信息科学技术学术著作丛书

恶意软件分析与检测

王俊峰　汪晓庆　张小松　苏璞睿　著

科学出版社

信息科学技术学术著作丛书

恶意软件分析与检测

王俊峰　汪晓庆　张小松　苏璞睿　著

科学出版社

北京

内 容 简 介

本书系统介绍了恶意软件分析的理论与研究现状，重点介绍了新型恶意软件检测方法中的各类模型和关键技术，内容涵盖恶意软件分析与检测所需的基础知识、软件加壳及检测技术、基于机器学习的恶意软件静态检测方法以及恶意软件动态检测方法等内容。本书对恶意软件检测方法的构造原理、实施过程、实验环境和检测性能等多方面进行了全面的分析，以便读者能够更加深刻地理解这些方法的实现原理与应用特点。本书部分反映了当前恶意软件分析领域的最新研究成果，并提供了详尽的参考文献。

本书结构清晰，内容丰富，论述详细，可作为信息安全或计算机相关专业研究生和高年级本科生相关课程的教材，也可供恶意软件分析领域的研发和管理人员参考。

图书在版编目(CIP)数据

恶意软件分析与检测/王俊峰等著. —北京：科学出版社，2017.2
(信息科学技术学术著作丛书)
ISBN 978-7-03-051300-7

Ⅰ.①恶… Ⅱ.①王… Ⅲ.①计算机网络-安全技术 Ⅳ.①TP393.08

中国版本图书馆 CIP 数据核字(2016)第 314833 号

责任编辑：张海娜 纪四稳 / 责任校对：郭瑞芝
责任印制：张 倩 / 封面设计：左讯科技

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

三河市骏杰印刷有限公司 印刷
科学出版社发行 各地新华书店经销
*
2017 年 2 月第 一 版 开本：720×1000 1/16
2017 年 2 月第一次印刷 印张：19 1/2
字数：390 000

定价：98.00 元

(如有印装质量问题，我社负责调换)

《信息科学技术学术著作丛书》序

21世纪是信息科学技术发生深刻变革的时代，一场以网络科学、高性能计算和仿真、智能科学、计算思维为特征的信息科学革命正在兴起。信息科学技术正在逐步融入各个应用领域并与生物、纳米、认知等交织在一起，悄然改变着我们的生活方式。信息科学技术已经成为人类社会进步过程中发展最快、交叉渗透性最强、应用面最广的关键技术。

如何进一步推动我国信息科学技术的研究与发展；如何将信息技术发展的新理论、新方法与研究成果转化为社会发展的新动力；如何抓住信息技术深刻发展变革的机遇，提升我国自主创新和可持续发展的能力？这些问题的解答都离不开我国科技工作者和工程技术人员的求索和艰辛付出。为这些科技工作者和工程技术人员提供一个良好的出版环境和平台，将这些科技成就迅速转化为智力成果，将对我国信息科学技术的发展起到重要的推动作用。

《信息科学技术学术著作丛书》是科学出版社在广泛征求专家意见的基础上，经过长期考察、反复论证之后组织出版的。这套丛书旨在传播网络科学和未来网络技术，微电子、光电子和量子信息技术、超级计算机、软件和信息存储技术，数据知识化和基于知识处理的未来信息服务业，低成本信息化和用信息技术提升传统产业，智能与认知科学、生物信息学、社会信息学等前沿交叉科学，信息科学基础理论，信息安全等几个未来信息科学技术重点发展领域的优秀科研成果。丛书力争起点高、内容新、导向性强，具有一定的原创性；体现出科学出版社“高层次、高质量、高水平”的特色和“严肃、严密、严格”的优良作风。

希望这套丛书的出版，能为我国信息科学技术的发展、创新和突破带来一些启迪和帮助。同时，欢迎广大读者提出好的建议，以促进和完善丛书的出版工作。

中国工程院院士
原中国科学院计算技术研究所所长
李国杰

前　言

恶意软件是指人为设计、内含安全风险的计算机软件，它们运行时会威胁到信息系统中程序或数据的机密性、完整性、可用性和可控性等安全特征。恶意软件研究是信息安全研究领域的重要内容之一，在维护国防安全、保障国民经济以及保护个人隐私等方面也起到了重要的作用。

随着软件在数据产生、交换以及处理等过程中起到日益重要的作用，新的恶意软件也不断地被开发，同时原有恶意软件的变种也层出不穷。卡巴斯基实验室公布的报告显示，实验室在 2015 年第三季度共发现 38233047 种恶意软件，包括脚本、网页、Exploits 和可执行程序等形式，恶意软件已经被看成与互联网软件生态系统长期共存的一部分。2016 年 7 月发布的《第十五次全国信息网络安全状况暨计算机和移动终端病毒疫情调查结果》显示：2015 年我国计算机用户的病毒/木马感染率高达 60.5%，比 2014 年增长了 9.1%。

