文章编号:1009 - 3907(2007)01 - 0059 - 04

一种启发式宏病毒扫描技术

孙 伟[1], 冯 萍[2]

(1. 长春大学 教务处, 吉林 长春 130022; 2. 长春大学 计算机科学技术学院, 吉林 长春 130022)

摘 要:传统病毒扫描采用特征值扫描技术,虽然这种方法能够非常有效地查知已知的病毒,但是对新病毒完全无能无力。启发式病毒扫描从一定意义上讲就是一种病毒预测技术,它针对病毒的普遍特征(自我复制,破坏系统),对未知病毒进行有效地预测。本文主要针对当前存在的危害极大的宏病毒提出一种简单的启发式宏病毒扫描技术方案。

关键词:宏病毒;启发式;虚拟机

中图分类号:TP309.5 文献标识码:B

0 引 言

装上新系统后,相信每一个人所做的第一件事情就是装上防火墙和杀病毒的软件。为什么? 因为现在病毒的肆虐已经达到了无孔不入的地步,只要有电脑的地方就有病毒! 装上防火墙和杀毒软件并不能让我们高枕无忧,它们都需要不断更新。大家都知道,即使最新的防杀软件也是出现在病毒发作之后,就像电影里面总是姗姗来迟的警察一样。

在病毒研究过程中我们发现大多数病毒有一定的公共特征,与普通程序的区别非常明显;并且考虑到如前所说的当今大多数杀毒软件的不足,我们提出一种启发式宏病毒扫描技术,也就是宏病毒预测。

1 宏病毒的传播方式与危害

宏病毒的产生,是利用了一些数据处理系统内置宏命令编程语言的特性而形成的。这些数据处理系统内置宏编程语言的存在使得宏病毒有机可乘,病毒可以把特定的宏命令代码附加在指定文件上,通过文件的打开或关闭来获取控制权,实现宏命令在不同文件之间的共享和传递,从而在未经使用者许可的情况下获取某种控制权,达到传染的目的。目前在可被宏病毒感染的系统中,以微软的 Word、Excel 居多。

以 Word 宏病毒为例,被感染的文档通常无法正常执行保存、打印等操作,情节轻者在用户执行特定操作时弹出一些含有威胁语言的对话框,情节严重者会在满足一定条件的情况下(例如在每个月的一号),修改一些系统文件或者破坏计算机上的数据。由于该病毒能跨越多种平台,并且针对数据文档进行破坏,因此具有极大的危害性,该病毒在公司通过内网相互进行文档传送时,迅速蔓延,往往很快就能使公司的机器全部染上病毒。

2 传统病毒扫描技术(特征值扫描)

用每一种病毒体含有的特定字符串对被检测的对象进行扫描,如果在被检测对象内部发现了某一种特定字节串,就表明发现了该特征值字节串所代表的病毒。国外把这种按搜索法工作的病毒扫描软件叫 SCANNER。

收稿日期:2006 - 12 - 27

作者简介:孙伟(1980 -),男,吉林省白山市人,长春大学教务处助理实验师,主要从事计算机网络技术以及主动式数据库的研究。

使用特征串的扫描法被查病毒软件广泛应用着。当特征串选择得很好时,病毒检测软件让计算机用户使用起来很方便,对病毒了解不多的人也能用它来发现病毒。另外,不用专门软件,用 PCTOOLS 等软件也能用特征串扫描法去检测特定病毒。这种扫描法的缺点也是很明显的。第一是当被扫描的文件很长时,扫描所花时间也越多;第二是不容易选出合适的特征串,例如 SCAN. EXE 时常会发出假警报。第三是新病毒的特征串未加入病毒代码库时,老版本的扫毒程序无法识别出新病毒。第四是怀有恶意的计算机病毒制造者得到代码库后,很容易地改变病毒体内的代码,生成一个新的变种,使扫描程序失去检测它的能力。第五是容易产生误警报,只要在正常程序内带有某种病毒的特征串,即使该代码段已不可能被执行,而只是被杀死的病毒体残余,扫描程序仍会报警。

3 虚拟机查毒

虚拟机,在反病毒界也被称为通用解密器,已经成为反病毒软件中最引人注目的部分,尽管反病毒者对于它的运用还远没有达到一个完美的程度,但虚拟机以其诸如“病毒指令码模拟器”和“Stryker”等多变的名称为反病毒产品的市场销售带来了光明的前景。

设计虚拟机查毒的目的主要是为了摆脱传统病毒扫描对特征码的依赖,虚拟机首先从文件中确定并读取病毒入口处代码,然后以上述工作步骤解释执行病毒。但是这里的解释执行和真正的执行有着很大的区别,因为我们的目标只是发现病毒而不是像 VMware 那样完全模拟执行程序。这里所谓的“虚拟”,并非是创建了什么虚拟环境,而是指染毒文件并没有实际执行,只不过是虚拟机模拟了其真实执行时的效果。根据病毒的两大特征,我们只需要模拟计算出程序的自我复制和对系统破坏方面的结果。

