doi 10.3969/j.issn.1671-1122.2013.04.003

基于特征码病毒扫描技术的研究

关欣[1]，朱冰[1]，陈震[1]，彭雪海[2]
（1. 清华大学计算机系，北京 100084；2. 北京经信委社会信息化处，北京 100029）

摘 要：随着互联网的快速发展，病毒以极其迅猛的速度大量出现并蔓延。病毒总数以爆炸性的速度增长。因此，增强对病毒的防范，加强对反病毒技术的研究，成为了当务之急。文章研究病毒扫描技术，重点研究了利用病毒的特征码进行病毒扫描的技术，尤其是基于十六进制特征码和 MD5 特征码的扫描算法。对基于十六进制特征码的算法，改进了其原来的二叉树结构，提出新的 Hash 表结构，从而加快了处理速度。然后，提出采用 Hash 表结构的基于 MD5 特征码的算法，对其性能进行了测试，并展望了其发展前景。

关键词：反病毒；特征码；二叉树；Hash 表；MD5
中图分类号：TP393.08 文献标识码：A 文章编号：1671-1122（2013）04-0008-06

Study on Virus Signatures based on Matching in Virus-detecting Technologies

GUAN Xin[1], ZHU Bing[1], CHEN Zhen[1], PENG Xue-hai[2]
(1.*Tsinghua University, Beijing* 100084,*China;*
2.*Beijing Economic and Information Committee of Social Informatization, Beijing* 100029*, China*)

Abstract: As the computing technology flourishes recently, the computing world is also in the peril of viruses. Network transferred viruses, such as worms, which threat millions of hosts in Internet. As countermeasure of such attacks, the research work for Anti-Virus is urgent affairs and a big problem. This paper describes an implementation of Anti-Virus engine. It focuses on technologies for scanning viruses by the use of virus signatures based on Hex Sig and MD5 Sig. The author implements two Hex-Sig-based scanning algorithms by using binary tree and hash table, and compares the performances of the two algorithms. Some conceived experiments are conducted, and results show that the hash-table-based algorithm achieves better performance in terms of the scanning speed. The author also investigates into a MD5-Sig-based scanning algorithm, and evaluates its performance. Finally, future work is planned and prospected.

Key words: anti-virus; signature; binary tree; Hash table; MD5

1 背景介绍

随着互联网的快速发展，病毒以极其迅猛的速度大量出现并蔓延，其总数以爆炸性的速度增长。病毒大面积的爆发，给全球互联网带来了灾难，使用户上网面临了威胁。因此，增强对病毒的防范，加强对反病毒技术的研究，成为了当务之急[1]。

近年来，病毒的发展总体上有以下特征：首先，互联网成为其传播的主要载体[1]，网络的迅速发展使网站浏览、邮件、即时通讯、电子商务、网上银行等与网络有关的服务得到全面的发展[14]，这些情况使得病毒的攻击更加有针对性[1]。其次，木马类间谍软件发展迅速，这类病毒一般以盗取隐私和网上财富、谋取经济利益为目的[2]。再次，许多病毒变种频繁，尤其是各种蠕虫病毒，为躲避杀毒软件的查杀，频繁变种，造成了识别与防治工作的困难。最后，病毒开始向多元化发展，例如许多针对 QQ 和 MSN 等即时通讯工具的病毒，针对某网络游戏的病毒，针对手机的病毒，还有 Unix 系统中的大量病毒[5]，等等。

在病毒发展猖獗的情况下，反病毒技术相应也在迅速发展。从当前的杀毒技术上来讲，反病毒软件的核心部分都是一个扫描器[6, 7]。基于特征码的扫描方法是当前最主要的查杀病毒方式，它利用“特征码”通过检查文件、扇区和系统内存来查杀已知病毒[3]。

同时，新的反病毒技术也正在研究、发展之中。由于 Internet 的普及，互联网已经成为病毒制作技术扩散、病毒传播的重要途径，

收稿时间：2013-01-05
基金项目：国家 973 项目 [2012CB315800]、国家自然科学 A3 重点基金项目 [61161140320]
作者简介：关欣（1982-），女，辽宁，硕士研究生，主要研究方向：网络信息安全；朱冰（1984-），男，河北，硕士研究生，主要研究方向：网络信息安全；陈震（1976-），男，浙江，副研究员，博士，主要研究方向：网络信息安全；彭雪海（1975-），男，处长，博士，主要研究方向：网络安全等。

