

# 《恶意软件分析诀窍与工具箱》

## 图书基本信息

书名：《恶意软件分析诀窍与工具箱》

13位ISBN编号：9787302274407

10位ISBN编号：7302274401

出版时间：2012-1

出版社：清华大学出版社

作者：Michael Hale Ligh, Steven Adair, Blake Hartstein, Matthew Richard

页数：584

译者：胡乔林, 钟读航

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《恶意软件分析诀窍与工具箱》

## 内容概要

针对多种常见威胁的强大而循序渐进的解决方案

我们将《恶意软件分析诀窍与工具箱——对抗“流氓”软件的技术与利器》称为工具箱，是因为每个诀窍都给出了解决某个特定问题或研究某个给定威胁的原理和详细的步骤。在配书光盘中提供了补充资

源，您可以找到相关的支持文件和原始程序。您将学习如何使用这些工具分析恶意软件，有些工具是作者

自己开发的，另外数百个工具则是可以公开下载的。如果您的工作涉及紧急事件响应、计算机取证、系统

安全或者反病毒研究，那么本书将会为您提供极大的帮助。

- 学习如何在不暴露身份的前提下进行在线调查

- 使用蜜罐收集由僵尸和蠕虫分布的恶意软件

- 分析JavaScript、PDF文件以及Office文档中的可疑内容

- 使用虚拟或基础硬件建立一个低预算的恶意软件实验室

- 通用编码和加密算法的逆向工程

- 建立恶意软件分析的高级内存取证平台

- 研究主流的威胁，如Zeus、Silent Banker、CoreFlood、Conficker、Virus、Clampi、Bankpatch、BlackEnergy等

# 《恶意软件分析诀窍与工具箱》

## 作者简介

Michael Hale Ligh是Verisign iDefense公司的恶意代码分析专家，专门从事开发各种用于检测、解密以及调查恶意软件的工具。在过去数年里，他在里约热内卢、上海、吉隆坡、伦敦、华盛顿特区和纽约等地讲授恶意软件分析课程，已经培训了数百名学生。在进入Verisign iDefense公司之前，Michael在全国最大的医疗保健服务提供商之一中担任漏洞研究员，并提供黑客伦理服务。正是由于担任过该职务，他对逆向工程以及操作系统内部的背景有着深刻理解。在此之前，Michael为新英格兰地区的金融机构提供网络防御以及取证调查方面的服务。他目前是MNIN安全有限公司的特别项目主管。

Steven Adair是Shadowserver Foundation的研究员，同时也是eTouch联邦系统的首席架构师。在Shadowserver组织中，Steven主要分析恶意软件和跟踪僵尸网络，并重点调查与网络间谍组织相关联的各种网络攻击。Steven经常出席该领域相关专题的国际会议，并且合著了论文“Shadows in the Cloud: Investigating Cyber Espionage 2.0”。在日常工作中，他在一个联邦机构中领导网络威胁行动小组以主动检测、降低以及预防网络入侵活动，他有效地集成了最佳安全实践和创新技术，成功地在全网中实现了企业级反恶意软件解决方案。Steven每天的工作都涉及恶意软件研究，无论是为公司客户提供支持或者在Shadowserver组织中贡献自己的业余时间。

Blake Hartstein是Verisign iDefense公司的快速响应工程师，主要负责分析以及报告恶意软件的可疑活动。他是Jsunpack工具的编写者，致力于自动分析以及检测基于Web的漏洞攻击，并分别在Shmoocon 2009和Shmoocon 2010会议中做了关于Jsunpack的报告。Blake同时还为Emerging Threats项目编写和贡献Snort规则。

Matthew Richard是雷神(Raytheon)公司的恶意代码操作部领导，主要负责分析以及报告恶意代码。Matthew之前是iDefense公司快速响应部门主管。在此7年之前，Matthew创建并运营了一家向130多家银行以及信用机构提供安全服务的公司。此外，他还对国内以及全球多家公司提供独立的网络取证咨询服务。Matthew现持有CISSP、GCIA、GCFA和GREM认证。

