第 38 卷 第 12 期
Vol.38 No.12

计 算 机 工 程
Computer Engineering

2012 年 6 月
June 2012

·安全技术· 文章编号：1000—3428(2012)12—0118—04 文献标识码：A 中图分类号：TP309.2

基于特征码的 PE 文件自动免杀策略

吴伟民，范炜锋，王志月，李晓峰，黄健炜

(广东工业大学计算机学院，广州 510006)

摘 要：设计一种以逐块恢复法替代传统逐块替换法的特征码定位算法，在此基础上提出一种针对不同区段进行自动免杀的策略。将该策略与改进的多重特征码定位算法相结合，在保持被免杀软件原有功能的前提下，使用等价代码替换技术、字符串与输入表函数名位移等方法自动进行特征码的去除和替换，由此避免被杀毒软件识别为恶意软件。实验结果验证了该策略的有效性。

关键词：特征码；定位；免杀；PE 文件；等价代码替换；输入表

PE File Auto Free-antivirus Strategy Based on Characteristic Code

WU Wei-min, FAN Wei-feng, WANG Zhi-yue, LI Xiao-feng, HUANG Jian-wei

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

【Abstract】This paper designs a characteristic code locating algorithm by using block-by-block recovery method instead of replacement method and proposes an auto free-antivirus strategy based on different PE sections. Under the premise of maintaining the original functionality of software after being modified to avoid killing, the combination of the strategy and the improved multi-characteristic code, it uses the equivalent code replacement, shifting methods of string and import table functions to automatically remove and replace the characteristic codes in tempt to avoid killing by anti-virus software. Experimental results verify the effectiveness of the strategy.

【Key words】characteristic code; locating; free-antivirus; PE file; equivalent code replacement; import table

DOI: 10.3969/j.issn.1000-3428.2012.12.035

1 概述

在计算机应用领域，杀毒软件起维护系统安全的重要作用，但仍然存在误杀一些非恶意软件的现象。遭误杀的软件主要分成两大类：一是某些具有专门用途的软件，可能涉及到操作系统底层甚至硬件层的访问及修改；另一种是基于对抗逆向分析目的，而在发布前经过代码混淆保护的正常软件。这两大类软件往往含有容易被杀毒软件检测到的特征码而遭误杀。

当今杀毒软件的主流技术以特征码检测为主，究其原因，一是该技术研究开始最早，发展较为成熟；二是其能够有效标识病毒，且基于特征码检测开发的引擎检测效率和准确性高，因此，为大多数防毒软件厂商所采用。鉴于此，国内外逐步开展了基于特征码的免杀技术研究。

目前研究成果有十六进制免杀、指令顺序调换法、加冷门壳等技术以及在此基础上发展而来的变形技术等，大多是对软件本身含有的特征码进行改写或隐藏。但此类技术具有明显的缺陷：一方面，已有的免杀技术都需要知识背景与有经验的程序员进行手工操作；另一方面，后来发展的变形免杀中大多针对特定软件个体进行免杀，并不适合批量的自动处理。为此，本文提出一种基于特征码的自动免杀算法，以突破需要手工介入修改特征码的限制。

2 特征码检测技术

特征码检测技术是当今众多杀毒软件的主流技术，其检测技术较成熟，正确率也较高。因此，基于特征码修改的免杀技术躲过杀毒软件的扫描成功率也较高。

2.1 杀毒软件检测技术

杀毒软件的检测技术具体分为特征码检测法、检验和检测法、行为检测法、软件模拟法、VICE 先知扫描法等。

特征码检测技术是发展至今最为成熟的病毒检测技术之一，防毒软件供应商通过收集病毒样本，提取在恶意代码中专有的十六进制作为该病毒标识并建立病毒库，供病毒扫描器使用串匹配算法 BF(Brute Force)、KMP 算法等模式匹配算法进行特征码匹配[1-2]。

