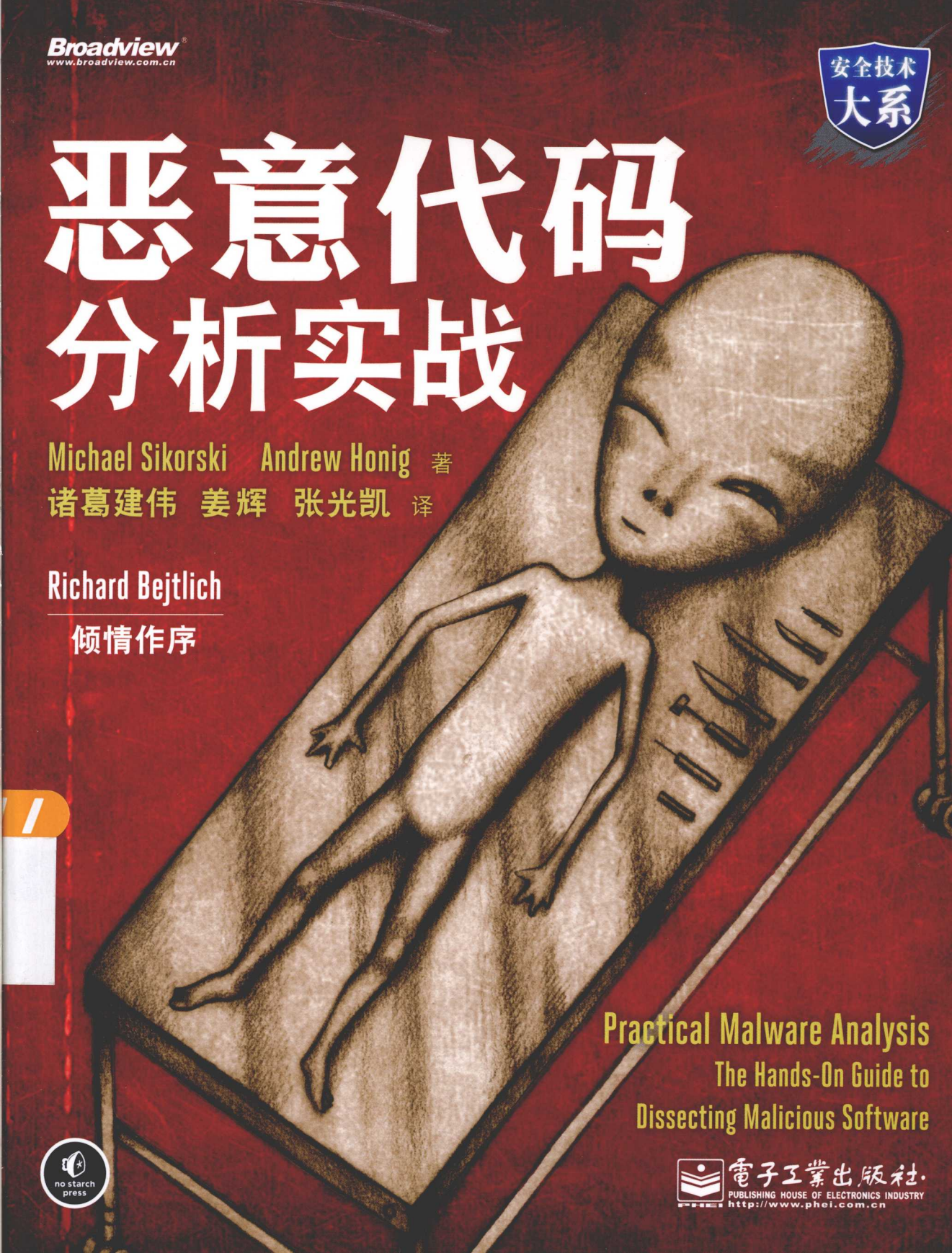
Broadview
www.broadview.com.cn

安全技术大系

恶意代码分析实战

Michael Sikorski Andrew Honig 著
诸葛建伟 姜辉 张光凯 译

Richard Bejtlich
倾情作序



Practical Malware Analysis
The Hands-On Guide to
Dissecting Malicious Software

no starch press

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Practical Malware Analysis

The Hands-On Guide to Dissecting Malicious Software

恶意代码分析实战

Michael Sikorski　Andrew Honig 著
诸葛建伟　姜辉　张光凯 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书是一本内容全面的恶意代码分析技术指南，其内容兼顾理论，重在实践，从不同方面为读者讲解恶意代码分析的实用技术方法。

