高强度或最大灵敏度)启发式，包括某种形式的模拟运行。可以检测出高比例的新病毒，但是误报率也随之增加。

不是所有的反病毒引擎都按上述级别进行灵敏度的划分，很多也不允许用户手动设置或重新配置这些阈值，即使那些支持灵敏度调节的杀毒软件也未必会以帮助文件的形式详述其用途。但需要强调的是，在上述阈值区间处处都可能应用了某种形式的模拟。

杀毒软件厂商在缺省设置中关闭高级启发式，不仅仅是为了降低减少误报的风险，实际上更是想提高产品的检测速度。所有级别的启发式分析都会增加扫描的处理时间，而且对于某些产品来说因此导致的性能下降则更加明显。可是，已如我们前述，尽管已知恶意软件的数量在增长，得益于在今天高性能计算机上程序的改良设计，这种影响可以降至一个可接受的水平。事实上，不同厂商的扫描器对运行速度的负面影响有着很大的差别。—个良好的启发式引擎应该把对系统性能的影响降到最小。

启发式的灵敏度不止是一个涉及如何精确诊断未知病毒技术问题。它还是一个社会心理学问题：对于一个可能的病毒，我们如何警示用户并提出指导性的建议？

可能出现的报警提示可以让消费者充分地了解厂商。一些产品很谨慎，它们使用这样不完全确定的警示“这可能是某病毒的变种”，这很有效。像这样把最终的决定权交给用户，可以消除厂商误报的风险。

事实上，大部分用户都更愿意让扫描器作出判断。想到杀毒软件可能出错，用户会感到不安，这让人觉得这种技术不是那么的可靠。

一些厂商的产品显示一些醒目的警示消息，如“检测并阻止了某某恶意软件”或者“检测并移除了W32/nastybackdoortrojan”。这看起来很好，用户会很感谢恶意软件被识别并且清除了。但是用户却不知道，这些名称只是启发式检测到疑似恶意软件时所用的简单广谱名，而并不表示这是某个特定的病毒。

不幸的是，没有可靠的统计数字可以说明，究竟有多少正常程序和邮件被某些过于自信的扫描器误报为恶意软件。

一些厂商建议，只在最可能出现新的恶意软件的环境下开启高级启发式，如邮件网关扫描器等。这会降低桌面端的误报率，但参数扫描一旦失败，也会加大漏报几率。

所有级别的启发式分析都会增加扫描的处理时间，而且对于某些产品来说因此导致的性能下降则更加明显

测试的争议

如何测试杀毒软件的检测能力一直存在着争议[14]，目前只有极少数测试者和测试机构能够得到该领域其他反病毒研究机构的认可。

被广泛认可的具备测试资格的机构有：

AV Comparatives (<http://www.av-comparatives.org/>)

AV-Test.org (<http://www.av-test.org/>)

ICSA Labs (<http://www.icsalabs.com/>)

SC Magazine/West Coast Labs (<http://www.westcoastlabs.org/>)

Virus Bulletin (<http://www.virusbtn.com/>)

Virus Research Unit, University of Tampere (<http://www.uta.fi/laitokset/virus>)

Virus Test Center, University of Hamburg
(<http://agn-www.informatik.uni-hamburg.de/vtc/naveng.htm>.)

(注意：近来，最后两个测试机构不是很活跃。)

目前只有极少数测试者和测试机构能够得到该领域其他反病毒研究机构的认可

不像测试者与反病毒研究机构没有任何联系，这些机构得到了反病毒研究机构的普遍信任（虽然未必都是该机构中的成员），被认为在有能力、安全、合乎职业道德的前提下，严守中立地对杀毒软件的检测能力进行测试。这种被信任的身份，使得他们通常可以得到经过正式鉴定的病毒样本，例如由 WildList 国际组织（<http://www.wildlist.org/>，一个与绝大多数主要反病毒厂商的研究人员、许多大公司及教育机构之间都有合作的组织）收集、测试和正式鉴定的病毒样本。