病毒开发者之间已经出现了团队合作的趋势，病毒制作技术也在与黑客技术进行融合。这对现在的反病毒技术提出了挑战，反病毒技术正在发生转变[9]，概括而言：反病毒的理念正在从作品对抗向思想对抗方向转变。

之前的反病毒技术，只能在病毒出现之后再进行防范，对未知病毒几乎没有防范能力[8]。而新的理论是基于对大量的病毒的特征、发作过程、传播变化统计的基础上，建立控制策略数学模型，采取分门别类的方法，有效解决应用同种思想开发出的各种病毒，可以极大提高对新病毒的反应时间[9]。由于这种方法是通过抑制病毒设计思想而实现的，因此，这是一种病毒制造者与安全专家之间在整体思想层面的博弈。具体一点讲，这些新型技术包括启发式扫描、行为判断等。行为判断就是通过驻留的杀毒软件截获那些对用户有病毒危险的行为，优点在于可以在病毒感染的早期发现并阻止。另外，还有利用虚拟硬件对未知病毒进行识别与清除的技术，其核心是以软件的形式设计虚拟CPU，如QEMU技术，然后将可疑文件放入这个虚拟的CPU进行解释执行，在执行的过程中对该可疑文件进行病毒的分析、判定。针对网络病毒，利用虚拟网络与虚拟主机技术，观察病毒在虚拟化环境中的执行，对其行为进行辨识和分析，如GQ[17]。这些理论的许多工作还在研究之中，虽然取得了一些进展，但还远未成熟，并未完全进入实用阶段，短时间内还不能期望它们成为主流的反病毒技术。

总体上来看，基于特征码的扫描方法还是目前最主要的查毒方法，如安全网关的网络查毒等[18-23]。因此，提高基于特征码的扫描方法的扫描速度具有重要意义。为提高扫描速度，本文从两方面考虑：一是利用不同类型的特征码进行扫描；二是改进扫描算法中特征码的组织结构。具体来讲，对基于十六进制特征码的算法，改进了其原来的二叉树结构，提出新的Hash表结构，从而加快了处理速度。然后，提出新的采用Hash表结构的基于MD5特征码的算法，对其性能进行了测试，并展望了其发展前景。

本文将首先介绍基于特征码的病毒扫描技术的原理，并分别介绍十六进制特征码和MD5特征码这两种不同的特征码及其扫描原理；然后，介绍基于十六进制特征码的算法，首先是采用二叉树结构的算法，然后详细介绍作者提出的采用Hash表结构的算法，并对这两种算法的性能进行对比测试及分析；接下来，介绍作者提出的采用Hash表结构的基于MD5特征码的算法；最后是结论部分，对全文作了总结。

2 基于特征码的病毒扫描技术

前面已经提到，最直接、最常用也是最有效的扫描方法就是利用已知病毒的特征码对病毒进行扫描和检测[10]。病毒的“特征码”就是病毒的源代码或可执行文件中的某段特征。利用该方法对某病毒进行检测之前，首先需要得到该病毒的特征码；然后对数据进行扫描，使数据与该病毒的特征码进行匹配[15]。如果匹配成功，就认为数据带有该种病毒，反之则认为数据是安全的；收集类似的大量特征码，组成的集合即为病毒数据库[16]；待扫描的数据与该数据库内的特征码依次进行匹配，就可以检测出是否带有病毒以及带有何种病毒[4]。

从原理上区分，病毒的特征码可以分为这样两种：十六进制特征码（Hexadecimal 特征码，以下简称为Hex Sig）和MD5特征码（MD5 特征码，以下简称为MD5 Sig）[11, 13]。