本书分为 21 章，覆盖恶意代码行为、恶意代码静态分析方法、恶意代码动态分析方法、恶意代码对抗与反对抗方法等，并包含了 shellcode 分析，C++恶意代码分析，以及 64 位恶意代码分析方法的介绍。本书多个章节后面都配有实验并配有实验的详细讲解与分析。通过每章的介绍及章后的实验，本书一步一个台阶地帮助初学者从零开始建立起恶意代码分析的基本技能。

本书获得业界的一致好评，IDA Pro 的作者 Ilfak Guilfanov 这样评价本书：“一本恶意代码分析的实践入门指南，我把这本书推荐给所有希望解剖 Windows 恶意代码的读者”。

本书的读者群主要是网络与系统安全领域的技术爱好者与学生及恶意代码分析研究方面的安全从业人员。

版权贸易合同登记号　图字：01-2013-4025

图书在版编目（CIP）数据

恶意代码分析实战 /(美)斯科尔斯基(Sikorski,M.)，(美)哈尼戈(Honig,A.)著；诸葛建伟，姜辉，张光凯译. —北京：电子工业出版社，2014.4
（安全技术大系）
书名原文：Practical Malware Analysis:The Hands-On Guide to Dissecting Malicious Software
ISBN 978-7-121-22468-3

Ⅰ. ①恶… Ⅱ. ①斯… ②哈… ③诸… ④姜… ⑤张… Ⅲ. ①电子计算机－安全技术－码 Ⅳ. ①TP309

中国版本图书馆 CIP 数据核字（2014）第 026844 号

策划编辑：刘　皎
责任编辑：贾　莉
印　　刷：北京中新伟业印刷有限公司
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　　印张：45.75　　字数：1134 千字
印　　次：2014 年 4 月第 1 次印刷
印　　数：3000 册　　定价：128.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。
服务热线：（010）88258888。

好评袭来

一本优秀的恶意代码速成教程。

Dino Dai Zovi，自由安全咨询师

一本最为全面的恶意代码分析技术指南，覆盖了所有恶意代码分析所需的关键技术，帮助你理解最新恶意代码带来的技术挑战。

Chris Eagle，美国海军研究生院，计算机科学系高级讲师

一本恶意代码分析的实践入门指南，我把这本书推荐给所有希望解剖Windows恶意代码的读者。

— Ilfak Guilfanov，IDA PRO创始人

一本恶意代码分析的优秀指南，所有章节都包含有详细的技术解释和动手实践案例教程，让你能够立即面对真实的恶意代码。

— Sebastian Porst，Google软件工程师

本书为各个技术层次上的读者带来逆向工程方法，提供了一大堆覆盖各种技术并且容易访问的实践案例，将带领你更加深入地理解逆向工程的艺术与科学。我强烈推荐本书，无论是新手还是领域专家。