虚拟执行的优点也是很明显的,首先是不可能被病毒觉察到。因为虚拟机将在内部缓冲区中为被虚拟执行代码设立专用的堆栈,所以堆栈检查结果与实际执行无二(不会向堆栈中压入单步和断点中断时的返回地址);其次由于虚拟机自身完成指令的解码和地址的计算,所以能够获取每条指令的执行细节并加以控制;最后,最为关键的一条在于虚拟执行确实做到了“虚拟”执行。系统中不会产生代表被执行者的进程。因为被执行者的寄存器组和堆栈等执行要素均在虚拟机内部实现,因而可以认为它在虚拟机地址空间中执行,所以不会有对系统产生破坏的危险。鉴于虚拟执行法诸多的优点,所以考虑将其运用于病毒扫描上。

4 启发式病毒扫描

结合病毒的两大特性(传染和破坏)和我们在对上千个二进制病毒的分析经验,在虚拟机的帮助下我们设计如下简单启发式病毒扫描:

4.1 只关注关键指令

不可否认,Intel 的 CPU 指令集是不可能被完全模拟,也没有精力对其进行完全理解分析;但更重要的是绝大多数病毒使用的都是常见指令(从我们分析过的病毒得出的总结),而能够区分病毒和普通程序的指令集合更是如此(譬如查找待感染文件的 INT 21H 的功能调用)。对病毒扫描有用的指令总结如下:

文件操作(查找,读写,修改属性,删除);

中断向量表的修改;

内存操作(修改,读取,驻留);

常见的赋值,计算,跳转指令;

磁盘的直接访问(修改 MBR,分区表)。

4.2 针对病毒的两大特性分别分析

①传染(自我复制)

发现文件操作后,考虑如下几种情况:

a. 拷贝指令

找出拷贝的源和目标,“大量”的拷贝当然是值得重点关注;

b. 写指令

找出写的数据来源;

c. 打开,查找文件指令

找出欲打开的文件的特征(比如"*.exe","*.com")。

②破坏(对系统的危害)

这个比较复杂,但是它并不是识别病毒的重要依靠,只是一个辅助性的目标,所以我们不要求能找出很多的破坏行为出来,也就是说只关注一些已经出现过的破坏行为,总结如下:

a. 直接修改磁盘;

b. 删除文件;

c. 重启;

d. 格式化硬盘。

4.3　对上一步分析的结果作出最后的评估

使用的方法就是对不同情况计算权值,然后报告给使用者目前有两种方法可以跟踪、控制病毒的每一步执行。

5　针对宏病毒的启发式扫描

5.1　提取宏

宏只存在于一些特殊的文档中,比如 MicroSoft Word 和 Excel,并且以二进制格式存在,不能直接浏览,只能在了解其文件格式的具体含义之后才能提取到文本格式的宏。

5.2　宏的语法分析

首先,任一个文档里面的宏都可以分为不同的小模块,它们由 Sub,End　Sub 或者 Function,End Function 来区分。因此语法分析的第一件事情就是把一个文档里的宏分为不同部分,每一个部分都是一个独立的模块;其实这个功能也很容易实现,特别是在 Linux 下,可以采用 Perl 的正则表达式来匹配:

'/Function/,/End　*Function/'

'/Sub/,/End　*Sub/'

其次就对每一个模块单独处理。我们只关心这些模块中可能存在的两件事情:拷贝和对系统的破坏(这也正是病毒的两大特性)。

不论宏传播的方式采用的是感染 Normal 模板,还是复制自身到其他文档,或者成为发送邮件的附件,它们都得有一个拷贝自身的动作。当然这个拷贝方法就很多了,最直接的莫过于调用系统的 MacroCopy,OrganizerCopy,稍微复杂点儿就是用 InsertLine(ReplaceLine)或者 Print 函数一行一行地选择性复制;或者先把自身代码存为字符串经过修改后用 AddFromString 函数;更复杂的就是把自己复制到某个临时文件,然后打开文件,用 Import 函数完成复制;如果是作为邮件的附件发送,一定有诸如 Attachments.Add 的语句。当然也不排除调用自己写的子函数来实现复制的可能,所以在对模块的第一次处理过程中,即使不能判断是否有拷贝动作,也要记录和其他模块的调用关系,这也是语法分析中必须做的一件事。

而病毒对系统的破坏只能是通过一些系统调用来实现,比如删除文件的 Kill 函数,删除文件夹的 DelTree,格式化硬盘的 Format,重启系统的 Reboot 等等,所以我们需要建立一个病毒破坏系统可能使用的函数名的表格,在语法分析中再查表,记录所有出现在表格中的函数。