使用特征码检测方法的优点在于特征码提取方便且唯一，能有效识别病毒的类型与行为，并给出相应的处理。但其缺点也很明显，不能检测出变形与暂未发现的病毒，且随着病毒的增多，特征码库的体积加大，会使程序效率低下。

2.2 病毒特征码

特征码是指防毒软件从病毒样本中提取的一段可用于唯一标识病毒特征码的二进制串，该二进制码串可位于文件的不同区段，特征码的提取方法与体积也因防毒软件病毒库设计规范的不同而不同[3]。特征码具体分为以下 3 类[4]：

(1)单一特征码：从病毒样本中抽取的连续的一小段二进制串。这种特征码的特点是所占空间小，需要匹配的时间短，内存的开销小。

(2)多重特征码：是指文件中含有一个以上的特征码串，只要杀毒软件成功匹配某一处特征码，便可以确定该文件为病毒并进行清除。

(3)复合特征码：该特征码与多重特征码类似，是指文件包含 2 处以上的特征码串，但与多重特征码不同的是，复合特征码需要多处特征码同时存在时才会被杀毒软件确定为病毒。

作者简介：吴伟民(1956－)，男，教授，主研方向：软件安全，数据结构与算法，可视计算；范炜锋、王志月，硕士研究生；李晓峰、黄健炜，本科生

收稿日期：2011-08-01 **E-mail：**strayfan@126.com

2.3　传统特征码定位算法

修改特征码必须对 PE 文件中包含的特征码进行定位，并根据其所在的位置选择相应的修改策略。

传统的定位方法是采用逐块替换法，其思想是将 PE 文件的某一个区段平均分割成 *N* 个小区间，每次使用 0x00 或者 0x90 替换其中一个小区间，分别进行 *N* 次操作[4]。

使用该方法定位单一特征码与复合特征码时，当替换到特征码的部分内容即可定位到特征码的大致位置。但此方法不适用于多重特征码，主要是由于需要多次手工操作，尝试覆盖所有特征码范围，但在进行精确定位的过程中相互之间的特征码又会产生干扰，以至于无法正确地定位到特征码。

传统的逐块替换法在定位多重特征码上存在不足，因此，需要对该方法进行改进，使其可以定位不同类型的特征码。

3　基于特征码的自动定位与免杀算法

3.1　多重特征码自动定位算法

本文基于逐块替换法思想，设计逐块恢复法进行定位。其基本思想为：假设当前要对区段.text 进行特征码定位，将该区段平均划分为 *N* 个区间，生成 *N* 个文件，第 *n*(*n*≤*N*)个文件中包含了前面 *n* 个区间的内容，以此类推，第 *N* 个文件包含了整个区段的内容。

逐块恢复法的核心意义在于该算法能将文件内容逐块暴露出来，如果文件中含有特征码，则该特征码在文件内容逐渐暴露的情况下也逐渐暴露，第 *n* 个被查杀的文件表明该区段的第 *n* 个区间里含有特征码。对于多重特征码，因第 *n* 个区间之后的内容尚未恢复，所以多个特征码之间不产生干扰。

在特征码定位过程中还需要注意 2 点：一是保留 PE 文件的头部内容；另一个是算法的效率。由于在定位的过程中通过生成大量的文件让杀毒软件扫描，时间开销主要在于文件的生成与杀毒软件的扫描上，因此需要控制区间的大小来提高程序的运行效率。