— Danny Quist，博士，OFFENSIVE COMPUTING创始人

如果你只要读一本恶意代码的书籍，或是希望进入到恶意代码分析师的世界，就来看这本书吧！

— Patrick Engbretson，达科他州立大学教授，*The Basics of Hacking and Pen Testing*一书的作者

为软件安全或入侵检测系统等研究生级别高级课程提供很好的课程资料，实践案例对学生学习逆向工程方法、恶意代码分析和理解等方面具有特殊的价值。

— Sal Stolfo，哥伦比亚大学教授

警　告

这是一本关于恶意代码的书籍，本书中描述的所有链接和软件都可能是恶意的。在执行任何未知代码和访问不可信URL时请务必保持极度的小心。

请参考第2章，以创建用于恶意代码分析的安全虚拟环境。

不要犯傻，保护你的计算环境安全。

关于作者

迈克尔·斯科尔斯基（Michael Sikorski）是在Mandiant公司任职的计算机安全顾问。他对恶意代码进行逆向分析，支持公司安全事件响应调查，并为公司的联邦政府部门客户提供专业研究与开发的安全解决方案。迈克创建了一套恶意代码分析的系列课程，并对不同的受众进行培训，包括联邦调查局和Black Hat黑客大会参会者。在加入Mandiant公司之前，他在麻省理工学院林肯实验室工作，在那里他对被动网络映射技术和渗透测试进行研究。迈克也是美国国家安全局为期三年的系统和网络跨学科研究生项目的毕业生。在美国国家安全局期间，他为逆向工程方面的研究做出了贡献，并获得在网络分析方面的多项发明奖项。

安德鲁·哈尼克（Andrew Honig）是美国国防部的一位信息保障专家。他在国家密码学院（National Cryptologic School）教授软件分析、逆向工程和Windows系统编程的课程，并是一位CISSP安全认证专家。安德鲁是VMware虚拟化产品几个零日漏洞的发现者，并开发了一些可以检测新型恶意代码（包括内核套件）的创新工具。作为一位精于恶意代码与良性软件分析技术的专家，他已经拥有超过10年的计算机安全业界分析师的经验。

技术编辑

斯蒂芬·劳勒（Stephen Lawler）是一家小型计算机软件和安全咨询公司的创始人和总裁。斯蒂芬一直在信息安全领域活跃工作了超过七年时间。主要涉及逆向工程、恶意代码分析和安全漏洞研究。他是Mandiant恶意代码分析团队的一位成员，协助调查分析了影响福布斯全球100强企业的几个高价值计算机入侵事件。在此之前，他曾在ManTech国际公司的安全与任务保证部门（SMA）工作，在那里，作为软件确保工作的职责内容，他发现了许多零日安全漏洞和软件利用技术。而在之前，他的生活与计算机安全没有关联，曾是美国海军SMMTT计划中声纳模拟部件的首席软件开发工程师。

贡献作者

尼克·哈波尔（Nick Harbour）是Mandiant公司的一位恶意代码分析师和逆向工程业务经验丰富的老将。他在信息安全领域有着13年的职业生涯，首先是作为国防部计算机取证实验室的一位计算

机取证分析师和研究员，在过去6年中，尼克都任职于Mandiant公司，主要工作内容都集中在恶意代码分析上。他是反逆向工程技术领域的一线研究员，编写了好几个加壳和代码混淆工具，如PE-Scrambler等。他已经在Black Hat和Defcon黑客大会做过多次关于反逆向工程与反取证技术的演讲。他也是Black Hat黑客大会高级恶意代码分析课程的主要开发者与培训师。

林赛·莱克（Lindsey Lack）是Mandiant公司一位有着超过12年信息安全工作经验的技术总监，精于恶意代码逆向工程、网络防御和安全运营等技术。他曾帮助创建和运营了一个安全操作中心（SOC），并领导网络防御方面的研究工作，开发了安全托管解决方案。他曾在国家信息安全保障研究实验室、总统执行办公室、Cable and Wireless公司，以及美国军队中任职。除了斯坦福大学计算机科学学士学位之外，林赛也获得了美国海军研究生院信息安全保障专业方向的硕士学位。

杰罗尔德·“杰”·史密斯（Jerrold “Jay” Smith）是Mandiant公司的首席顾问，他擅长恶意代码逆向工程与取证分析。在这个角色上，他已经在来自于福布斯500强企业许多客户的入侵响应事件中做出了贡献。在加盟Mandiant公司之前，杰曾经在美国国家安全局工作，但他不被允许谈论这段经历。杰持有加州大学伯克利分校的电子工程与计算机科学学士学位，以及约翰·霍普金斯大学（JHU）的计算机科学硕士学位。