Hex Sig是比较常见也比较常用的一种特征码。其原理如图1所示。它是指从病毒文件（可以是可执行文件、源代码等任意确定含有恶意代码的文件）中截取的一段特征码，以十六进制（或者说作为二进制的数据）的格式保存。进行病毒检测的时候，把Hex Sig与等待检测的数据进行匹配，如果匹配成功，也就是说扫描的数据中有相同的一段十六进制的数据，则说明该段数据带有这种Hex Sig代表的病毒。

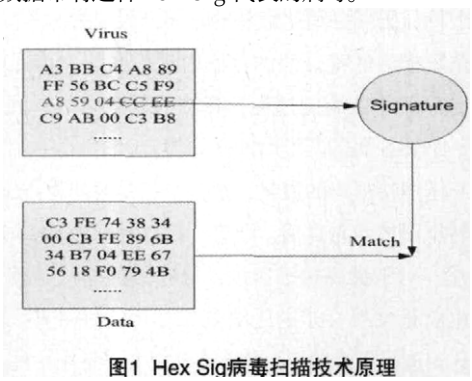

图1 Hex Sig病毒扫描技术原理

MD5 Sig是另外一种特征码，目前还比较少见，应用也比较少。其原理如图2所示[13]。通过MD5算法，可以计算某个病毒文件（例如病毒的源代码、可执行文件等）的MD5摘要（MD5 digest），该摘要就作为这种病毒的特征码，称为MD5特征码。进行病毒检测的时候，对等待检测的某段数据（或者说是某个文件），首先计算其MD5摘要。然后与已知的病毒MD5摘要进行匹配，如果匹配成功，也就是说和已知病毒的MD5摘要相同，说明数据带有病毒。

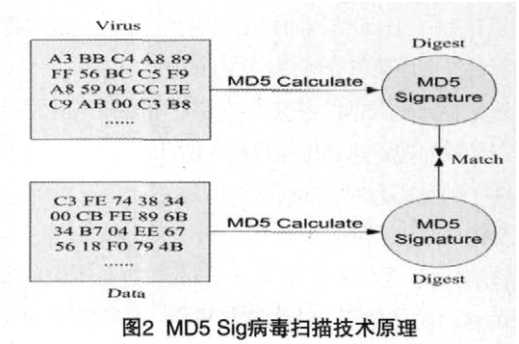

图2 MD5 Sig病毒扫描技术原理

下文针对这两类不同的特征码，分别进行病毒扫描算法的研究。

3 基于 Hex Sig 的扫描算法

3.1 算法思想

对于利用 Hex Sig 特征码进行病毒扫描的算法来说，其核心部分就是数据与病毒数据库中特征码的匹配。与任何一种特征码匹配成功都说明数据带有病毒。只有与所有的特征码都匹配失败，数据才能通过检测。因此，数据库中病毒的数量是扫描速度的瓶颈所在。目前已知病毒的数量一般在几万左右，著名的杀毒软件 Norton AntiVirus 最新的病毒库的容量为 7 万左右[12]。如果把这些特征码逐个与等待检测的数据进行匹配，将耗费大量的时间，会极大地限制扫描速度。

为了提高扫描的速度，本文把扫描过程划分为两个阶段来进行：

(1) 压缩需要进行匹配的特征码的数量。假设特征码的总数为 n。再假设通过病毒数据库的组织和算法组织，可以把需要进行匹配的特征码的数量减少到 k。从 n 个特征码中压缩出这 k 个特征码需要消耗额外的时间，假定为 t_s。

(2) 在压缩后的特征码的范围内进行匹配。该匹配过程实际上就是字符串匹配的过程，算法的复杂度基本固定。假定数据与一个特征码的匹配时间平均为 t_m。

如果不压缩特征码的数量，则整个扫描时间为 $n\times t_m$

压缩特征码的数量之后，则整个扫描时间为 $t_s+k\times t_m$ ：

很显然，对于提高扫描速度，最有效的办法是使 k 尽可能的小，也就是使第一步的压缩效果尽可能好一些。同时压缩过程的代价应尽可能的小，也就是要使 t_s 尽量小一些，这样，将使算法得到最好的扫描速度。