反病毒团体认为，绝大多数由行业认可的测试机构以外的测试是无效的，或者换句话说是不适宜的。原因是：无法判定其测试能力，因此，也就不能判定

——其测试方法是否适当

——其是否遵守安全技术规章，行业的道德规章和标准

由于存在这些问题，反病毒研究机构的成员们出于职业道德层面的考虑，是不能将病毒样本与未经认可的测试者共享的。因此，测试者是不能假定测试样本的来源及其可靠性的。通常，那些无权使用反病毒机构样本库的测试者，会通过病毒交流网站和其它途径（可能有争议的）来获得样本。通过这些途径获得的样本可能包含各种各样的非病毒样本（如垃圾文件，未遂病毒，尸文件等等）。如果评论媒体委托一个认可的机构来进行测试，一些问题就能够得到解决。（例如，AV-Test 为杂志评论专栏所做的几种类型的测试。）

奇怪的是，这些困难达成了（但不是引起）这样一种状况：关注杀软检测未知病毒能力的测试者，早在启发式技术这一名称正式出现并拥有“21 世纪的能力”之前，就已经通过制造变种的方式，对其进行测试了。遗憾的是，这一般都使用了不可靠的病毒生成器和不恰当的病毒模拟器，而且随意的放置或掩盖病毒代码和文本串等等[15]。

当然，测试一款杀毒软件的启发式能力是一项很有意义的工作（特别是现在扫描器具有了启发式检

测能力)。但是合理安全地进行这种测试，与检测已知病毒有着同样重要的意义。在缺乏妥善管理的测试病毒库的情况下，不能保证用于测试的都是有效的，具有活性的病毒。那些由于缺少与反病毒研究机构的直接接触，其能力受到质疑的测试者，如果不公开他们的测试方法，尤其是样本的有效性，会给他们自己以及依赖于其测试结果的人们带来更多的误解。

在缺乏妥善管理的测试病毒库的情况下，不能保证用于测试的都是有效的，具有活性的病毒

我们认为有效性是指被测试代码是否的确是恶意的——也就是说病毒必须具备自我复制的能力，蠕虫必须能以某种方式传播等等。通常，如果测试者在测试时没有遵循这些有效性，很多代码就会在事后被证实并不是恶意的，而是错误地使用了一些遭到破坏的或本身就合法的文件进行测试。

在最近的一个例子中[16]，有人间接性地建议委托机构，使用病毒生成机完成测试。这直接导致反病毒研究者们怀疑测试者的能力，因为众所周知，病毒生成器在产生活动病毒时是极其不可靠的。因为没有描述详细的测试方法，所以也就不知道他们是否验证，以及如何验证了所使用样本的有效性。

病毒测试样本中，如果含有一些或全部的非病毒文件，就会使测试无效。既如此，那么最高的病毒检测率未必等于最佳的性能，因为即使被测试的杀毒软件都采用了一致设置的情况下，也产生了大量的误报[15]。

反病毒行业不愿宽恕那些制造新的恶意软件或病毒代码的行为，即使仅仅是为了测试。这有很多原因：大多数研究者坚持严格的道德规范，担心新的病毒落入缺乏经验的测试者手中造成安全问题，有效性验证方面的困难等等。尽管如此，测试扫描器的启发式能力，实际上是不必制造病毒的。

“回溯测试”就是用经过验证的恶意软件（这些恶意软件是在被测试的扫描器最后一次更新后才出现的）对一段时间（一般选择三个月）没有更新的扫描器进行测试。这就合理的保证了测试的是启发式能力，而非特征码算法检测已知病毒的能力。这样的测试，在没有减少有效性需求的基础上，避免了为测试而制造新病毒，带来的道德层面上和实践过程中的困难。然而，这并不意味着不需验证病毒样本和认真进行有意义的测试。

几乎所有的主流反病毒厂商，每天（或者更频繁地）都提供病毒库的更新，所以测试一个过期三个月的扫描器，并不能说明它当前的病毒检测能力。一个更有效的方法可能是采用不同的时间表测试其能力，或者是抽取一个病毒测试各杀软首次检测到的时间。显然，扫描器在恶意软件被纳入病毒库之前就能够检测到是很值得注意的。

