基于导入表迁移的PE文件信息隐藏技术研究

田祖伟 李勇帆 刘 洋
(湖南第一师范学院信息科学与工程学院 长沙 410205)

摘 要 在分析PE文件导入表结构的基础上,利用Windows加载器的工作原理和PE文件导入表存储位置不确定性的特点,提出了一种迁移PE文件导入表并将信息隐藏在PE文件原导入表位置的信息隐藏算法。理论分析与实验结果表明,该算法较好地弥补了传统PE文件信息隐藏算法中隐藏信息过于集中、交换PE文件导入表数据结构元素将破坏隐藏信息的不足,提高了隐蔽性和抗攻击性。

关键词 PE文件,信息隐藏,导入表,导入表迁移
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.046

Research of PE File Information Hiding Based on Import Table Migration

TIAN Zu-wei LI Yong-fan LIU Yang
(School of Information Science & Engineering, Hunan First Normal University, Changsha 410205, China)

Abstract On the base of analyzing the import table of PE file, using the work principle of Windows loader and the uncertainty of PE file import table's storage location, an algorithm based on the import table migration was proposed. Information is hidden in the original import table space of PE file. Theory analysis and experiment result show that the algorithm overcomes the disadvantages of previous hiding information schemes, such as hidden information convergence and destruction of import table, which improves hidden and anti-attack ability.

Keywords PE file, Information hiding, Import table, Import table migration

1 引言

导入表是PE文件中一个很重要的数据结构,是PE文件实现动态链接的基础。Windows平台下绝大部分PE文件都需要通过导入表调用核心动态链接库所提供的系统函数以完成软件自身的功能[1,2]。一般来说,Windows应用程序都会调用3个重要的动态链接库kernel32.dll、user32.dll、gdi32.dll中的相关函数,其中kernel32.dll属于内核级文件,它提供了系统的存储管理、I/O管理、中断管理等功能函数;user32.dll文件是Windows用户界面相关应用程序的接口,提供Windows处理、基本用户界面等特性,如创建窗口和发送消息等系统功能函数;gdi32.dll是Windows GDI图形用户界面相关程序,提供绘制图像和显示文字等功能的函数。

导入表是PE文件中一个重要的数据结构,其中存储了PE文件调用的全部导入函数的相关信息。在加载PE文件时,Windows加载器将相关的动态链接库文件读入内存,并将导入函数在内存中的实际地址填充到导入表的相关结构中,为调用导入函数做好准备。导入函数又称为引入函数或输入函数,是被程序调用但其执行代码位于动态链接库(DLL)中的函数,调用程序仅保留一些函数的基本信息,包括函数名和所在DLL文件名等,这些信息都保存在PE文件的导入表中。PE文件未加载时,导入函数在内存的实际地址不确定,当程序执行时,Windows加载器加载所有导入DLL文件,并通过PE文件导入表将调用导入函数的指令和函数在内存中的真实地址关联起来。

龙飞宇等人[3]提出了一种对PE文件引入表数据结构进行变换的信息隐藏算法。该方法通过分析PE文件引入表数据结构的特点和导入函数调用方法,将隐秘信息嵌入到PE文件引入表结构和导入函数的排列顺序之中。实验结果表明,该方法比利用PE文件冗余空间和资源结构的水印算法有更好的隐蔽性和更强的鲁棒性,提供了更加安全的软件版权保护方式。但PE文件导入表结构有其默认的顺序,同一模块内导入函数的排列是按函数名字符串升序排列的,改变导入表的结构和函数的排列顺序将会引起攻击者的怀疑,安全性不高;同时,重排导入表的结构和函数的排列顺序将会破坏隐藏的信息,鲁棒性不好。

端木庆峰等[4]利用扩频通信的工作原理,提出一种新的扩频软件水印方案,即将PE文件中不同导入函数的调用次数作为程序的特征向量,通过修改特征向量的各分量值,将水印信号分散嵌入到PE文件代码节中。实验表明,与现有软件水印相比,该方案隐蔽性高,实现简单,且能够有效抵抗多种攻击。但该算法是通过改变导入函数的引用次数来实现