压缩特征码数据库的关键是合理组织数据库的结构。可以根据一些数据结构上的技巧或者先验知识等首先剔除掉那些不可能与数据匹配成功的特征码，然后定位到那些剩余的还有可能匹配成功的特征码。压缩特征码数据库的基础也就是对特征码数据库进行适当的组织，数据库的组织是算法效果的关键。

3.2 基于Binary Tree结构的扫描算法

基于 Binary Tree 结构的扫描算法是目前一些反病毒软件应用的算法，下面简单分析一下它的原理。

算法把特征码数据库组织成二叉树(Binary Tree) 的结构，通过对二叉树的搜索达到压缩数据库的目的。

以 3 个特征码为例 ：001xxx、100xxx、110xxx，把它们组织成二叉树的结构，见图 3。

在该结构中，特征码根据前 n 个 bit 位进行组织。建立该二叉树的算法简述如下 ：从根节点开始，如果特征码的对应 bit 位是 0，则向左进行插入 ；如果特征码的对应 bit 位是 1，则向右进行插入。如此进行递归插入，直到遇到叶结点，则把叶结点分裂为两个叶结点。这样建立好的二叉树结构，所有的特征码都存放在叶结点。

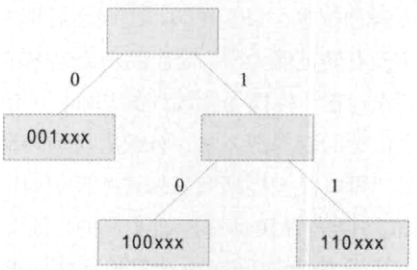

图3 二叉树结构的特征码数据库

对数据进行扫描时，只要根据数据的前 n 个 bit 位，沿二叉树搜索下去，就可以找到唯一有可能与数据匹配成功的特征码。这是该算法最大的好处——得到了最好的压缩结果，无论原来特征码的数量是多少，压缩结果都是 1。

二叉树算法的两个关键的性能指标是 ：

1) 把特征码的数量从 n 压缩到 k 消耗的时间 ：当数据库的规模为 n 时，平均搜索次数大约为 $\log_2 n$，该时间代价为 $O(\log_2 n)$ ；

2) 压缩的结果 ：把特征码的数量从 n 压缩到了 1。

二叉树算法的空间浪费 ：除去特征码占据的必要的存储空间，非叶节点造成了空间上的浪费。若特征码数量为 n，那么，非叶节点的数目为 n-1，也就是说浪费的空间为 $O(n)$。

3.3 基于Hash Table结构的扫描算法

为提高扫描速度，作者提出了基于 Hash Table 结构的扫描算法。

该算法的特点是 ：把特征码数据库组织成 Hash 表 (Hash Table) 的结构，把对整个数据库中的特征码的处理转化为对一个桶 (Bucket) 内的特征码的处理，从而达到压缩数据库的目的。

具体来讲，首先设计一个 Hash 表，将特征码通过 Hash 函数 (Hash Function) 映射到不同的桶内，如图 4 所示。这样，当对数据进行扫描时，首先将应用同样的 Hash 函数对数据进行映射，然后只需要与对应的桶内的特征码进行匹配就可以了，从而达到了压缩特征码数据库的目的。

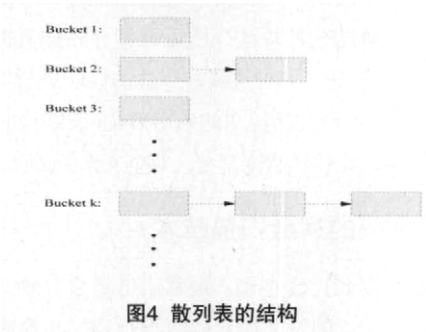

图4 散列表的结构

Hash 表都会涉及到冲突的问题。在最好的情况下，可以没有冲突发生，这时每个桶内最多有一个特征码，压缩效果最好，把特征码的数量从 n 压缩到 1。如果冲突发生太多，会导致某些桶太长，这样压缩的效果会很差。