面对恶意软件的常态化及其破坏性，计算机用户不得不投入更多的成本来维护系统安全；同时，信息安全人员也希望通过对恶意软件的工作原理和行为规律进行更深入的认识和研究，开发出新的理论和技术以应对恶意代码分析、检测和抑制中出现的新问题。

恶意软件检测技术是计算机代码安全性的判定方法，它是恶意软件研究的重要组成部分。许多恶意软件分析技术，如恶意代码结构分析、功能分析和防御技术等，都是以检测技术为前提的。检测方法先进性和完备性在很大程度上决定了其他恶意软件分析方法的有效性。

尽管传统基于特征码的恶意软件检测方法在工业领域有着广泛的影响，但在应付不断涌现出的恶意软件时容易表现出响应不够及时、分析成本过高等缺陷，尤其是在面对 zero-day 恶意软件时，漏报率会大大增加。而一些部署在计算机系统上的启发式动态检测工具，其检测规则由系统或用户自行定义，不但增加了系统负载，而且无法适应多变的用户需求，经常会出现误报的情况。

针对当前恶意软件攻击手段更加多样性、制作技术更加复杂以及隐蔽性更强等特点，研究新型的恶意软件检测技术成为保障信息系统安全的当务之急，此类技术在借鉴传统检测方法的基础上，主要突出以下特点：

(1) 高效的分析处理能力。能够克服传统人工分析方法中分析速度慢、成本高等缺点，检测模型需要保证自动或少量人工干预情况下，完成对软件信息尤其是恶意软件特有的信息进行快速提取、识别和重构等过程。

(2) 更加广泛的适应性。恶意软件采用加壳、多态或变形等隐藏技术时,部分代码会经常发生变化,很难使用固定的特征值来描述。新型检测方法应该更加注重软件内部比较稳定的特征,或者使用多类型特征抽象出恶意软件特有的结构或行为。

(3) 识别 zero-day 恶意软件。zero-day 恶意软件的特征很少完全出现在原有的特征库中,新型检测方法需要利用统计或数据挖掘等方式建立更广谱的检测规则,识别这类恶意软件的新特征,同时还需要保持较低的误报率。

本书结合当前国内外恶意软件领域的研究现状,总结作者在恶意软件研究领域近年来的研究成果。主要介绍并讨论当前主流操作系统平台上的新型恶意软件检测技术,结合各种类型的软件结构和运行机制,重点分析恶意软件特征的不同抽象表示方法,基于机器学习和数据融合中经典算法提出多种恶意软件检测方法,并且利用真实的软件样本进行实验,详细分析检测性能的各项数据指标。

本书共 15 章,具体内容如下:第 1 章介绍 Windows、Linux 和 Android 操作系统中可执行文件的内部结构格式以及运行机制;第 2 章主要介绍恶意软件抽象理论和基于机器学习的恶意软件检测所需的算法和工具软件;第 3 章详细讨论 x86 平台下的常用加壳技术和代码保护技术;第 4 章阐述基于机器学习的加壳检测框架;第 5 章研究基于函数调用图签名的恶意软件检测方法;第 6 章重点介绍基于挖掘格式信息的恶意软件检测方法;第 7 章重点介绍基于控制流结构体的恶意软件检测方法;第 8 章讨论基于控制流图特征的恶意软件检测方法;第 9 章介绍软件局部恶意代码识别通用方法;第 10 章介绍基于多视集成学习的恶意软件检测方法;第 11 章介绍基于动态变长 Native API 序列的恶意软件检测方法;第 12 章介绍基于多特征的移动设备恶意代码检测方法;第 13 章介绍基于实际使用的权限组合与系统 API 的恶意软件检测方法;第 14 章介绍基于敏感权限及其函数调用图的恶意软件检测方法;第 15 章介绍基于频繁子图挖掘的异常入侵检测新方法。

本书相关研究工作得到了国家科技重大专项项目(2015ZX01040101)、国家重点研发计划项目(2016YFB0800605)和装备预研教育部联合基金(青年人才基金)等的大力资助;课题组的赵宗渠博士、白金荣博士、丁雪峰博士、刘辉硕士、祝小兰硕士,博士研究生杜垚、吴鹏、方智阳、袁保国,硕士研究生郭文、肖锦琦、刘留等在分析技术的研究和书稿的撰写中付出了大量心血,在此一并致谢。

软件技术的发展日新月异,恶意软件的规模越来越大,所涉及的专业技术日趋复杂,对其进行准确高效的分析与检测难度极大,由于作者学识及经验有限,书中难免会有疏漏或不当之处,恳请广大读者批评指正。