到稿日期:2014-11-12 返修日期:2015-01-22 本文受国家自然科学基金项目(61373132),基础教育信息化技术湖南省重点实验室(2015TP1017),湖南省普通高等学校教学改革研究项目(2012[528]),湖南省大学生研究性学习和创新性实验计划项目(2014[248])资助。
田祖伟(1973-),男,博士,教授,主要研究方向为信息安全、算法分析与设计、编译优化,E-mail:tianzuwei@126.com;李勇帆(1959-),男,教授,主要研究方向为多媒体技术、教育技术学。

的，需要在代码中添加冗余的导入函数调用代码，一方面冗余导入函数调用代码的添加将降低程序性能，另一方面通过修改已编译好的二进制代码来增加导入函数的调用次数非常困难。该算法只适用于有源代码的程序，在源代码文件中增加对导入函数的调用，从而改变程序的特征向量。

2 PE 文件导入表结构

导入表是数据目录中注册的数据类型之一，其描述信息位于数据目录的第 2 个目录中。在 winnt. h 中定义了 IMAGE_IMPORT_DESCRIPTOR 结构以描述导入表。导入表采用一个二级索引结构，第一级是一个模块目录 IMAGE_IMPORT_DESCRIPTOR 数组（简称 IID 数组），它的每个元素描述了一个 DLL 动态链接库文件，以全 0 字节作为数组的结束标志；第二级是两个类似的导入名称表 INT 或导入地址表 IAT，INT 或 IAT 的每个元素指向一个 IMAGE_IMPORT_BY_NAME结构（包括一个函数序号和一个函数名称字符串）。

PE 文件导入表结构如图 1 所示。

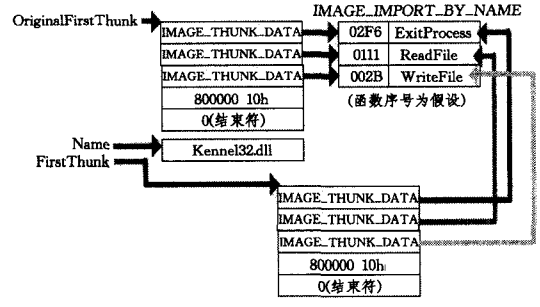

图 1 PE 文件导入表结构

2.1 IMAGE_IMPORT_DESCRIPTOR 结构的定义

在 PE 文件中导入表用 IMAGE_IMPORT_DESCRIPTOR(IID)结构的数组进行定义。每个 IID 结构数组元素描述 PE 文件引用的一个动态链接库 DLL 的相关信息，有 5 个域成员，共占 20 个字节，最后一个 IID 结构的所有成员都为 0，表示导入表描述符结构结束，即通过导入表起始地址和这个全 0 的元素可以计算出导入表中引用的动态链接库的个数。该结构的具体定义如下：

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR
{ union
    { DWORD Characteristics;
      DWORD OriginalFirstThunk;
    };
  DWORD TimeDateStamp;
  DWORD ForwarderChain;
  DWORD Name;
  DWORD FirstThunk;
} IMAGE_IMPORT_DESCRIPTOR;
```

其中结构成员 OriginalFirstThunk 为双字数据类型（DWORD 类型），其值为导入名称表（Import Name Table，INT）的相对虚拟地址。导入名称表是一个 IMAGE_THUNK_DATA 结构数组，一个数组元素描述一个导入函数的信息，数组元素的值为全 0 表示 IMAGE_THUNK_DATA 结构结束。该结构成员有两种情况：1）当最高位为 0 时，表示导入符号是一个数值，该数值是一个 RVA；2）当最高位为 1 时，表示导入符号是一个名称。结构成员 FirstThunk 指向导入地址表（Import Address Table，IAT）的 RVA。IAT 是一个 IMAGE_THUNK_DATA 数组，其中的数据成员 FirstThunk 和 OriginalFirstThunk 分别指向两个本质相同的 IMAGE_THUNK_DATA 结构。

2.2 IMAGE_THUNK_DATA 结构的定义

IMAGE_THUNK_DATA 用于定义导入表中导入函数的相关信息。IMAGE_THUNK_DATA 的详细定义如下：