根据上述情况，设计的多重特征码定位算法如下：

(1)指定杀毒软件安装启动目录与创建文件输出目录 output 文件夹。

(2)读取要处理的文件，检验 DOS 头部与 PE 头部的 Magic 字段。如果所处理文件非 PE 文件，则退出程序。

(3)读取文件所有头部信息与输入表。

(4)生成文件副本，以命令行启动杀毒软件对副本文件进行扫描，如果该文件未被删除，则退出程序。

(5)根据第(3)步所得的区段信息，从第一个区段开始进行特征码定位：

1)将整个文件读取到缓冲区中，保留该区段与 PE 文件所有头部信息，生成文件并启动杀毒软件，如果文件未被删除，转第 7)步。

2)应用逐块恢复法，生成 *N* 个文件到 output 目录，待杀毒软件扫描完毕之后浏览文件夹，查找第一个消失的文件，并记录下该文件中包含的区间信息。

3)将文件缓冲区中在第 2)步中记录下的区间内容全部替换，重复第 2)步、第 3)步，直到生成的文件都不会被查杀为止。

4)根据前面得到的特征码区间信息，对得到的第个区间采用逐字节替换的方法，每次只替换该段中 1 个或 2 个字节的内容，并生成文件。

5)启动杀毒软件进行扫描，经过扫描后剩下的文件说明第 4)步中替换到的位置是特征码所在的精确位置，可以通过修改该位置的内容来达到免杀的效果，因此，将该位置信息保留下来。

6)判断第 3)步中得到的特征码区间是否已全部精确定位结束，是，则转第 7)步，否则，重复第 4)步。

7)判断当前区段是否为最后一个进行定位的区段，是，则继续第(6)步，否则，取下一个区段重复第(5)步，直至所有区段处理结束。

(6)对前面得到的特征码进行整理，合并相邻的特征码区间。

3.2　基于特征码的自动免杀策略

由于特征码可能出现在 PE 文件不同区段，因此需要根据区段的属性与作用来选择不同的免杀策略。本文针对特征码较常出现的 3 个区段——.text 段、数据段与输入表分别设计了适合自动处理的免杀策略。

(1)基于代码段的等价代码替换算法

等价替换法是针对特征码出现在.text 段时采用的一种处理方法。在 Windows NT 系列 OS 下的 PE 文件所采用的二进制编码是 Intel 80X86 指令编码，某些汇编指令可以由某些功能相近的指令进行替换，或由几条指令组合完成该指令的功能。例如将 eax 置 0 的汇编指令有“Mov eax, 0”和“Xor eax, eax”，它们是等价的。因此，等价代码替换算法是指用功能相近或者相等的汇编指令来替换当前的指令。如果该指令是病毒库中特征码串中的某个特征码字节，由于不同指令的二进制编码不一样，替换之后特征码串匹配失效，因此该文件具有免杀效果。

使用等价替换法需要构建一个等价代码库，下面列举一些常见的等价代码表，如表 1、表 2 所示的 ADD 等价指令，表 3 所示的 CMP 等价指令，表 4 所示的 PUSH、RETN 等价指令，表 5、表 6 所示的跳转指令的等价替换[5]。

表 1　ADD 等价代码表 1

汇编指令	操作码	替换指令	操作码
add　eax, op	/x83/xC0	adc　eax, op	/x83/xD0
add　ebx, op	/x83/xC3	adc　ebx, op	/x83/xD3

表 2　ADD 等价代码表 2

汇编指令	操作码	替换指令	操作码
add　eax, op	/x83/xC0	sub　eax, -op	/x83/xE8
add　ebx, op	/x83/xC3	sub　ebx, -op	/x83/xEB

表 3　CMP 等价代码表

汇编指令	操作码	替换指令	操作码
cmp　eax, op	/x83/xF8	sub　eax, op	/x83/xE8
cmp　ebx, op	/x83/xFB	sub　ebx, op	/x83/xEB

表 4　PUSH、RETN 等价代码表

汇编指令	操作码	替换指令	操作码
push　reg/addr retn	/x50 或者/x68	call	/xFF

表 5　JMP 等价代码表 1

汇编指令	操作码	替换指令	操作码
jb　addr	/x72	jbe　addr	/x76
jb　addr	/x72	jl　addr	/x7C
jb　addr	/x72	jng　addr	/x7E

表 6 JMP 等价代码表 2

汇编指令	操作码	替换指令	操作码
jmp eax	/xFF/xE0	call eax	/xFF/xD0
jmp dword ptr ds:[addr]	/xFF/x25	mov eax, dword ptr ds:[addr] jmp eax	/xA1 /xFF/xE0