结论：启发式技术的尴尬

有趣的是，虽然当今的启发式技术与 90 年代相比要成熟得多，整体查杀率却有了大幅下降，只是对一些老牌的恶意病毒（如宏病毒、邮件群发器等）还保持着相对较高的查杀率。

有时候，这种整体性的退步，看上去是由反病毒软件领域的低迷和依赖于病毒特性的查杀模式造成的，其实事实并非如此。造成退步一个很重要的因素是由于恶意软件的编写者们技术的进步，他们已经总结出了一套最大限度躲避启发式查杀的方法，并且针对定期升级和优化设置后的杀毒软件做有效性测试。这个问题要比几年前棘手得多，因为在当时，如果某个杀毒软件除了病毒还能查出其他的恶意软件，至少对于生产厂商来说，已经是意外收获了。

恶意软件作者已经总结出了一套最大限度躲避启发式查杀的方法，并且针对定期升级和优化设置后的杀毒软件做有效性测试

而在当下，病毒（指具有可识别复制功能的程序）只在恶意软件中占有很小很小的一部分[17]，这在某种程度上大大增加了启发式扫描的难度；虽然说要查出某个可复制的程序在技术上并不一定总是可行，但如果能通过解读代码来确定某程序试图复制的话，那用启发式方法查杀病毒在概念上就要容易些。要自动确认某个程序是个远程操控，还是某种类型的木马，或者只是判断是否具有恶意，这绝非易事。[5]

举个简单例子：某个格式化磁盘的程序不会被定义为恶意软件，事实上，它可能只有这显而易见的唯一功能；但是，如果电脑使用者被误导认为该程序能够播放影片或者能够改善网络数据传输而使用它的话，它当然就是恶意的。此类案例的实质性问题就在于，要建立一种算法，能够基于使用者对程序的目的和设计者的意图的理解来进行判别，而不是程序本身的特性。

虽然我们无法建立可靠的针对恶意软件的启发式算法，但是我们可以应用其他的启发式算法并针对某个程序给出分值。与已知的恶意软件极为相似的（程序）得分可能会更高一些。还有很多其他的行为会触发预警提示，这是根据应用环境决定的，比如打开了 SMTP（邮件传输协议）或 IRC（在线聊天系统）通道，或者启用了文件传输机制等等。对可执行文件的分析可以发现许多的异常的编译代码，例如可疑的补丁和特征位组合，不一致的文件头特征，及文件大小的不匹配等等。更为广泛的计算机环境找到恶意软件的几率会较大，因此也能提供更多有价值的关于恶意软件特征的线索。通讯信息分析不仅可以查出已知的邮件群发以及通过邮件传播的木马，有时甚至可能包含有诸如加密文档密码等的有价值的信息。

虽然很多病毒扫描器都具备类似功能，但就目前情况乐观估计，启发式扫描器有可能在将来通过自动扫描来获取复杂口令，这对于包含有较高比例图像内容的信息尤为适用。如果某个信息与其他恶意信息有相似性的话，获取这类复杂口令的成功几率会更高。带有恶意软件或网址的信息也可能与其他的诸如钓鱼或垃圾邮件的恶意信息传输相类似，因为病毒作者或垃圾邮件的发送者近几年一直在相互借鉴对方的技术，有证据表明，这些原来各自为战的组织已经有了越来越多的共同利益。人们通常会使用邮件扫描来对各种滥发邮件及纯粹的恶意软件进行检查，数据流量分析也会显示与恶意攻击相关的模式，例如海量垃圾邮件，以远程控制为目的的恶意邮件、欺骗程序等等。鉴于此，(利用启发式或其他技术)对垃圾邮件进行网关扫描，也会对恶意软件的查杀提供很大帮助。

虽然用户和厂商一样，都希望主动防御能够做得像当年启发扫描那样有如此之高的查杀率，但这绝不意味着肯定能够成功。恶意软件作者有着不同的优先级。原来那种追求最快变种传播速度的霰弹枪式的方法已经被摒弃了，现在他们的方式集中在了可能以特定个体或集体为目标的，单个作用期短但高频发的恶意软件上。即使是简单的改动，如一系列短平快的针对壳特征的修改，都会使程序特征发生变化，从而降低被查杀的几率(启发式或非启发式)，并加大对所有反病毒实验室的资源占用。植入电脑的恶意软件，一旦运用远程控制技术，定期进行自动升级和自动修改，将很难被检测出来。