```
typedef struct _IMAGE_THUNK_DATA32
{ union { DWORD ForwarderString;
          DWORD Function;
          DWORD Ordinal;
          DWORD AddressOfData;
  } u1;
} IMAGE_THUNK_DATA32;
```

这是一个双字的共用体数据结构，当最高位为 1 时，表示函数按序号方式导入，共用体成员 Ordinal 的低 31 位表示导入函数序号，最高位为 0 时，表示函数按字符串方式导入。共用体成员 AddressOfData 的值为指向导入函数名称的相对虚拟地址。

2.3 IMAGE_IMPORT_BY_NAME 结构的定义

IMAGE_IMPORT_BY_NAME 结构用于存放一个导入函数的相关信息，定义如下：

```
typedef struct _IMAGE_IMPORT_BY_NAME {
    WORD Hint;
    BYTE Name[1];
} IMAGE_IMPORT_BY_NAME;
```

成员 Hint 标识了导入函数在 DLL 中的索引号，PE 加载器利用 Hint 字段可以加快在动态链接库 DLL 文件的导出表中查询函数的速度，该成员的值一般被赋为 0。Name 字段标识导入函数的函数名，即一个以 0 为结束标志的 ASCII 码字符串。

3 基于导入表迁移的信息隐藏算法

导入表是 PE 文件中一个重要的数据结构，当程序中调用了动态链接库中的相关函数并在编译和链接时，编译器就会将调用函数的相关信息写入 PE 文件的导入表中[5-9]，通常位于. idata 节内。导入表是数据目录中注册的数据类型之一，其描述信息位于数据目录表 IMAGE_DATA_DIRECTORY 的第 2 个元素中，此元素存储了导入表的 RVA 和导入表的大小。访问 IMAGE_DATA_DIRECTORY 表中的第 2 元素，即可获取导入表在文件中的起始位置和大小，根据得到的 RVA 即可找到导入表相关的数据。在头文件 winnt. h 中定义了 IMAGE_IMPORT_DESCRIPTOR 结构用于描述导入表。IAT 表也是数据目录中注册的数据类型之一，IAT 表的 RVA 和大小存储在数据目录表的第 11 个元素中，通过同样的方法，可以访问 IAT 表中的数据。

通过对 PE 文件导入表数据结构的分析，以及 PE 文件利用导入表查找所调用的导入函数的真实入口地址的工作原理，可以得知，PE 文件的导入表和 IAT 表在文件内的位置是可以迁移的，可以在常量节中，也可以在代码节或其他任何可读属性的节里。为此，提出了一种基于移动 PE 文件导入表的信息隐藏算法，其主要思想是迁移 PE 文件导入表，将秘密

信息嵌入到导入表原来的位置区域。

3.1 信息嵌入

本算法通过将 PE 文件整个导入表移动至节内的冗余空间位置，然后将信息写入导入表原来的位置，来实现信息隐藏（不迁移 IAT 表），详细描述如下。

输入：原始载体 PE 文件 P，待隐藏的信息 M，公钥 pk

输出：隐写 PE 文件 P′

步骤 1：分析原始载体 PE 文件 P，判断是否为合法 PE 文件，读取 PE 文件头信息，并封装成数据对象备用；

步骤 2：判断是否已经隐藏了信息，判断 IMAGE_DATA_DIRECTORY 数组的第 7 个对象的 SIZE 属性是否为 0xFFFFFFFF，是则说明已经隐藏了信息，退出；否则转步骤 3；

步骤 3：计算出导入表中 IMAGE_IMPORT_DESCRIPTOR(IID)在文件中的地址，并读出 IID 数组；

步骤 4：确定 IID 迁移后的新地址 ADDR；

步骤 5：根据 IID 找出所有 INT 表，读出所有的 INT 表；

步骤 6：根据 IID 数组和 INT 表的大小计算隐藏容量 C，如果隐藏容量 C 小于 M 的长度则退出，否则转步骤 7；

步骤 7：计算所有 INT 表的新文件地址和虚拟地址及该 PE 程序中各个节的实际大小；

步骤 8：将读取的 IID、INT、API 函数序号和名称的结构数组迁移到已经计算好的地址 ADDR 处；

步骤 9：修正各个节的实际大小、IID 的虚拟地址和 IID 数组中 OriginFirstThunk 地址；

步骤 10：将原 IID 的文件地址写入 IMAGE_DATA_DIRECTORY 数组第 7 个对象中的 ADDRESS 属性，并在其 SIZE 属性处写入隐藏标志 0xFFFFFFFF；

步骤 11：利用公钥 pk 和非对称加密算法 RSA，对待隐藏的信息进行加密，得到加密后的信息 M′：M′=E(pk,M)；

步骤 12：将原 IID 数组开始位置写入 IID 数组中可隐藏的数据的大小 C1，并将长度为 C1 个字节的隐秘信息 M′写入原 IID 数组的后续单元中，原 IID 数组最后 4 字节写入原 INT 数组的地址；

步骤 13：将 C−C1 的值写入原 INT 数组的开头两字节中，随后将长度为 C−C1 个字节的隐秘信息 M′写入原 INT 数组的后续单元中。

3.2 信息提取

首先分析 PE 文件，找到原 IID 数组的首地址，然后按约定读取隐藏在原 IID 数组中和原 INT 表中隐藏的信息并进行组装，最后解密即可，详细算法描述如下。