(2)基于数据段的字符串大小写修改与位移算法

当定位到的特征码出现在.data 段或者.rdata 段，且该位置中的内容是一段连续的字符串时，可以采取改变字母大小写的方法，但是做如此修改有一个前提，是该字符串用于 I/O 显示的或者是用于查找使用的目录，该方法的使用范围比较窄，如图 1 所示，是字符串的一种情况。

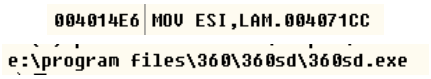

图 1 字符串作为文件目录使用的情况

上述情况之下，由于该字符串用于 CMD 的 I/O 窗口或 Windows 程序的窗口显示，又或者是访问磁盘的绝对或相对路径，对其中的字母进行大小写修改的时候，并不会使得程序崩溃。但并非所有的字符串都能修改，例如一个系统 I/O 函数中的参数，类似 printf，sprintf 这些格式 I/O 函数的参数中大多都带有字符%与格式说明符，此类字符并不能应用大小写修改法。针对该种情况，需要采用字符串位移算法。对字符串进行位移是指将该字符串向左边或者右边空白的地区移动数个字节，移动之后必须对代码段中引用到该字符串的所有指令的引用地址进行修正，使得程序能够正确地引用到该字符串。

(3)基于输入表的输入函数名位移算法

可执行文件使用来自其他 DLL 的代码或者数据时，称为输入。当 Windows 加载器装入 PE 文件时，它的工作之一是将 PE 文件中的输入表的函数和数据进行定位，使得装入的文件可以正确地使用装载器加载的地址而调用所需要函数[6-7]。

当应用程序调用一个 DLL 的代码和数据时，属于隐式地链接该 DLL。应用程序同样可以自己显式地链接 DLL，自己完成加载器的工作，通常所有的数据和地址的填写都是通过调用 LoadLibraryA 和 GetProcAddress 来完成的。

对输入函数在源码中的调用处经过编译后可能的形式会变成：

```
401011:    CALL 00401198
...
401198:    JMP DWORD PTR [402010]
```

也可能被优化成一条语句：

```
CALL DWORD PTR [402010];
```

输入表的结构如图 2 所示[6-7]。

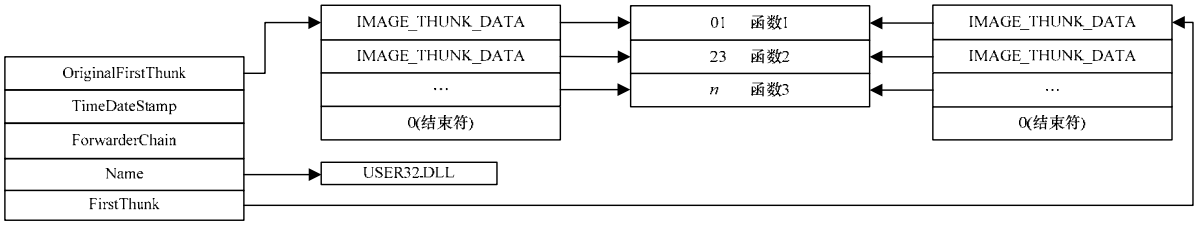

图 2 输入表结构

各字段说明如下：

1)OriginalFirstThunk：该字段中存放着指向输入名称表 INT 的虚拟地址，INT 是一个 IMAGE_THUNK_DATA 的结构数组，其中每一个 IMAGE_THUNK_DATA 都包含着输入函数名字字符串的地址与序号。

2)TimeDateStamp：该字段用于标示时间。

3)ForwardChain：这是第一个被转向的 API 的索引，一般为 0。

4)Name：该字段是 DLL 名字的虚拟地址。

5)FirstThunk：包含指向输入地址表 IAT 的虚拟地址。IAT 在未载入内存之前与 INT 的内容是相同的，都指向于 IMAGE_THUNK_DATA 数组。在载入内存之后，由 PE 装载器将该地址中的内容填充为输入函数的内存地址[6-7]。

