PE文件隐型加壳技术的研究与实现*

徐向阳[1], 解庆春[1,2], 刘 勇[2], 俞 笛[1], 刘 寅[1]
(1.湖南大学 计算机与通信学院, 长沙 410082; 2.中国科学院 近代物理研究所,兰州 730000)

摘 要: 通过对 PE(portable executable)文件格式的了解,编写 PE分析工具对文件内部结构进行分析。详细介绍了对 PE可执行文件加壳的全过程,在此过程中巧妙地使用 MD5、CRC32等成熟的 hash算法及防 API断点跟踪等多种反破解技术,并采用自动隐藏加密方案,大大地提高了软件的保护力度。

关键词: 可移植的可执行文件; 加壳; 哈希算法; 反跟踪
中图分类号: TP309 +. 7 **文献标志码**: A **文章编号**: 1001-3695(2009)01-0337-02

Research and implementation of PE file stealthy shell technique

XU Xiang-yang[1], XIE Qing-chun[1,2], LIU-Yong[2], YU-Di[1], LIU-Yin[1]
(1. *College of Computer & Communication, Hunan University, Changsha* 410082, *China*; 2. *Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou* 730000, *China*)

Abstract: Wrote an analysis tool at the base of understanding the portable executable file format. It presented the whole processes of the encrypting on the PE file in detail. In these processes, used variety anti-crack techniques such as MD5, CRC32 algorithms of mature hash as well as anti-track from the interrupt of API functions, and also used some methods of automatic hiding and encrypting. The result is that the protection strength of the software is enhanced.

Key words: PE file; shell; hash algorithm; anti-track

众所周知,当今盗版软件问题很严重,侵犯了软件开发者的知识产权,对软件产业的健康发展造成了不利的影响。为了防止软件被非法复制、修改,除了用法律对产权进行保护之外也需要对产品本身从技术上进行保护。软件加壳技术是目前保护知识产权的一种有效方法。有软件加壳就有脱壳,有时候越有名的加密技术反而会更不安全,因为一个软件的加密程度越强,就会引来更多人对它进行研究,使其被破解的机会就越大。所以说软件开发者有必要根据自己需要,建立模型、提出自己的加壳保护方案,以达到软件在生命周期内被保护的目的。本文研究对象就是 Win 32平台下的 PE文件,笔者综合应用了当今优秀的加壳技术,如比较成熟的密码学算法和多种反破解方法,结合自己提出的独特的隐形加密方案实现,大大提高了软件的保护力度。

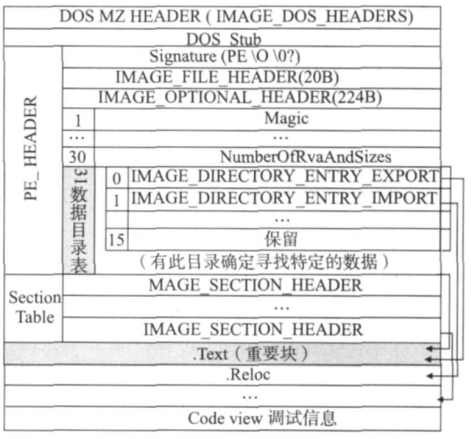


图1 PE文件内部结构总体框架图

文件结构

PE是 Win 32环境自身所带的执行文件格式。通过对 PE的深刻了解,根据对 PE各个部分之间的联系,得到 PE文件总体框架结构图[1,2],如图 1所示。PE各个模块的详细解释参见文献[1]。可以将一个 PE文件看成一本书,将每个 section看成书的内容。这本书有两个目录,第一个目录是块表,有块表就可以找到相应的块;第二个目录则是 data directory,这是一个类似于索引的目录,利用它可以很快找到相应的数据。要明确一点:一个块表只可以定位一个相应的块,而几个数据目录可以指向同一个块。

自动隐藏加密方案的设计和实现

加壳是加密的一种,本方案主要是介绍密码型加壳,这种加壳方案一般应用在共享软件的注册、对重要执行文件设置特定权限等领域。在运行带壳的程序时,首先会提示用户输入用户名、口令或者注册码。输入指令正确,外壳程序对内部加密的块表等还原之后定位到原程序并执行。如果破解者强行更改密码检测指令,会导致程序不正确地执行。因为被加密的代码并没有用相应的口令进行解码,块表没有被还原,指令就不能准确地定位。

本文设计的软件加壳方案主要由三部分组成,即 PE分析

收稿日期: 2008-04-14; **修回日期**: 2008-07-27 **基金项目**: 国家自然科学基金委员会重点资助项目(10635090)

作者简介:徐向阳(1973-),男,湖北麻城人,副教授,博士,主要研究方向为信息安全、自动化控制;解庆春(1982-),男,山东巨野人,硕士研究生,主要研究方向为网络与信息安全(xieqingchun@126.com);刘勇(1973-),男,湖北人,副研究员,博士,主要研究方向为加速器物理与技术;俞笛(1981-),男,湖南长沙人,硕士研究生,主要研究方向为网络与信息安全、数据挖掘;刘寅(1984-),湖南岳阳人,硕士研究生,主要研究方向为图形图像处理.