输入：含隐秘信息的隐写 PE 文件 P′，私钥 sk

输出：隐秘信息 M

步骤 1：分析含隐藏信息的隐写 PE 文件 P′，判断是否已经隐藏了信息，查看 IMAGE_DATA_DIRECTORY 数组的第 7 个对象的 SIZE 属性是否为 0xFFFFFFFF，是则说明已经隐藏了信息，转步骤 2，否则退出；

步骤 2：先根据 IMAGE_DATA_DIRECTORY 数组第 7 个对象中的 ADDRESS 属性找到原 IID 数据处，读出原 IID 中隐藏的数据大小和数据；

步骤 3：若原 IID 最后 4 字节不为 0，则读出该地址，找到原 INT 处读出隐藏的信息，并进行组装得到隐秘信息 M′；

步骤 4：利用私钥 sk 和非对称加密算法 RSA，对水印信息 M′进行解密，得到解密后的信息 M：M=D(sk,M′)。

4 实验结果与讨论

实验中用到的 PE 文件来自 Windows 系统文件夹和互联网。为了能够准确地计算出所提方法的嵌入容量，从 Windows 系统文件夹选择了 200 个 PE 文件，另外选择了系统中安装的 100 多个常用软件如 QQ. exe、thunder. exe 等 PE 文件作为测试文件。

1）嵌入容量

在测试集中，把水印信息嵌入到测试文件原导入表的 IID 和 INT 中，即满嵌。

从表 1 可以看出，基于导入表迁移的隐藏方案的隐藏容量与导入表的大小有关，一般来说，程序的功能越复杂，调用的导入函数越多，隐藏容量越大。

表 1 基于导入表迁移的信息隐藏算法隐藏容量分析

文件名	文件大小	引用 DLL 文件个数	导入函数个数	嵌入容量
write. exe	5632	3	19	474
notepad. exe	66560	9	201	4592
winmine. exe	119808	8	102	2347
acrord32. exe	1498552	9	544	13209
qq. exe	99744	7	93	2318
360sd. exe	1697400	14	845	3436
telnet. exe	85504	9	185	776
thunder. exe	1808176	29	1194	24738

与其他方法相比，本方案通过迁移 PE 文件的导入表后，在 PE 文件原导入表位置进行信息隐藏，隐蔽性较高，同时由于 PE 文件导入表结构的复杂性和在程序加载、运行时的重要性，一旦破坏，程序将不能正常加载和运行，因此本方法具有较高的安全性。

2）对程序启动时间的影响

本方案仅对导入表及相关数据进行迁移，没有移动及破坏 IAT 表中的内容。即使 PE 文件已进行绑定导入，此方案也不会破坏这种绑定，不会影响 PE 文件的加载速度。

表 2 导入表迁移后程序启动时间的分析

文件名	正常启动时间（ms）	迁移导入表后启动时间（ms）	性能损失
write. exe	15.4	15.4	0.0
notepad. exe	20.3	20.3	0.0
telnet. exe	64.2	64.2	0.0
winmine. exe	70.8	70.8	0.0
winrar. exe	39.4	39.4	0.0
qq. exe	27.3	27.3	0.0
360sd. exe	113.7	113.7	0.0
thunder. exe	83.8	83.8	0.0

所提方法的嵌入容量与程序调用的导入函数的数量有关，调用的导入函数越多，INT 表越大，API 函数序号和名称的结构数组越大，隐藏容量越大。由于没有迁移 IAT 表，对绑定导入表的程序来说，不会影响其加载速度和程序性能。与龙飞宇等人提出的变换 PE 文件引入表结构的软件水印算法相比，本文提出的方法能够抵抗针对导入表的结构变换攻击。

结束语 PE 文件是 Windows 平台可执行文件的标准格式，应用非常广泛，非常适合作为信息隐藏的载体。本文在分析 PE 文件导入表结构的基础上，利用 Windows 加载器的工作原理，提出了一种基于导入表迁移的 PE 文件信息隐藏算法。该算法较好地解决了传统 PE 文件信息隐藏算法中隐藏信息过于集中、交换 PE 文件导入表数据结构元素将破坏隐藏信息等问题，增强了隐蔽性和抗攻击性。

参 考 文 献

[1] Petzold C. Windows 程序设计(第 5 版 珍藏版)[M]. 方敏,张胜,梁路平,等译. 北京:清华大学出版社,2010:333-382

[2] Zaidan A A, Zaidan B B, Alanazi O H, et al. Novel approach for high (secure and rate) data hidden within triplex space for executable file[J]. Scientific Research and Essayss, 2010, 5(15): 1965-1977

[3] Long Fei-yu, Liu Jia-yong, Yuan Xi Software watermark based on structure transform of PE file import table[J]. Journal of Computer Applications, 2010, 30(1): 217-219(in Chinese)