推荐序

没有几个数字安全的领域，像本书讨论的恶意代码方向这么不对称。

2011年夏天，我参加了在内华达州拉斯维加斯召开的黑客大会，聆听了Peiter（麦基）Zatko的主题演讲。麦基在他的演讲中，引入了现代软件中本质存在的不对称性。他解释了他如何分析了9 000个恶意代码的二进制文件样本，而他的样本集平均源代码行数为125行（LOC）。

你可能认为麦基的样本集中只包含了那些“简单”或是“寻常”的恶意代码。你也许会问，那些真正的“武器”会有多大规模呢？比如说（能够让你屏住呼吸的）——Stuxnet病毒？根据Larry L. Constantine的说法[1]，Stuxnet病毒包含了约15 000行源代码，因此是样本集平均125行代码的120倍。Stuxnet病毒是一个高度专业性和目标性的“战争武器”，大概可以理解为其代码行数高于平均大小的原因。

我们稍微离开一会儿恶意代码的世界，来看看我所使用的文本编辑器（gedit，GNOME中的文本编辑器），其中包含了295行源代码的gedit.c，而gedit.c仅仅是其中128个源文件（以及其他3个目录）中的1个。[2] GNOME GIT源代码库经过统计之后，总共有128个文件和3个目录，共计70 484行源代码。合法程序的源代码数量与恶意代码的比值超过了500：1。与一个如文本编辑器这样相当简单的工具相比，恶意代码样本从平均源码行数上看起来似乎非常高效！

麦基的恶意代码样本集平均125行源代码对我来说似乎有点低，因为不同类型“恶意代码”定义的存在。很多恶意应用程序是一类“套件”，拥有许多功能和基础设施元素。为了看看这类恶意代码的大致规模，我统计了Zeus网银木马中你应该认为是合理的“源代码”部分（.cpp、.obj、.h等）的代码行数，结果是253 774行。将一个像是Zeus网银木马这样的恶意程序与麦基的恶意代码样本集平均大小相比，我们现在看到了一个新的比值，超过了2 000：1。

然后麦基将恶意代码源代码行数与用来检测与查杀恶意代码的安全产品源代码行数进行比较，他估计现在主流的安全防御产品源代码行数大概是1000万行左右。为了让算术变得简单一些，来想象一样存在有至少1250万行源代码的安全防护产品，这样就使得防御性软件代码与进攻性软件代码的比值进入了100 000：1的级别。换句话说，对于每1行源代码的进攻火力，防御者需要写出10万行作为防御堡垒的源代码。

---

1 *http://www.informit.com/articles/article.aspx?p=1686289*

2 *http://git.gnome.org/browse/gedit/tree/gedit?id=3.3.1*

麦基也将恶意代码源代码行数与恶意代码试图颠覆的操作系统源代码行数进行了比较。分析师估计Windows XP操作系统的源代码行数为4 500万行。而没人知道Windows 7有多少行源代码。麦基认为一个现代操作系统大概会有15 000万行源代码，想必他指的是Windows的最新版本。同样让我们保守估计到12 500万行，以简化我们的计算，这样我们就得出了一个100万比1的比值，来估计目标操作系统的大小规模与滥用操作系统的恶意武器大小规模之间的比例。

让我们停下来，概括一下我们已经获得的源代码行数计数分析结果。

- 120:1：Stuxnet病毒与平均恶意代码大小规模的比值。
- 500:1：简单的文本编辑器与平均恶意代码大小规模的比值。
- 2 000:1：恶意代码套件与平均恶意代码大小规模的比值。
- 100 000:1：防守工具软件与平均恶意代码大小规模的比值。
- 1 000 000:1：目标操作系统与平均恶意代码大小规模的比值。