工具、装配程序设计和外壳程序设计。其中后面两部分是整个程序的核心[3]。

. 文件分析工具的设计和实现

在对 Window的 PE文件格式了解的基础上,可以根据 PE文件的结构写出分析工具。该分析工具可以显示加壳前后 PE文件的变化,可进一步了解加壳原理,并检验加壳成功与否。具体步骤请查看分析工具编写及实现[4]。

. 装配程序设计

首先装配程序就是对原程序在外壳程序嵌入到原程序之前,对原程序文件作一些记录和块的加密操作。其基本流程如图 2所示。

1)文件数据的读入 文件的读入方式采用内存映射文件的方式。如果文件被分析工具分析过,则可知程序已载入内存,直接读取。如果没有经过分析工具分析过,则需要先把文件载入内存。在文件读入过程中要注意两点: a)使用内存影射时对文件设置为可读、可写的权限; b)文件映射时映射的内存要比磁盘上的文件要大一些,以便在文件的尾部添加可执行的代码。

2)判断文件是否是 PE-EXE文件 可直接调用 PE分析工具中的文件格式的判断模块来处理。

3)自动隐藏加密 加壳文件的验证密码由用户输入获得,也应放在壳中,但是不能以明文存放。本文采用 MD5算法将密码加密。为了表述简单,用变量 HashPassword记录存储的固定长度的 MD5(用户名 &密码)作为 hash值。当然为了防止静态分析,可以用一些不太明显的名字记录此 hash值,如 abc。由于采用的是 hash算法,即使得到了密文,也不能够根据密文反推出明文,保证了密码的安全性。原程序的入口点,即 PE文件首部的 AddressOfEntryPoint字段的值必须要保存。入口点可以通过读取 PE文件结构的方法获取,入口点也应保存在外壳中。

为了更有效地防止动态跟踪。对程序的入口点进行加密。采取了对 HashPassword二次利用的方法,利用 VC ++的一个函数 CString SubString(int *a*, int *b*)取 128 bit hash值的一部分(*a*, *b*值可以在界面自己设置)来当做第二次加密的密钥。过程伪代码如下:

```
密文 = Encrypt(入口点, HashPassword. substring(a, b));
//用输入的密码作为密钥对程序的入口点进行加密
入口点 = Decrypt(密文, HashPassword. substring(a, b));
//当输入的密码正确时,用密码作为密钥对密文进行解密
```

这里的加密算法选用一般的对称加密。在壳程序中,直接使用语句“Jump to Decrypt(密文,输入密码 (*a*, *b*))”来代替“Jump to Entrypoint”这条指向真正入口点语句的调用。由于在程序中没有明文的密钥,该方法称之为自动隐藏加密。

该方法实现一个密码执行两次加密,更有效地防止破解者在找不到用户名和密码的情况下,应用修改跳转指令而获得成功的可能性。而这样入口点和密码紧密地结合在一起,无论修改哪一个值,程序都无法正常地运行,大大地增强了加密强度。

4)对导入表等所在的块 (一般为. text)进行加密 对块加密的目的是为了防止通过分析工具查找到程序入口点并对其直接修改,有效地防止程序被非法修改。

5)添加新的 section并对块表进行修改、对齐 把外壳组织为一个块的形式且新块要考虑文件对齐。

6)修改文件头信息 原有的文件头信息需要保存在外壳程序当中,如要保存原来程序入口的 RVA,设置新的入口地址,修改 NumberOfSections, SizeOfImage等。

7)构造外壳程序的导入表 外壳程序是被装配上去的,而不是由程序直接编译链接的,相应的导入表不会自己生成。所以外壳程序的导入表就必须人工构造。可以在外壳程序中使用任何 API调用,以丰富外壳程序的功能[3]。

8)调用 CRC32算法 将头信息和外壳程序写入目标文件。

外壳程序设计

外壳程序就是加密软件将一段加密代码附加到执行程序上,并把程序入口指向附加的代码中。但被加密的程序装入内存之后,附加代码首先执行。具体的外壳程序的执行过程如图 3所示。