完全消除 Hash 表的冲突是不太可能的，但是可以尽量减少冲突的发生，从而得到最好的压缩效果。减少冲突的最关键的一点是 Hash 函数的选择。合理的 Hash 函数可以使数据尽量分散到各个桶，从而减少冲突的发生。同时，选择 Hash 函数另外一个重要的标准是 Hash 函数的计算复杂度。如果 Hash 函数本身计算太过复杂，将花费大量的计算时间，失去了散列表查找方便、查找代价小的意义。

综合考虑，最后选定 Hash 函数为简单的线性函数：

$$Hash(x,y,z)=a\times x+b\times y+c\times z$$

其中 x，y，z 分别是特征码的前三个字符。通过调节 a，b，c 这 3 个参数可以得到不同的 Hash 函数。为减少冲突，a，b，c 一般选择质数可以得到较好的效果。经过对多组质数的测试与比较，最终确定的 Hash 函数为：$Hash(x,y,z)=211x+37y+z$。

由于 $0=Hash(0,0,0)\le Hash(x,y,z)\le Hash(255,255,255)=255(a+b+c)$，并且 0~255($a$+$b$+$c$) 之间的值都可以取到，可以计算出桶的总数为 63496。从计算过程也可以看出，通过调整 a，b，c 这三个参数，可以调整桶的总数，也就是 Hash 表的规模。

为了达到冲突尽量少的目的，桶的数目肯定要多于特征码的数目。从后面的测试可以看出，桶的数目需要达到特征码数目的多倍才能获得良好的效果。

抽取不同数目的特征码，对该 Hash 函数的效果进行测试，结果如图 5 所示。

Signature Quantity	1000	5000	10000	25000
Percentage of "Bucket Capacity=1"	89.2%	84.2%	81.3%	74.2%
Percentage of "Bucket Capacity=1 or 2"	96.8%	94.1%	92.8%	89.8%

图5 Hash函数效果测试

从测试结果看，该 Hash 函数基本可以保证不冲突的桶（容量为 1 的桶）的数目达到非空的桶的总数的 80% 以上。如果考虑不冲突的桶（容量为 1 的桶）的数目加上容量为 2 的桶的数目，基本可以保证不冲突的桶的数目在非空桶总数的 90% 以上。这样的效果是在多次调整 Hash 函数的几个参数后得到的，减少冲突的效果已经很不错。

另外，也可以看出，随着特征码数目的增加，冲突的发生也越来越多。但是冲突增加的趋势并不太快。这样的冲突增长幅度会对最终扫描的性能造成什么样的影响呢？在后面会有详细的分析。

Hash 表算法的两个关键的性能指标是：

1）把特征码的数量从 n 压缩到 k 消耗的时间：仅仅通过 Hash 函数的一次计算，所以复杂度为 $O(1)$。该复杂度不随数据库规模的增加而增加。

2）压缩的结果：在选择适当的 Hash 函数和特征码的数量的情况下，可以认为压缩的结果接近 1，比较理想的情况是趋近 1。

Hash 表算法的空间浪费：除去特征码占据的必要的存储空间，空桶造成了空间上的浪费。从后面的分析可以得知，一般桶的总数是特征码数量 n 的几倍，也就是说浪费的空间为 $O(n)$，与二叉树算法相近，或者说，是同样的数量级。

与二叉树算法相比，Hash 表算法最大的好处是大大加快了压缩的速度。同时，Hash 表算法得到的压缩结果尽可能接近了最好的结果，也就是二叉树算法的压缩结果。综合这两种影响，对于最终扫描的速度，Hash 表算法比二叉树算法有所改进，详细的比较与分析见后文。

3.4 算法性能测试与比较

分别使用完全没有被感染的数据和 20%、40%、60%、80%、100% 被感染的数据对 Hash 表算法和二叉树算法进行测试。测试在 Linux 系统（Redhat 9.0）下进行，硬件系统基于 AMD Athlon XP 2500+。测试结果如图 6 所示。