这些问题伴随着我们的生活已经好几年了，所以没有必要恐慌。有了基本的电脑清洁意识、良好的补丁习惯以及反恶意软件的适时更新，您的计算机就会继续得到良好的保护。除此之外，虚拟和模拟技术的日益发展成熟，配合启发性分析，构成安全软件厂商强有力，不断进取，坚不可摧的甲冑。然而，无论是反病毒软件厂商还是其他热门替代技术的追随者，都不能断言可以预测到将来所有的威胁。

所以最好让你的期望现实一些。

参考文献

- [1] "A Short Course on Computer Viruses 2nd Edition", pp 2, 49 (Dr Frederick B Cohen): Wiley, 1994.
- [2] "VIRUS-L/comp.virus Frequently Asked Questions (FAQ) v2.00" (N. FitzGerald et al., 1995):
<http://www.faqs.org/faqs/computer-virus/faq/> (Date of access 12th January 2007)
- [3] "Analysis and Maintenance of a Clean Virus Library" (Dr. V. Bontchev):
<http://www.people.frisk-software.com/~bontchev/papers/virlib.html> (Date of access 12th January 2007)
- [4] "The Anti-Virus or Anti-Malware Test File": http://www.eicar.org/anti_virus_test_ile.htm
- [5] "Trojans" (Harley), in "Maximum Security 4th Edition" (ed. Anonymous): SAMS, 2003
- [6] Oxford Compact English Dictionary, Oxford University Press:
<http://www.askoxford.com/> (Date of access 12th January 2007)
- [7] Merriam-Webster Online: <http://www.m-w.com/> (Date of access 12th January 2007)
- [8] "Viruses Revealed" (Harley, Slade, Gattiker) pp158-159: Osbome 2001
- [9] "Evolution Discussion Group Fall 1996 Phylogenies and Evolution, Useful Terms" - University of British Columbia Zoology Department: www.bcu.ubc.ca/~otto/EvolDisc/Glossary.html (Date of access 12th January 2007)
- [10] "Virus Proof" (P. Schmauder), page 187: Prima Tech (2000)
- [11] Dr. Solomon's Virus Encyclopaedia (Solomon, Gryaznov), pp30-31: S&S International (1995).
- [12] The Art of Computer Virus Research and Defense (Szor), page 441, pp451-466: Addison-Wesley (2005).
- [13] "Heuristic Programming":
http://www.webopedia.com/TERM/h/heuristic_programming.html (Date of access 12th January 2007)
- [14] Anti-virus programs: testing and evaluation (Lee): in "The AVIEN Guide to Malware Defense in the Enterprise" (Ed. Harley): Syngress (2007, in preparation).
- [15] "AV Testing SANS Virus Creation" (Harley): Virus Bulletin pp6-7, October 2006
- [16] "Consumer Reports Creating Viruses?" (Sullivan):
http://redtape.msnbc.com/2006/08/consumer_report.html (Date of access 12th January 2006).
- [17] "Email Threats and Vulnerabilities" (Harley). In "The Handbook of Computer Networks" (Ed. Bidgoli): Wiley (2007 – in press).