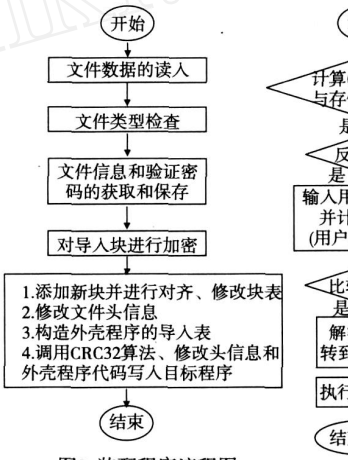


图2 装配程序流程图

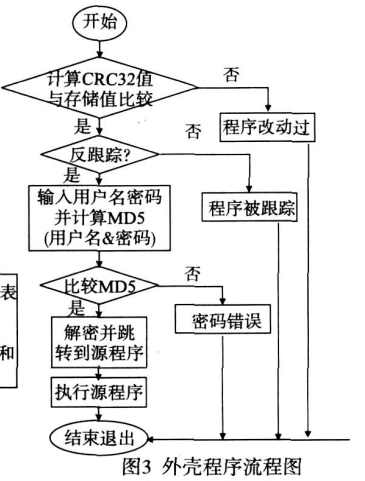


图3 外壳程序流程图

1) CRC校验 最后外壳程序要执行 CRC32,计算文件的校验值,进一步来检查、判断程序是否被修改过。如果为否,程序按流程图顺序执行。

2)反跟踪 检查内存中是否存在静态分析技术代码指令和动态调试软件的跟踪。根据当今 cracker应用较多的破解技术,本文采用了三种方法。该技术在程序加壳中起着举足轻重的作用,主要包括防静态汇编、防调试器、防监视等[5]。

3)合法性验证 外壳程序必须对密码是否正确进行验证,由于密码已经用 MD5加密后放入壳中,外壳程序也必须对输入的密码进行 MD5的 hash运算。如果经 hash后的值与壳中保存的密文一致,那么说明密码正确,加壳后程序会正常执行;否则,程序直接退出。

4)解密还原块 通过合法性检查后,现在要做的就是对源程序的一些块进行解密还原。装配时已经保存了被加密的块数、块的长度起始 RVA地址。由被加密块的 RVA地址就可以找到该块的密文在内存中的首址。再使用装配加密时的口令并根据保存的该块的长度进行解密。

5)转入源程序执行 在完成了对源程序一些关键块的解密还原和 import表的人工填写后,源程序已经完全还原。将保存的源程序的入口 RVA加上装入基址,就是源程序的实际入口地址,使用一条“Jump to Decrypt(密文,输入密码 (*a*, *b*)) + ImageBase;”指令跳入源程序执行。

实验结果

实验用的系统是 Windows XP,可执行文件选用的是操作系统自带的记事本程序 notepad. exe。 (下转第 341页)

量超过一定限度时,会向邻居节点发出建立防御联盟的请求。以下是实验分析结果。

1)无防御下性能分析　笔者分析了无防御情况下系统所有节点平均性能与受害节点性能的对比情况,定义平均性能(avgPerformance)如下:

avgPerformance =有效下载流量 /当前周期系统总流量

在没有任何防御措施情况下,所有攻击流量均由受害节点处理,运行一定周期后,性能如图 1所示。从图中曲线可以看到,随着 DDoS攻击流量的加大,受害节点的性能急剧下降,在第 110周期时,性能已经接近 0;而系统其他节点的性能也会有所下降,原因是受害节点无法为其他节点提供正常服务。可以看出,随着系统的运行,受害节点与其他节点之间的性能差距越来越大,最终导致受害节点因性能不足而被边缘化,使得攻击者攻击成功。

2)AntDA模型性能分析　加入防御联盟后,如图 2所示。在 DDoS流量增加的初始阶段,如第 200周期左右,受害节点的性能有一个明显下降的过程,此时,受害节点还没有及时形成防御联盟。在形成防御联盟后,由于攻击流量还处于相对可以承受的程度,受害节点会应用策略 a),图中可以看出,受害节点的性能逐渐得到回升。随着 DDoS攻击流量继续加大,在第 212周期时,受害节点性能达到最低,应用策略 b)后,受害节点性能迅速回升。