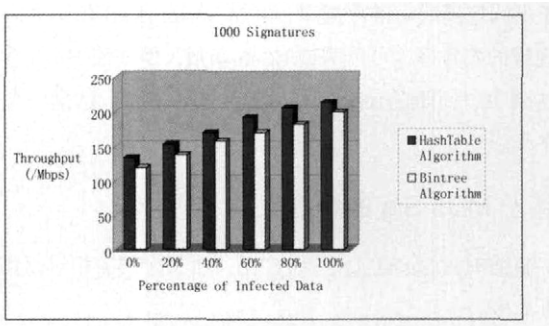

图6 对Hash表算法与二叉树算法的测试结果

Hash 表算法的吞吐量平均比二叉树算法要好 10.8%。改进幅度主要在于 Hash 表算法的压缩代价比较小，仅仅需要进行 Hash 函数的计算。

另外可以看出，随着被感染的数据的增多，吞吐量基本上成线性增长。这是因为扫描数据的过程中，如果匹配成功，也就是发现该数据被感染，就不会再继续向后扫描。极端情况下，扫描 100% 被感染的数据，要比扫描完全没有被感染的数据快 60%~70%。

前文提到过，在 Hash 表算法中，特征码数目的增加会导致 Hash 表冲突的增加，从而影响扫描性能。抽取不同数目的特征码，对 Hash 表算法的性能进行测试，结果如图 7 所示。

以数据库规模为 1000 时的性能为基准，注意观察当特征码数目分别在 1000~10000 和 10000~25000 段空间时曲线的变化趋势。特征码数目从 1000 增长到 10000，特征码数目增

加了9倍，性能从100%下降到了83%，曲线比较平缓，性能下降较慢；从10000增长到25000时，特征码数目增加了1.5倍，性能从83%下降到了26%，曲线比较陡，性能下降很快。

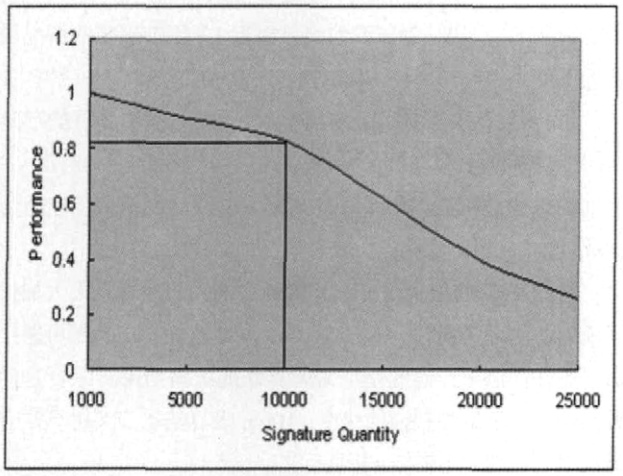

图7 Hash表算法性能测试

因此，对于这样规模的Hash表(6万多个桶)，5000~10000左右的特征码数目是比较合适的，可以达到较好的性能，又不会浪费太多的空间。可以推断，桶的数目需要是特征码数目的10倍左右才能达到较好的性能。根据这个结论，对于不同规模的病毒数据库，例如Norton AntiVirus的7万左右规模的病毒库，可以调整Hash函数，例如使Hash表的规模达到70万个桶左右，就可以在性能和空间上达到一个较好的平衡。

4 基于MD5 Sig的扫描算法

目前Hex Sig的应用十分广泛，而MD5 Sig相对而言应用还不太多。

虽然应用Hex Sig检测病毒是非常普遍的，但是有时候找出一种病毒的Hex Sig却比较困难，因为病毒相关文件中可能没有特别明显的特征。如果勉强取出某段数据作为特征码，有很大可能性会将不带有病毒的文件误认为带有病毒(false positive)的检测结果。

而取得一种病毒的MD5 Sig是很容易的事情，只要有病毒的相关文件，通过MD5算法计算其MD5摘要就可以了。MD5摘要的长度为128bit，已经可以从概率上基本杜绝false positive现象的出现。MD5 Sig的缺点是对病毒的任何轻微的变种都无法识别，因此需要针对病毒变种把新的MD5 Sig加入到数据库中。

通过前面对基于Hex Sig的扫描算法的研究，可以看到，采用Hash表结构对病毒数据库进行组织可以拥有更好的性能。在此基础上，作者提出了采用Hash表结构的基于MD5特征码的算法。