输入表函数位移算法是指对输入表函数的字符串做移位操作。在进行相应操作时，要对 OriginalFirstThunk 与 FirstThunk 中的偏移地址进行修改，以便 PE 装载器在加载 PE 文件时可以正确地找到函数名字的字符串所在地址。另外，当输入表被并入.data 或者.rdata 区段时，还需要对所修改的位置进行判断，辨别该位置的内容是数据还是输入表。

(4)基于特征码的自动免杀算法

在特征码定位结束后，根据特征码所在的不同区段，选择上文所提出的策略，设计如下自动免杀算法流程[8]：

1)对前面得到的特征码区间进行分队列，属于同一区段的分裂成同一队列。同时读取整个文件到缓冲区。

2)根据当前队列所属的区段，选择不同的策略：

①如果该特征码属于.text 段，则首先调用反汇编器，对.text 段进行反汇编。将反汇编结果保存到 txt 文件中，并读取该 txt 文件，同时判断特征码所在的位置是代码还是数据，如果为代码，则调用转换函数，查找所构造的等价代码表，查询是否存在可供替换的指令。如果未能找到可供替换的指令，但该指令的字节数等于远跳转指令的字节数，采用通用跳转法将该指令移动到其他位置，原位置用跳转指令替换。如果该特征码是数据，则递归处理下一处特征码。

②如果该特征码属于.data 或者.rdata 段，则判断该处特征码是否在输入表的 OriginalFirstThunk、FirstThunk 或者在函数名字符串上，如果是，则只处理函数名字符串的情况，对其进行位移，并改变相应的 OriginalFirstThunk、FirstThunk，其他情况则递归处理下一处特征码；否则，读取附近数十字节的内容，并判断是否存在无法改变大小写的字符，如果是，则递归处理下一处特征码；否则，改变该字节处理特征码字符的大小写。

③如果该特征码属于.idata 段，则进行函数名位移处理。

3)用第 2)步修改的缓冲区生成文件，同时启动杀毒软件进行查杀，如果该文件不被查杀，表示免杀成功；否则，免杀失败。

4 实验结果

利用上述算法实现的特征码定位与修改工具对某个被 360 杀毒软件报毒的 PE 文件进行处理，得到的特征码结果如

图 3 所示。

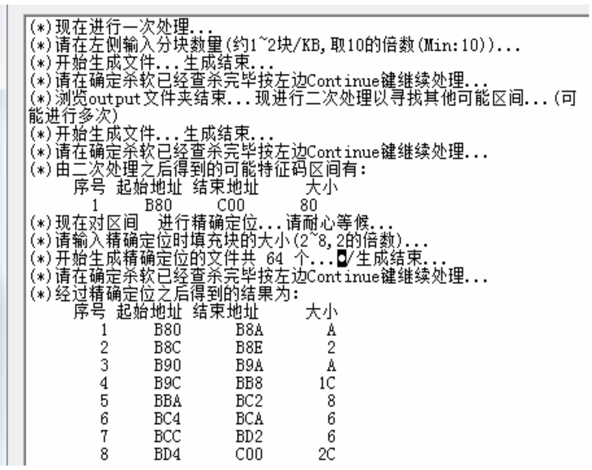

图 3　特征码定位结果

用 OD 打开该处的特征码可以看到如图 4 所示的内容。由图 4 可以看到，该处特征码是输入表函数的调用，但由于可供替换的指令字节数比原来的指令多一个字节，因此采用通用跳转法进行修改。处理结果如图 5 所示。