术语表

广告软件	通过某些手段（例如弹出一个窗口或打开某个网站）来吸引用户，发布广告或宣传商品的软件程序。如果未在用户许可或知晓的状态下安装，通常被视为木马程序。
半精确识别	识别方法在应用过程中，只能保证不会因为使用了不当的清除方法，造成目标文件的损坏。病毒体的不变部分都不被单独地定义。
校验值	文中所提及的校验值是指依据某一个具体文件的信息内容而得出的计算值。文件的内容改变，校验值也将随之改变。（某些校验值算法易产生碰撞，也就是某个文件被误算出与另一个文件有着相同的校验值，但在绝大多数的情况下对于一个单独的文件，文件的改动将影响计算出的校验值——这已经足以用于对绝大多数的完整性、正确性校验。）
腐败病毒	因改动或功能减弱，或因失去活性而受到损坏的恶意软件（文中特指病毒）。
分布式拒绝服务攻击	从特性上看，远程攻击者使用恶意安装在计算机网络上的僵尸进程或代理软件，对其他功能不完善的系统发动的攻击。
破坏性木马	相对于某些破坏性较弱的，盗取密码或其他数据的木马来说，通常是指蓄意的，能引起直接性破坏的木马软件。
释放器	这种程序通常指本身不是病毒，但安装其它蠕虫或病毒等恶意软件。
EICAR 测试文件	固定格式的程序文件，对绝大多数的反病毒软件来说是一个测试程序，将对它做出与对病毒极为相似的反应。EICAR 文件不是病毒，不带有恶意威胁：如果执行，它会简单地在屏幕上显示，证实其本身是只是一个测试文件。
精确识别	当病毒体恒定部分的每一段被唯一识别后，将其鉴别为病毒的一种技术。
漏报	指反恶意软件未能检测出真正的恶意软件的情形。
误报	指反恶意软件错误地把正常文件当作恶意软件检测出来的情形。

垃圾文件	在反病毒研究中，这种文件不是一种恶意的程序，但却被当做恶意软件，收集在管理不善的恶意软件样本库中。
广谱	安全程序不对具体的威胁进行识别，而是用阻止一整类威胁的方法进行防护。广谱特征码是其中的特例；所有同类变种都使用相同的一个特征码识别，而不是对每个变种使用单独的特征码。
病毒原体	未繁殖的病毒，还没有感染任何文件（比如一个仅有病毒编码组成的文件，而不是一个具有感染性的二进制文件）。
启发式检测/扫描	当检测对象表现出足够的病毒或恶意软件特征时，提示它可能是病毒或其它恶意软件的技术。
未遂病毒	由于某些原因而不能运行的病毒文件（或是其它恶意软件，较少），通常是因为病毒作者未进行足够的测试。
玩笑程序	做出一些意料不到的令人反感行为的程序，但不会产生实际性的损害。玩笑和木马之间的界限，有时很微妙。
键盘记录器	监听键盘按键的程序，通常是被恶意安装，从事非法目的，如密码偷取等。
已知病毒扫描，病毒特征扫描	扫描已知病毒，并在扫描环境中识别发现的病毒名称。
否定式启发	一种检测规则或标准，如果检测对象符合标准，则不是病毒或恶意软件的可能性减小。
口令	相对于简单的单词或字符串密码，口令通常是一组更长的字符，作为更加安全的密码形式应用。
肯定式启发	一种检测规则或标准，如果检测对象符合标准，则是病毒或恶意软件的可能性加大。
回溯测试	一种测试扫描器启发式能力的方法，即停止更新病毒库一段时间后，用扫描器扫描在这段时间内新出现的恶意软件

内核后门	一个或一组隐密安装的程序，用来对系统进行未经许可，具有高级权限的访问。有时也用 stealthkit 这个术语指代，这时可以指未经许可，但不具备高级权限的访问。
扫描字符串，检索字符串	已知病毒中的一组字节，在合法程序中不应存在。这个术语不仅仅限于指代静态搜索字符串，也可能包括各种通配符和通用表达，或者另一种病毒检测算法。有时也称为“扫描特征码”。
自启动	这个术语用于描述不需要借助于受害体任何动作，就能传播和触发的恶意软件。
特征码	与“检测字符串”同义。可用于说明静态的字符串检索，但最好避免使用这个术语，特别是它会让人产生所有病毒扫描器都使用相同字节序列，来识别某个病毒或变种的误解。
间谍软件	一种程序，用来秘密收集计算机用户的信息并传送给感兴趣的一方。其中包括一些类型的广告软件。
病毒生成机	本身不是病毒，但能生成病毒程序。也被我们称作“病毒制造器”。
病毒特征检测	通过对病毒及其变种的一系列特征的字符串检索，查杀已知病毒的方法。
通配符	被用以替代其它字符或者连续字符串的字符，也应用于特殊格式的规则表达。
僵尸程序	一种后门程序，电脑被感染后，等待并遵照来自远程计算机或者被感染机器本身的一系列操控指令行事。