在此算法中，MD5 Sig数据库也被组成Hash表结构。以MD5摘要的第一个字节作为索引，Hash表被组织为256个桶。

具体的算法流程相对简单，分为两个步骤：首先计算数据的MD5摘要；然后把数据的MD5摘要与病毒数据库里存有的MD5 Sig进行匹配，如果与某病毒的MD5摘要相同，那么数据带有该种病毒。其中数据库的组织结构采用Hash表结构，所以其匹配过程与前面的采用Hash表结构的基于Hex Sig的算法相同。

分别使用完全没有被感染的数据和20%、40%、60%、80%、100%被感染的数据对算法进行测试。测试结果如图8所示。

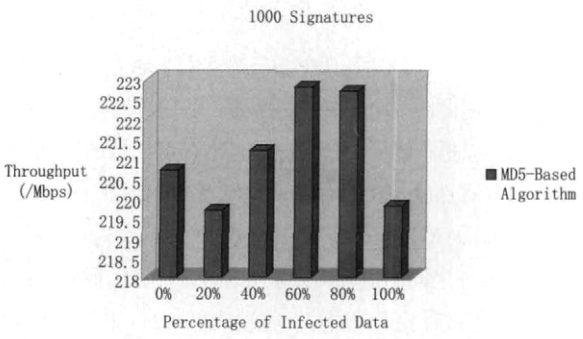

图8 基于MD5 Sig的扫描算法

对完全没有被感染的数据进行测试时，这种基于MD5 Sig的算法的性能大约要比上面基于Hex Sig的两种算法好60%~70%。只有对100%被感染的数据进行测试时，基于Hex Sig的算法才能接近基于MD5的算法的性能。这是因为基于MD5 Sig的算法需要对所有数据进行MD5摘要的计算，随着被感染数据的增加，其计算量并没有减小。所以对不同感染程度的数据，系统性能几乎没有变化。

在基于MD5 Sig的算法中，MD5摘要的计算占据了大部分的处理时间。MD5算法方便用硬件实现，如果利用硬件单元进行MD5摘要的计算，将大大提高其处理速度，从而拉大与基于Hex Sig的算法的性能差距。这是该算法在性能上的最大优势。

5 结束语

本文首先对计算机病毒的发展现状和反病毒的技术背景进行介绍，然后重点介绍了被广泛应用的基于特征码的病毒扫描技术，实现并分析了两种基于不同类型的特征码技术——Hex Sig和MD5 Sig，并分别应用这两种不同的特征码进行了病毒扫描算法的研究。在应用Hex Sig的算法中，为提高处理速度，改进了原有的基于二叉树结构算法，提出了基于Hash表结构的扫描算法，并与原有算法进行了性能比较。然后提

出了采用Hash表结构的基于MD5 Sig的算法，并对其性能进行了测试。

虽然目前已经出现启发式扫描、行为判断等更先进的反病毒技术，但是基于特征码的病毒扫描技术还是最直接、最常用也是最有效的扫描方法。对这类扫描算法的最高要求是扫描的速度要快。本文通过测试比较了基于二叉树结构和Hash表结构来组织数据库的算法，得出基于Hash表结构的算法具有更好性能的结论。然后在测试中验证了基于MD5 Sig的算法比基于Hex Sig的算法的性能优越。如果能够利用硬件单元来实现MD5摘要的计算，则可以期待基于MD5 Sig的算法的产品性能有一个大的飞跃。（责编 杨晨）