但是对于病毒的复制就不仅仅是记录函数名那样简单,因为正常的程序也可能使用这些函数,所以为了进一步考察,我们的语法分析还需要记录这些函数被调用时的上下文,比如它是否嵌入在循环语句中,它复制的对象是谁,它复制的目的地是哪儿。

5.3　对模块的行为作出分析

分析的目标无非两个:是否有复制,是否有破坏系统。

对一个模块是否有拷贝动作,我们不能简单地回答有还是没有,因为即使程序员拿到一个模块的代码也不一定能准确判断其含义,更何况电脑呢? 所以分析的结果只要求对每个模块有否拷贝动作作出一个评估,也就是赋予一个权值(0~100),权值越高,复制的可能性就越大,具体方法如下:

(1)首先对前面提到的系统拷贝函数赋予不同的权值见表 1。

表1 拷贝函数赋予不同的权值

函数	权值
MacCopy	100
OrganCopy	100
InsLine	85
RepLine	75
Prt	65
AddFromString	60
Import	70
Attach. Add	85

(2)其次考虑这些函数被调用时的上下文环境。如果它还同时在某个循环体(for,while)里面,则权值乘以110%(最大值不超过100)。

(3)最后考虑模块的调用关系,相当于对调用自定义函数重新赋予权值(第一次没有考虑,相当于是0),这种互相调用的关系只考察一次,也就是A调用B,B调用C,最终结果只是A调B,而不会体现A调用C。因为如果C有可疑的话,在B里面应该得到体现。

5.4 分析、计算权值

根据前面提到的关键指令(可以有相应变化)和病毒的两大特性赋予权值。具体方法如下:

首先参照初始表格进行赋值,见表2。

表2 初始表格

文件操作	95
中断向量表的修改	80
内存操作	65
磁盘的直接访问	70

赋值后,再结合前面用虚拟机分析时记录的与这些关键指令有关的上下文,重新计算;这就与具体指令有关。举个例子,如果是文件操作,再加上操作目标是"*.com",那么就乘以130%。

最后再一次调整权值,使得值都在100以下。

6 结 论

我们的启发式搜索算法和虚拟机技术结合的非常紧密。以行为监视为基础,根据启发式搜索算法分析程序的代码,在经过初步检查,发现可疑代码后,再将代码载入虚拟机运行,扫描并记录其操作以判断是否为病毒。这样结合判定,使启发式搜索技术比其他同类技术更加有效。可以预计启发式宏病毒扫描技术作为一种崭新的技术虽然不能完全代替传统的特征码扫描技术,但它在病毒界会得到越来越多的重视,在反病毒"战场"上也会起到越来越大的作用。

参考文献:

[1] Mario Camou, Aaron Von Cowenberghe. Debian/GNU Linux 2.1 Unleashed[M]. 陈河南,等译. 北京:清华大学出版社,2004.
[2] 杨守君. 黑客技术与网络安全[M]. 北京:中国对外翻译出版社,2004.
[3] 张友生. 计算机病毒与木马程序剖析[M]. 北京:科海电子出版社,2003.
[4] kidding. 病毒编写教程(DOS篇) EB/OL. (2006-03)[2006-11-10]http://codeguru.cn/show thread.aspx? postid=63.
[5] 程胜利,等. 计算机病毒及防止技术[M]. 北京:清华大学出版社,2004.

责任编辑:钟 声

A method of heuristic scanning for macro virus

SUN Wei[1], FENG Ping[2]
(1. Administration Office, Changchun University, Changchun 130022, China;
2. Computer Science and Technology Institute, Changchun University, Changchun 130022, China)

Abstract: The traditional method for detecting virus is signature scanning. Although it's very efficient for the known virus, it has no use for the new virus about which we have not known. With the two common characteristics(replication and destroying), the heuristic scanning can detect the unknown virus efficiently. This paper will introduce a detailed method of heuristic scanning for macro virus.
Keywords: macro virus; heuristic; virtual machine

一种启发式宏病毒扫描技术

作者： 孙伟， 冯萍， SUN Wei， FENG Ping
作者单位： 孙伟,SUN Wei(长春大学,教务处,吉林,长春,130022)， 冯萍,FENG Ping(长春大学,计算机科学技术学院,吉林,长春,130022)
刊名： 长春大学学报（自然科学版）
英文刊名： JOURNAL OF CHANGCHUN UNIVERSITY
年，卷(期)： 2007,17(1)

参考文献(5条)

1. Mario Camou;Aaron Von Cowenberghe;陈河南 Debian/GNU Linux 2.1 Unleashed 2004

2. 杨守君 黑客技术与网络安全 2004

3. 张友生 计算机病毒与木马程序剖析 2003

4. kidding 病毒编写教程(DOS篇) 2006

5. 程胜利 计算机病毒及防止技术 2004

本文读者也读过(10条)