龙飞宇,刘嘉勇,袁熹. 一种变换 PE 文件引入表结构的软件水印[J]. 计算机应用,2010,30(1):217-219

[4] Duanmu Qing-feng, Wang Yan-bo, Zhang Xiong-wei, et al. A spread spectrum software watermarking scheme based on the improt functions[J]. Journal of Computer Research and Development, 2009, 46(Suppl.): 88-92(in Chinese)
端木庆峰,王衍波,张雄伟,等. 基于导入函数引用次数的扩频软件水印方案[J]. 计算机研究与发展,2009,46(Suppl.):88-92

[5] Zhou Qing-lei, Li bin. Double software watermark scheme based on tamper-proofing[J]. Computer Engineering, 2013, 39(7): 185-188(in Chinese)
周清雷,李斌. 基于防篡改的双重软件水印方案[J]. 计算机工程,2013,39(7):185-188

[6] Zhang Meng, Chen Gou-xi, Zhang Peng-cheng. Executable file backdoor steganographic algorithm with highly efficient[J]. Application Research of Computers, 2013, 30(4): 1198-1200(in Chinese)
张萌,陈够喜,张鹏程. 高效可执行文件后门隐写算法[J]. 计算机应用研究,2013,30(4):1198-1200

[7] Jang J, Ji H, Hong J M, et al. Protecting Android applications with steganography-based software watermarking[C]// Proceedings of the 28th Annual ACM Symposium on Applied Computing. Coimbra, 2013: 1657-1658

[8] Wei Wei-min, Liu Kun, Wan Xiao-peng. High capacity information hiding based on PE file format[J]. Journal of Nanjing University of Science and Technology, 2015, 39(1): 45-49(in Chinese)
魏为民,刘锟,万晓鹏. PE 文件格式的大容量信息隐藏技术[J]. 南京理工大学学报,2015,39(1):45-49

[9] 刘家超. 基于行为分析的未知 PE 病毒检测技术研究[D]. 北京:北京邮电大学,2014

[10] Ji Yi-mu, Zhu Tong-hui, Chai Bo-zhou, et al. Hybrid encryption scheme and performance analsysi for user's privacy in cloud[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition), 2015, 27(5): 631-638(in Chinese)
季一木,朱曈晖,柴博周,等. 云环境下用户隐私混合加密方案及其性能分析[J]. 重庆邮电大学学报(自然科学版),2015,27(5):631-638

(上接第 190 页)

[2] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data [C]// CCS2006. Alexandria, Virginia: ACM, 2006: 89-98

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]// IEEE Symposium on Security and Privacy 2007. Berkeley, CA: IEEE, 2007: 321-334

[4] Hinek J, Jiang S, Safavi R, et al. Attribute-Based Encryption with Key Cloning Protection: Report 2008/478[R]. 2008

[5] Yu Shu-cheng, Ren Kui, Lou Wen-jing, et al. Defending Against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems[C]// Proceedings of the Security and Privacy in Communication Networks. Athens, Greece, 2009: 311-329