图 4　特征码内容

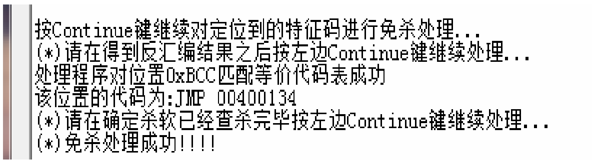

图 5　处理结果

实验结果验证了本文免杀策略的正确性。从图 3 可看到，改进的多重特征码定位算法能精确地确定特征码的分布情况，其精确程度可达到单个字节且定位所需要的时间非常短。通过匹配等价代码表，使用等价代码修改之后，由于特征码匹配失败，修改后的程序都能躲避杀毒软件的强制删除。

5　结束语

本文针对主流的特征码检测技术提出了适用于自动免杀的特征码免杀策略，并基本实现了这些免杀策略。该策略基于特征码所在区段的不同属性与作用，在改进的多重特征码定位算法的辅助下，通过等价代码替换、字符串大小写与位移、函数名位移 3 种方法分别处理特征码处于代码段、数据段、输入表的情况。该策略达到了自动定位与处理非恶意软件的特征码，避免杀毒软件误杀的目的。另外所构建的等价代码表易于扩充，并在保证不破坏程序原有功能的前提下作最全面的修改。下一步是将本文所设计的定位方法与修改策略扩展应用于内存特征码的免杀，并作进一步完善。

参考文献

[1] 周才学. 计算机病毒与反病毒检测技术[J]. 九江学院学报: 自然科学版, 2005, 20(2): 33-35.

[2] 陈晋福. 基于 PE 文件的启发式特征码自动提取的研究[D]. 成都: 电子科技大学. 2010

[3] 陈　健，范明钰. 基于恶意软件分类的特征码提取方法[J]. 计算机应用, 2011, 31(1): 83-84.

[4] 张迎春. 基于特征码技术的攻防策略[J]. 计算机系统应用, 2009, 18(3): 114-117.

[5] Intel 64 and IA 32 Architectures Software Developers' Manual Volume 2: Instruction Set Reference[M]. [S. l.]: Intel Corporation, 2007.

[6] 段　钢. 加密与解密[M]. 3 版. 北京: 电子工业出版社, 2008: 265-314.

[7] Pietrek M. Windows 95 System Programming Secrets[M]. [S. l.]: IDG Books Worldwide Inc., 1996.

[8] Christodorescu M, Jha S. Testing Malware Detectors[C]//Proc. of ACM SIGSOFT International Symposium on Software Testing and Analysis. Boston, USA: ACM Press, 2004.

编辑　张　帆

~~~~~~

(上接第 117 页)

参考文献

[1] Trusted Computing Platform Alliance. Trusted Computing[EB/OL]. [2011-06-22]. http://www. trustedcomputinggroup.org.

[2] Sailer R, Zhang Xiaolan, Jaeger T, et al. Design and Implementation of a TCG-based Integrity Measurement Architecture[C]//Proc. of USENIX Security Conference. Berkeley, Germany: USENIX Association, 2004: 223-238.

[3] Intel Corporation. Intel Trusted Execution Technology(Intel TXT) Software Development Guide & Measured Launched Environment Developer's Guide[EB/OL]. [2011-04-24]. http://download.intel.com/technology/security/downloads/315168.pdf.

[4] Intel Corporation. Trusted Boot(tboot)[EB/OL]. (2007-09-12). http://tboot.source forge.net/.

[5] State Password Administration Committee. Functionality and Interface Specification of Cryptographic Support Platform for Trusted Computing[EB/OL]. [2011-04-24]. http://www.oscca.gov.cn/.

[6] Wright C, Cowan C, Smalley S, et al. Linux Security Modules: General Security Support for the Linux Kernel[C]//Proc. of the 11th USENIX Security Symposium. Berkeley, Germany: USENIX Association, 2002: 17-31.

[7] LibPCAP[EB/OL]. [2011-05-24]. http://www.tcpdump.org/.

[8] 徐　震，沈丽红，汪　丹. 一种可配置的可信引导系统[J]. 中国科学院研究生院学报, 2008, 25(5): 626-630.

编辑　张　帆
~~~~~~