目　录

第 1 章　二进制可执行文件简介

可执行文件作为操作系统最重要的文件类型之一，是功能操作的真正执行者。操作系统支持的可执行文件格式与操作系统的文件加载机制密切相关，不同的操作系统支持不同格式的可执行文件，而可执行文件的格式决定了可执行文件的大小、运行速度、资源占用、扩展性、移植性等文件的重要特性。

对可执行文件结构及相关技术的研究是病毒研究的基础，因为病毒程序的执行必将直接或者间接地依赖于可执行文件。本章简要介绍 Windows 操作系统、Linux 操作系统、Android 操作系统的可执行文件格式及相关技术。

1.1　Windows PE 文件

微软 Win32 环境可执行文件的标准格式是 PE(portable executable)文件，其目标是为所有 Windows 平台设计统一的文件格式，即为 Windows 平台的应用软件提供良好的兼容性、扩展性[1]。微软自 Windows NT3.1 首次引入 PE 文件格式以来，后续操作系统结构变化、新特性添加、文件存储格式转换等都没有影响 PE 文件格式。Window 下常见的 EXE、DLL、OCX、SYS、COM 等多种文件类型都属于 PE 文件格式。

1.1.1　PE 文件结构

PE 文件结构是 Windows 操作系统管理可执行文件的代码、数据及其相关资源的数据组织方式。从资源存储角度来看，PE 文件使用一个平面地址空间，将所有代码、数据合并在一起组成一个大结构。从文件结构来看，PE 文件由文件头部和文件体组成，文件头部包含文件的结构、属性等信息，如该文件支持的操作系统、程序的入口地址等；文件体包含代码、数据及相关资源等。

为了管理可执行文件的资源，PE 文件从两个维度来对数据资源进行规范管理：其一是数据类型，主要按照数据功能对数据进行集中管理，如将数据资源划分为导入表、资源表等；其二是数据属性，不同的数据需要不同的访问权限，即代码的只读、只写、可读、可写等属性，如代码数据在运行时不允许修改，而数据段数据允许读写等。如何方便地查找资源、实现资源定位是资源管理面临的另一个重要问题。PE 文件数据资源定位采用链表与固定格式相结合的方式，前者利用链表管理资源，资源的具体位置灵活，后者要求数据结构大小固定，其位置也相对固定。

典型的 PE 文件结构如图 1.1 所示。

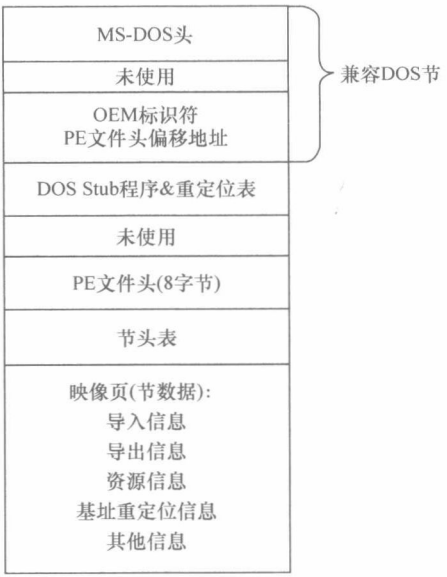

图 1.1 PE 文件结构

1. MS-DOS 结构

PE 文件引入初期，为兼容 DOS 操作系统，设计了 4 字节的 MS-DOS 可执行文件体。PE 文件 MS-DOS 头部的数据结构如下：

```
_IMAGE_DOS_HEADER{
    WORD e_magic;//魔幻数,EXE 文件标志"MZ"
    ......
    LONG e_lfanew;//PE 头文件的文件相对地址
}IMAGE_DOS_HEADER;
```

e_magic(魔幻数)字段是 DOS 文件的标志，其值固定为 0x5A4D，对应的字符串为"MZ"，是 DOS 操作系统设计者的名字缩写；

e_lfanew 字段是一个 4 字节的文件偏移量，代表 PE 文件头的偏移地址，PE 文件头部利用此字段来定位。

2. PE 结构

PE 结构是 PE 文件的主体，实现对 PE 文件所有代码、数据信息的存储、管理，在 PE 文件布局中 DOS 节以外的部分都属于 PE 结构。

PE 文件采用节作为存放代码或数据的基本单元，PE 文件常用的节有代码节

(.text)、数据节(.data)、资源节(.rsrc)等。节是一个没有大小限制的连续结构,可以存放不同的数据类型(如代码、数据、资源等),每一个节都有独立的访问权限,文件的内容被分割为不同的节,各节按页边界来对齐,相同属性的文件内容必须放到同一个节中,节的名字仅用于节标识。