[6] Li Jin, Ren Kui, Zhu Bo, et al. Privacy-aware Attribute-based Encryption with User Accountability[C]// Proceedings of the Information Security Conference 2009. 2009: 347-362

[7] Wang Yong-tao, Chen Ke-fei, Chen Jian-hong. Attribute-Based Traitor Tracing[J]. Journal of Information Science and Engineering, 2011, 27(1): 181-195

[8] Ostrovsky R, Sahai A, Waters B. Attribute Based Encryption with Non-Monotonic Access Structures[C]// Proceedings of the 14th ACM Conference on Computer and Communication Security. Alexandria, New York, USA, 2007: 195-203

[9] Attrapadung N, Imai H. Conjunctive broadcast and attribute-based encryption[C]// Proceedings of the Pairing-Based Cryptography-Pairing 2009. Palo Alto, USA, 2009: 248-265

[10] Waters B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions[C]// Advances in Cryptology-CRYPTO 2009. Springer Berlin Heidelberg, 2009: 619-636

[11] Lewko A, Okamoto T, Sahai A, et al. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption[C]// Advances in Cryptology-EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 62-91

[12] Ma Hai-ying, Zeng Guo-sun. An Attribute-Based Encryption Scheme for Traitor Tracing and revocation together[J]. Chinese Journal of Computers, 2012, 35(9): 1845-1855(in Chinese)
马海英,曾国荪. 可追踪并撤销叛徒的属性基加密方案[J]. 计算机学报,2012,35(9):1845-1855

[13] Naor D, Naor M, Lotspiech J. Revocation and Tracing Schemes for Stateless Receivers[C]// Proceedings of the CRYPTO 2001. Santa Barbara, California, USA, 2001: 41-62

[14] Freeman M. Converting pairing-based cryptosystems from composite-order groups to prime-order groups[C]// EUROCRYPT 2010. Berlin: Springer, 2010: 44-61

[15] Lewko A. Tools for simulating features of composite order bilinear groups in the prime order setting[C]// EUROCRYPT-2012. Berlin: Springer, 2012: 318-335

[16] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]// CRYPTO 2001. Berlin: Springer, 2001: 213-229

[17] Lewko A, Waters B. Functional Encryption: New Proof Techniques and Advancing Capabilities[D]. The University of Texas at Austin, 2012

[18] Beimel A. Secure Schemes for Secret Sharing and Key Distribution[D]. Haifa, Israel: Israel Institute of Technology, Technion, 1996

[19] Feng Deng-guo, Chen Cheng. Research on Attribute-based Cryptography[J]. Journal of Cryptologic Research, 2014, 1(1): 1-12 (in Chinese)
冯登国,陈成. 属性密码学研究[J]. 密码学报,2014,1(1):1-12