从一位防御者的角度来看，防御性工具和目标操作系统和平均恶意代码大小规模的比值，是多么地让人心情沮丧。甚至将平均恶意代码大小规模换成恶意代码套件的大小规模，也不会很好地改善防御者所面临的境遇！它看起来像是我们的防御者（以及他们的供应商）花费了很多努力生产出成千上万的源代码，而最终只能看到它们被装备着少得多源代码的入侵者们无情戏弄。

那防御者应该怎么做呢？答案就是不要被那些火力强大的领跑者牵着鼻子走，而将一个“障碍”重新定义为“机会”！忘记安全防御工具和目标操作系统的大小规模——那可不是你可以左右的东西。回到你可以欢呼雀跃的事实上来，因为恶意代码样本的规模还是比较小的（相对来说）。

想象一下如果你试图去了解安全防御工具在源代码级别上是如何工作的，那里便有1 250万行源代码等着你来分析，这无疑将是一项非常艰巨的任务，尽管一些研究人员也将这种不可思议的项目分配给自己。作为一个令人难以置信的例子，请阅读由Tavis Ormandy 撰写的“Sophail：对Sophos反病毒软件的批判性分析”[3]，同样也在2011年的Black Hat拉斯维加斯黑客大会上进行了展示。这种庞大规模的分析是一种另类，而并不是常人可以选择的准则。

不要试图去担心数以百万行计的源代码（或者是成千上万行），关注那些大概一千行甚至更少的源代码——这也是目前世界上大多数恶意代码主要的组成部分。作为一名防御者，你对恶意代码进行分析的主要目标是：确定它做了什么、它在你的环境中是如何表现的，以及我们应该如何处理它。当处理这些合理规模的样本，并积累合适的技能之后，你便有机会来回答这些问题，从而降低企业的安全风险。

如果说恶意代码编写者随时会向你提供此类规模的样本，而你正在阅读的这本书会为你提供所需要的技能。《恶意代码分析实战》是一本我认为每一个恶意代码分析师都应该作为指导手册而珍

[3] *http://dl.packetstormsecurity.net/papers/virus/Sophail.pdf*

藏的书籍。如果你是一个初学者，你需要仔细研读技术介绍，以及动手实践每个实验作业，来进入到逆向工程的世界。而如果你是一位中等水平的分析师，本书会带着你提升到一个新的水平。如果你是一个资深的安全工程师，你也会发现那些额外的宝石，来推动你达到更高的级别——你就能够在被你所指导的一些新手问到问题时，告诉他们说：“去读读这本优秀的指南”。

本书实际上是一本二合一的书籍——首先，它是一本教材，指导读者如何分析现代的恶意代码。你可能只是以这个理由购买了本书，并已经从本书的技术指导中受益匪浅。然而，作者们又加倍努力，基本上是又写了一本书，其中包含了大量的实验练习、简短答案，以及详细分析过程，呈现在书籍中每章的后面与附录C中。作者们同时编写了所有用作例子的恶意代码样本，确保了一个既丰富又安全的学习环境。

因此，当你作为数字领域防御者，面对明显的不对称局面时，请不要绝望，相反的，你应该对恶意代码目前采用的技术形式表达出乐观态度。有了像《恶意代码分析实战》这样优秀的指导书籍之后，你就可以拥有所需的技能，来更好地检测与应对你的企业或者客户所面对的入侵事件。作者们都是这个领域内的技术专家，你会发现很多从工作一线提取的建议，而不是从一个孤立研究实验室中得到的一些理论。接下来你可以沉浸在本书的阅读中，获得关于恶意代码每一个细节的知识，并提升你的逆向工程技巧，通过将攻击者的黑暗艺术曝光在知识的阳光下，让他们尝到失败，付出代价。

Richard • Bejtlich（@ taosecurity）

Mandiant公司首席安全官，TaoSecurity创始人

马纳萨斯公园，弗吉尼亚州

2012年1月2日

致　谢

感谢林赛·莱克（Lindsey Lack）、尼克·哈波尔（Nick Harbour）、杰罗尔德·“杰”·史密斯（Jerrold “Jay” Smith）在他们各自专业特长领域为本书贡献的章节。