参考文献：
[1] Zuo Y, Panda B. Network viruses: their working principles and marriages with hacking programs [J]. IEEE Systems, Man and Cybernetics Society, 2003, June 18-20, 306-307.
[2] Spinellis D. Reliable identification of bounded-length viruses is NP-complete [J]. Information Theory, IEEE Transactions on, 2003, 49(1), 280-284.
[3] F Cohen. Computer viruses: Theory and experiments [J]. Computer Security, 1987, 6(1), 22-35.
[4] P K Singh, A Lakhotia. Analysis and detection of computer viruses and worms: An annotated bibliography [J]. ACM SIGPLAN Notices, 2002, 37(2), 29-35.
[5] T Duff. Experience with viruses on UNIX systems [J]. Computing System, 1989, 2(2), 155-171.
[6] F Cohen. Computational aspects of computer viruses [J]. Computers & Security, 1989, 8(4), 325-344.
[7] C Cowan, P Wagle, C Pu, et al. Buffer overflows: Attacks and defenses for the vulnerability of the decade [J]. Proc. DARPA Information Survivability Conf. and Exposition Hilton Head, 2000, Jan, 119-129.
[8] Hruska J. Virus detection [J]. Security and Detection, 1997, ECOS 97, European Conference, 1997, April 28-30, 128-130.
[9] Edwards J. Next-generation viruses present new challenges [J]. Computer, 2001, 34(5), 16-18.
[10] Subramanya S R, Lakshminarasimhan N. Computer viruses [J]. Potentials, IEEE, 2001, 20(4), 16-19.
[11] Peter Shaohua Deng, Jau-Hwang Wang, Wen-Gong Shieh, et al. Intelligent automatic malicious code signatures extraction [J]. Security Technology, 2003, Proceedings, IEEE 37th Annual 2003 International Carnahan Conference, 2003, Oct 14-16, 600-603.
[12] Symantec Worldwide [Z]. http://www.symantec.com.
[13] ClamAV Project [Z]. http://www.clamav.net.
[14] Jun Li, Reiher P L, Popek G J. Resilient self-organizing overlay networks for security update delivery [J]. Selected Areas in Communications, IEEE Journal, 2004, 22(1), 189-202.
[15] Lakhotia A, Mohammed M. Imposing order on program statements to assist anti-virus scanners [J]. Reverse Engineering, 2004, Proceedings, 11th Working Conference, 2004, Nov 8-12, 161-170.
[16] Buchen-Osmond C. The universal virus database ICTVdB [J]. Computing in Science & Engineering, 2003, 5(3), 16-25.
[17] C. Kreibich, N. Weaver, C. Kanich, W. Cui, and V. Paxson, GQ: Practical Containment for Measuring Modern Malware Systems, Proc. ACM IMC, November 2011.
[18] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters in Giga-Ethernet LAN", ICC'2006.
[19] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters for TCP-IP Content Processing in Giga-Ethernet LAN, IEEE LCN WoNS2005.
[20] Jia Ni, Zhen Chen et al., A Fast Multi-pattern Matching Algorithm for Deep Packet Inspection on a Network Processor, ICPP 2007.
[21] Beipeng Mu, Xinming Chen, Zhen Chen. A Collaborative Network Security Management System in Metropolitan Area Network. Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.
[22] Xinming Chen, Beipeng Mu, Zhen Chen, NetSecu: A Collaborative Network Security Platform for in-network Security. Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.
[23] Xinming Chen, Kailin Ge, Zhen Chen and Jun Li. AC-Suffix-Tree: Buffer Free String Matching on Out-of-Sequence Packets. Proc. of the 7th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), 2011.

资 讯

中国人民公安大学网络安全保卫学院成功举办“首届网络技能大赛”

按照专业培养方案和公安网络安全保卫工作形势需求，，中国人民公安大学网络安全保卫学院自去年7月起，部署并开展了第一届网络技能大赛，并于2013年3月13日至4月3日进行总决赛。遵照程琳校长针对网安执法专业提出的六大技术要求，大赛按不同年级层次和技能类别，分解成“建站、管站、查站、夺站”四个环节，以全面考核学生专业知识和实操技能，为公安一线培养和输出高级专业型、应用型人才作系统培养和专业训练。

以本次技能竞赛为契机，锻炼了队伍，提升了广大参赛学生的实践应变能力；跟踪了网络应用与安全的新动态，把握了公安基层新需求，注重实用、强化实战；达到了紧扣人才培养、促进教学练战一体化、贴近公安实战的竞赛目的。

据本次竞赛组委会徐云峰院长介绍，本次竞赛得到了公安部相关部门的大力支持，竞赛将以每年一届的形式继续主办，条件允许竞赛将会逐步扩大到校际赛、校局赛、校企赛和全国性大赛，着手打造“中国蓝帽大会”国家级精品赛事。（记者 程斌）