节的管理使用节表技术实现,节表是节的索引目录,包含节名及节内容的位置指针,节表内部按照顺序依次对齐排列,节表中节内容的位置指针指向真实的节。节表总尺寸=每个节表的尺寸×节的数目,其中每个节的描述信息长度固定 40 个字节,而节的个数是不确定的,最小取值为 1,且节的数目必须与 PE 文件头中 IMAGE_FILE_HEADER 数据结构中的 NumberOfSections 一致。

1.1.2　PE 文件头结构

PE 文件头结构是 PE 文件最重要的结构之一,由 PE 头标识(Signature)、标准 PE 头(IMAGE_FILE_HEADER)和扩展 PE 头(IMAGE_OPTIONAL_HEADER32)三部分组成[1]。从数据结构的角度来看,PE 文件结构如图 1.2 所示。

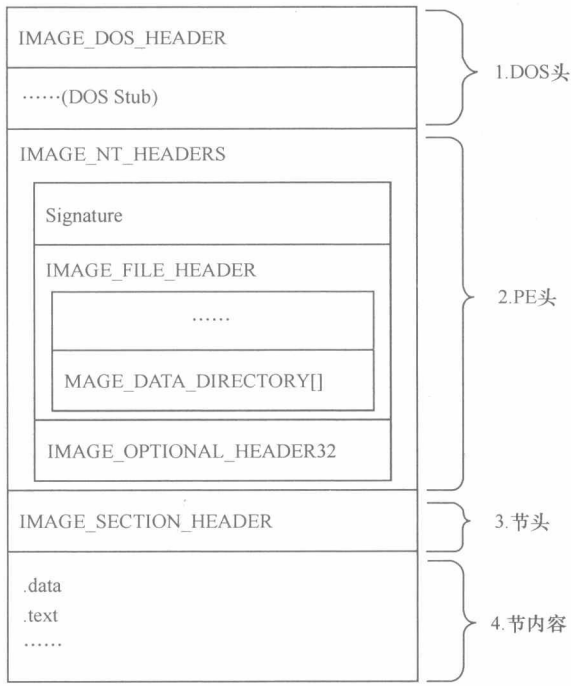

图 1.2　PE 文件结构图(数据结构视角)

1. IMAGE_NT_HEADERS 数据结构

```
_IMAGE_NT_HEADERS{
    DWORD Signature;//PE 头标识
    IMAGE_FILE_HEADER FileHeader;//标准 PE 头
    IMAGE_OPTIONAL_HEADER32 OptionalHeader;//扩展 PE 头(32位 CPU)
}IMAGE_NT_HEADERS32;
```

Signature 字段与 MS-DOS 头中的魔幻数(e_magic)功能类似,是 PE 文件的标识,该字段的高 16 位为 0,低 16 位为固定的值 0x4550,对应的字符为“PE”,完整的 Signature 值为“PE\0\0”。

IMAGE_FILE_HEADER 类型的 FileHeader 字段代表 PE 文件基本信息和属性,即本字段描述了 PE 文件的概貌。PE 文件加载时会用本字段的属性来检查当前的运行环境,如果不一致则会终止加载 PE 文件。

IMAGE_OPTIONAL_HEADER32 类型的 OptionalHeader 字段包含 PE 文件基本信息外的其他信息(如代码段、数据段的基准位置、数据目录等),本字段在 Object 文件(编译中间文件)无具体内容。

对 PE 文件头的数据结构有了整体认识后,接下来详细介绍 PE 文件头中的具体细节,主要是 PE 文件头中两个主要的数据结构 IMAGE_FILE_HEADER、IMAGE_OPTIONAL_HEADER32 的详细信息。

说明:数据结构均来自 Microsoft 官方的 WINNT. H,本章只给出重要的字段,针对各字段侧重于功能说明,具体的字段取值信息请参考官方手册[2]。

2. 标准 PE 头(IMAGE_FILE_HEADER)

标准 PE 头的概要数据结构如下:

```
_IMAGE_FILE_HEADER{
    WORD Machine;//运行平台
    WORD NumberOfSections;//文件节的数目
    ……
    WORD Characteristics;//文件属性
}IMAGE_FILE_HEADER;
```

Machine 字段代表 PE 文件支持的 CPU 类型,PE 文件的设计初衷是兼容主流的 CPU 型号。不同 CPU 的指令集是不相同的,不同平台编译的可执行文件依赖的运行硬件环境由此字段确定,如此字段取值为 0x14C,代表 PE 文件支持的 CPU 类型是 Intel 386 及其后续兼容处理器。

NumberOfSections 字段记录当前 PE 文件中节的数目,必须与节表中节的数