感谢我们的技术编辑——斯蒂芬·劳勒（Stephen Lawler），他单枪匹马地回顾了超过50个实验，以及我们撰写的所有章节。

谢谢Seth Summersett、William Ballenthin和Stephen Davis为本书贡献的代码。

特别要感谢在No Starch出版社工作的每个人为本书做出的努力。艾利森（Alison）、比尔（Bill）、特拉维斯（Travis）和泰勒（Tyler）：我们很高兴与你们以及No Starch出版社的其他人一起工作。

个人致谢

迈克尔：我将本书献给Rebecca——如果没有这位在我生命中这么支持我的爱人，我不可能取得现在的成就。

安迪鲁：我想感谢Molly、Claire、Eloise，你们和我一起组成了最好的家庭。

前　言

电话铃声急促响起，网络管理员告诉你说公司网站被黑了，网站上的客户敏感信息被盗了。于是你立马开始调查分析，首先检查了日志记录，来确定事件涉及的主机。你用杀毒软件对这些主机进行了扫描，检查是否感染了恶意代码。你的运气还算不错，杀毒软件检测到一个木马程序，名为TROJ.snapAK。你删除这个文件，并清理了现场，同时你还部署了一个入侵检测系统，来确认没有其他主机被感染。最后你修补了一个你认为是被攻击者利用来入侵主机的安全漏洞，来确保这种攻击事件不会再次发生。

不幸的是，几天之后网络管理员再次打电话过来，告诉你说敏感信息又被窃取了。这看起来似乎是相同的攻击，但你却不知道该做什么。很显然，你部署的入侵检测系统特征库失效了。因为更多的主机被感染了，而你的杀毒软件并没有提供足够的保护来隔离攻击威胁。现在，公司高层管理人员要求你解释发生了什么，而你可以告诉他们的只是一个名为TROJ.snapAK的恶意代码。你没有针对最重要问题的答案，这让他们认为你是一位不称职的安全工程师。

你该如何确定TROJ.snapAK恶意代码在做什么，从而可以让你消除这个威胁？你如何才能写出一个更有效的网络检测特征？你怎样才能找出其他感染了这个恶意代码的主机呢？你该如何确保你删除了整个恶意代码程序包，而不只是其中的一部分呢？你该如何回答管理层关于这个恶意代码干了些什么的问题呢？

如果你所有能做的，只是告诉你的老板，说你需要聘请昂贵的外部咨询顾问，因为你不能保护自己的网络，这真的不是确保工作饭碗的好办法。

幸运的是，你有着足够的智慧，马上啃起了这本《恶意代码分析实战》，从这本书中你将学到的技能，可以教你如何来回答这些困难的问题，并为你展示保护网络免受恶意代码侵害的方法。

什么是恶意代码分析

恶意代码，也称为恶意软件，在大多数计算机入侵事件中都扮演了重要角色。任何以某种方式来对用户、计算机或网络造成破坏的软件，都可以被认为是恶意代码，包括计算机病毒、木马、蠕虫、内核套件、勒索软件、间谍软件，等等。尽管各种不同的恶意代码类型会做一些完全不同的事情（你将会在本书中看到），作为恶意代码分析师，我们拥有一组核心的工具和技术，用来解剖分析各式各样的恶意代码。

恶意代码分析是一种解剖恶意代码的艺术，了解恶意代码是如何工作的、如何识别它，以及如何战胜或消除它。你并不是需要成为一名超级黑客，才能进行恶意代码分析。

网络上每天有着数以百万计，甚至更多的恶意代码，恶意代码分析成为了任何一位从事计算机安全事件响应安全工程师的必需技能。此外，由于恶意代码分析专业人才的短缺，熟练的恶意代码分析师正处于强烈的人才需求之中。