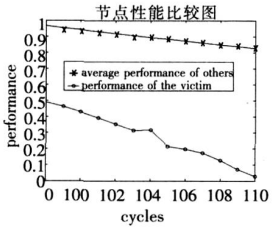


图 1　无防御下性能分析图

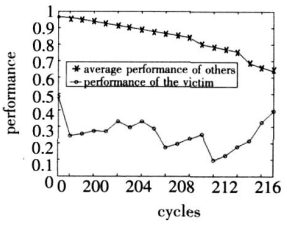


图 2　AntDA 下性能分析图

对于其他节点,在应用策略 b)之前,平均性能有一个平缓下降的过程,原因是受害节点无法提供正常服务;在应用策略 b)后,由于这些节点中的一部分作为受害节点的联盟成员,需要参与 DDoS攻击数据包的截获工作,因而,平均性能下降幅度加大。然而不久,平均性能下降趋于平缓,原因是随着受害节点性能的回升,它可以为其他节点提供正常服务了。从图 2可以看到,随着系统的运行,受害节点与其他节点在性能上的差距逐渐缩小,系统逐渐趋于平衡。

结束语

本文结合 P2P网络的特点,提出一种基于防御联盟的 DDoS攻击防范模型,模型充分考虑 P2P网络小世界特性以及节点理性等拓扑特征,采用蚁群智能算法选择联盟成员,有效避免了全局拓扑信息的计算。实验证明,该方案可以在系统性能不受太大影响的情况下有效防范 DDoS攻击。下一步的工作是在参数选择上增加灵活性,使系统能应对各种攻击现象,增强系统的鲁棒性。

参考文献:

[1] LIANG J, NAOUMOV N, ROSS K W. The index poisoning attack in P2P file sharing systems[C]//Proc of the 25th IEEE International Conference on Computer Communications. Barcelona: [s. n.], 2006: 1-12.

[2] MA Xin-xin, ZHAO Yang, QING Zhi-guang. Improving resilience against DDoS attack in unstructured P2P networks[J]. Journal of Electronic Science and Technology of China, 2007, 5(1): 18-22, 28.

[3] MARCO D. Ant colony optimization and swarm intelligence[M]. Brussels: Springer, 2006: 100-109.

[4] ELKE M, ARNO P, GERTI K. Using taxonomies for content-based routing with ants[J]. Journal of Computer Networks, 2007, 51(16): 4514-4528.

[5] 田慧蓉,邹仕洪,王文东,等. 激励一致的自适应 P2P拓扑构造[J]. 软件学报, 2006, 17(4): 845-853.

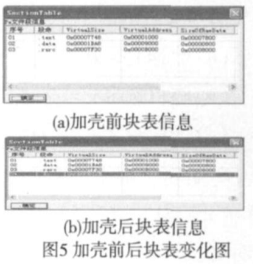
[6] KAMVAR S D, SCHLOSSER M T, GARCIA M H. The eigentrust algorithm for reputation management in P2P networks[C]//Proc of the 12th International Conference on World Wide Web. New York: ACM Press, 2003: 640-651.

(上接第 338页)

正确性测试:

首先打开要加壳的文件,运行 PE分析工具模块,图 4给出文件的相关信息。图 5(a)显示了文件加壳之前块表的信息;(b)显示了文件加壳之后块表的信息,发现文件多了一个新的块. Xy(是自定义的新块名字),说明加块成功。执行加壳后的程序,除多了一道弹出对话框让用户输入密码,检验口令正确与否流程之外,几乎与加壳前没有区别。说明外壳已完整地嵌入到源程序,验证了隐形加壳方法编写程序的正确性。


图4 PE分析图


(a)加壳前块表信息

(b)加壳后块表信息

图5 加壳前后块表变化图

结束语

本文对 PE文件结构及 Windows底层运作机制进行了深刻剖析,通过自己设计的自动隐藏的加密方案即实现用密码 Hash值对程序入口点加密,防止分析工具直接修改跳转指令来达到破解的目的,大大提高了程序的反静态分析能力。还巧妙地使用 MD5、CRC32等成熟的 hash算法及防 API断点跟踪等多种反破解技术,大大提高了加壳软件的反跟踪能力。

参考文献:

[1] Microsoft portable executable and common object file format specification revision 6.0[K]. USA: Microsoft Corporation, 1999.

[2] 刘晓冬. 软件加壳技术的研究与实现[D]. 沈阳:沈阳工业大学, 2006.

[3] 张建明. PE可执行文件通用加密工具的设计与实现[J]. 计算机系统应用, 2004(8): 21-24.

[4] 看雪学院. 软件加密技术内幕[M]. 北京:电子工业出版社, 2004.

[5] SMOKINGROOM. Delphi程序编程语言之常用反跟踪技术[EB/OL]. (2005-04-12)[2006-08-15]. http://industry.ccidnet.com/art/1077/20051202/809153_1.html

[6] 段钢. 加密与解密[M]. 2版. 北京:电子工业出版社, 2003.