这么说吧，这不是一本关于如何找到恶意代码的书籍。我们的重点是在如何分析已经找到的恶意代码。我们专注于Windows操作系统上发现的恶意代码——因为到目前为止，Windows操作系统还是最为常用的操作系统。但你所学到的技能可以为你在任何操作系统上分析恶意代码提供支持。我们还将专注在可执行文件上，因为它们是最常见的，也是你所遇到的最难以分析的一些文件。与此同时，我们选择不讨论如恶意JavaScript脚本、Java程序等其他类型的恶意代码，相反的是，我们选择对方法进行深入讨论，用于分析更加高级的威胁，比如后门、隐蔽性恶意代码和内核套件。

先决条件

不管你是否有恶意代码分析的背景或经验，你都会从本书中受益。

第1～3章将讨论基础的恶意代码分析技术，即使你没有安全或编程经验，也可以用这些技术来进行恶意代码分析。第4～14章则覆盖中等级别的内容，可以让你武装上一些用来分析大多数恶意程序的主流工具与技能。这些章节都需要一些关于编程语言的基本知识。第15～19章，则提供最先进的技术材料，即使对资深的恶意代码分析师来说都是有用的，因为这部分内容涵盖了恶意代码分析的一些战术和技巧，在分析最为复杂的恶意代码样本时都用得上，比如那些应用了对抗反汇编、反调试技术或加壳技术的恶意代码。

本书将教你如何以及何时使用各种恶意代码分析技术。了解何时应该使用特定的技术与掌握技术本身一样重要，因为在某个特定状况下使用了错误的技术，可能会是在令人沮丧地浪费时间。我们不会涵盖每一个工具，因为工具会随时改变，而它的核心功能才是最重要的。此外，我们将在整本书中使用切合实际的恶意代码样本（你可以从*http://www.practicalmalwareanalysis.com/*或*http://www.nostarch.com/malware.htm*下载），来为你揭示在分析真实世界中恶意代码时会遇到的各种状况。

实践动手学习

我们有着逆向工程和恶意代码分析专业课程的丰富教学经验，这些经验已经告诉我们，学生只有通过使用所学习的技能进行动手实践练习时，才能真正掌握和学到这些技能。我们也发现了实验作业的质量与讲授的课程内容同等重要，如果没有一个实验作业部分，要学会如何分析恶意代码是几乎不可能的。

从始至终，本书中绝大多数章节最后都会给出一些实验作业，让你来练习这一章中所讲授的技

术。这些实验作业为你提供了真实恶意代码样本的挑战，旨在展示你将在真实世界中遭遇到恶意代码中最为普遍的类型和行为。这些实验作业旨在加强每章中所介绍的基本概念，而不会用一些无关信息来让你无所适从。每个实验都包括一个或多个恶意文件（可以从*http://www.practicalmalwareanalysis.com/*，或者*http://www.nostarch.com/malware.htm*下载），以及一些特意设计来引导你完成实验的问题，此外也给出了对这些问题的简短答案，以及对恶意代码样本的详细分析过程。

这些实验都模拟了真实的恶意代码分析场景。比如，它们都以通用化的文件名字进行命名，而不会提供任何能够洞察到恶意代码功能的信息。对于真正环境中的恶意代码，你也同样在开始分析时不会有任何信息，而你需要用你所学到的技能，来收集线索，并找出恶意代码在做些什么。

每个实验所需的时间将取决于你的经验。你可以尝试自己来完成实验，或者沿着详细分析过程，来了解如何在实践中使用各种技术。

大多数章节都包含了三个实验作业。第一个实验通常是最简单的，绝大多数读者都应该能够完成它。第二个实验是中等难度的，大多数读者会需要解答中的一些援助来完成。而第三个实验是最困难的，如果没有从参考答案取得提示，只有最勤奋和技术大拿的读者们才能够完成它们。

本书内容预览

《恶意代码分析实战》以使用简单的方法，从相对而言不那么复杂的恶意代码中获取信息开始，然后逐步提升难度，讲解可以用来对抗最为先进恶意程序的复杂技术。以下是本书每章的内容预览：

- 第0章，“恶意代码分析技术入门”，建立起恶意代码分析的整体过程和基础方法学。
- 第1章，“静态分析基础技术”，传授无须执行就能从可执行文件获取信息的方法。
- 第2章，“在虚拟机中分析恶意代码”，带你一起设置虚拟机，用作运行恶意代码的安全环境。
- 第3章，“动态分析基础技术”，介绍一些通过执行恶意程序进行分析、易于使用但非常高效的技术方法。
- 第4章，“x86反汇编速成班”，是对x86汇编语言的一个简要介绍，这章为使用IDA Pro进行恶意代码深入分析提供了基础。
- 第5章，“IDA Pro”，为你显示如何使用IDA Pro，一个最为重要的恶意代码分析工具。我们将在全书的其余章节使用IDA Pro工具。
- 第6章，“识别汇编中的C代码结构”，提供了一些C语言代码的汇编语句案例，并教你如何理解汇编代码的高层功能结构。
- 第7章，“分析恶意Windows程序”，覆盖范围广泛的Windows程序特定概念，而这些是理解恶意Windows程序所必需的。
- 第8章，“动态调试”，解释调试的基本知识，以及恶意代码分析师该如何使用调试器。
- 第9章，“OllyDbg”，为你展示如何使用OllyDbg，恶意代码分析师中最流行的一款调试器。

- 第10章，“使用WinDbg调试内核”，包括了如何使用WinDbg来分析内核模式恶意代码和内核套件。
- 第11章，“恶意代码行为”，介绍了常见的恶意代码功能，并告诉你在分析恶意代码时该如何识别恶意功能。
- 第12章，“隐蔽的恶意代码启动”，讨论如何分析一类将自己的执行隐藏至另一进程中的特殊恶意代码。
- 第13章，“数据加密”，演示了恶意代码如何加密数据，使其更难在网络流量或受害主机上被发现。
- 第14章，“恶意代码的网络特征”，教你如何通过恶意代码分析来创建网络检测特征，并演示这类特征要优于单独从捕获网络流量中提取的特征。
- 第15章，“对抗反汇编”，解释一些恶意代码编写者如何设计自己的恶意代码，使得它们难以被反汇编，并说明如何识别和击败这些技术。
- 第16章，“反调试技术”，描述恶意代码编写者可以让他们的代码难以被调试的伎俩，以及克服这些障碍的方法。
- 第17章，“反虚拟机技术”，演示恶意代码所使用的反虚拟机技术，这些技术会让分析师在虚拟机中难以分析这些恶意代码，并介绍绕过这些技术的方法。
- 第18章，“加壳与脱壳”，告诉读者恶意代码是如何使用加壳来隐藏自己真正目的的，然后提供一步一步的脱壳的技术方法。
- 第19章，“shellcode分析”，解释了shellcode是什么，并展示分析恶意shellcode的一些技巧和窍门。
- 第20章，“C++代码分析”，为你指明C++代码在编译之后存在什么样的差异，并教授你如何对由C++编写的恶意代码进行分析。
- 第21章，“64位恶意代码”，讨论恶意代码编写者为何使用64位恶意代码，以及你所需要了解的x86与x64之间的区别。
- 附录A，“常见Windows函数列表”，简要介绍了恶意代码中普遍使用的Windows函数。
- 附录B，“流行的恶意代码分析工具列表”，列出了对于恶意代码分析师们来说最有用的工具。
- 附录C，“实验作业参考解答”，对全书每个章节的实验给出了参考解答。
- 附录D，“致青春，基础软件开发的中国故事”，讲述中国程序员开发Windows内核调试器Syser的幕后故事。
- 附录E，“Syser操作入门”，提供Syser内核调试器的入门指南。

我们整本书的目标就是为你武装能够击败各种类型恶意代码的分析技术。正如你看到的，我们涵盖了大量的技术材料，以及能够加强这些技术材料的实验作业。当你读完本书时，应该学到了用来分析任何恶意代码的技能，包括快速分析常规恶意代码样本的基础技术，以及解剖那些甚至是“来自外星”的神秘恶意代码样本所需的高超技术。

那么，让我们开始吧！

目　录