## 书籍目录

### 目 录

#### 第1章 行为隐匿

1

##### 1.1 洋葱路由器(Tor)

1

##### 1.2 使用Tor研究恶意软件

4

##### 1.3 Tor缺陷

7

###### 1.3.1 速度

7

###### 1.3.2 不可信赖的Tor操作员

7

###### 1.3.3 Tor阻止列表

8

##### 1.4 代理服务器和协议

8

###### 1.4.1 超文本传输协议(HTTP)

8

###### 1.4.2 SOCKS4

9

###### 1.4.3 SOCKS5

9

##### 1.5 基于Web的匿名代理

14

##### 1.6 保持匿名的替代方法

15

###### 1.6.1 蜂窝Internet连接

15

###### 1.6.2 虚拟专用网

16

##### 1.7 唯一且匿名

18

#### 第2章 蜜罐

19

##### 2.1 nepenthes蜜罐

20

###### 2.1.1 利用nepenthes收集恶意软件

样本

20

###### 2.1.2 使用IRC日志进行实时攻击

监视

23

###### 2.1.3 使用基于Python的 HTTP接收nepenthes提交的文件

25

##### 2.2 使用dionaea蜜罐

27	
2.2.1	使用dionaea收集恶意软件样本
27	
2.2.2	使用基于Python的HTTP接收dionaea提交的文件
30	
2.2.3	实时事件通告以及使用XMPP共享二进制文件
31	
2.2.4	分析重放dionaea记录的攻击
32	
2.2.5	使用p0f工具被动识别远程主机操作系统
33	
2.2.6	使用SQLite和gnuplot绘制dionaea记录的攻击模式图
35	
第3章	恶意软件分类
39	
3.1	使用ClamAV分类
39	
3.1.1	检查现有ClamAV特征码
40	
3.1.2	创建自定义ClamAV特征码数据库
42	
3.2	使用YARA分类
46	
3.2.1	将ClamAV特征码转换到YARA格式特征码
46	
3.2.2	使用YARA和PEiD识别加壳文件
47	
3.2.3	使用YARA检测恶意软件的能力
49	
3.3	工具集成
53	
3.3.1	使用Python识别文件类型及哈希算法
54	
3.3.2	编写Python多杀毒扫描软件
56	

3.3.3 Python中检测恶意PE文件	60
3.3.4 使用ssdeep查找相似恶意软件	64
3.3.5 使用ssdeep检测自修改代码	66
3.3.6 使用IDA和BinDiff检测自修改代码	68
第4章 沙箱和多杀毒扫描软件	73
4.1 公用杀毒扫描软件	73
4.1.1 使用Virus Total扫描文件	74
4.1.2 使用Jotti扫描文件	76
4.1.3 使用NoVirusThanks扫描文件	77
4.1.4 启用数据库的Python多杀毒上传程序	78
4.2 多杀毒扫描软件比较	82
4.3 公用沙箱分析	82
4.3.1 使用ThreatExpert分析恶意软件	82
4.3.2 使用CWSandbox分析恶意软件	84
4.3.3 使用Anubis分析恶意软件	85
4.3.4 编写Joebox AutoIt脚本	86
4.3.5 使用Joebox应对路径依赖型恶意软件	88
4.3.6 使用Joebox应对进程依赖型动态链接库	89
4.3.7 使用Joebox设置主动型HTTP代理	

91	
4.3.8	使用沙箱结果扫描项目
92	
	第5章 域名与IP地址
99	
5.1	研究可疑域名
99	
5.1.1	利用WHOIS研究域
100	
5.1.2	解析DNS主机名
104	
5.2	研究IP地址
107	
5.3	使用被动DNS和其他工具进行
	研究
109	
5.3.1	使用BFK查询被动
	DNS
110	
5.3.2	使用robtex检查DNS
	记录
111	
5.3.3	使用DomainTools执行反向IP
	搜索
112	
5.3.4	使用DIG启动区域
	传送
113	
5.3.5	使用dnsmap暴力攻击
	子域
114	
5.3.6	通过Shadowserver将IP地址
	映射到ASN
115	
5.3.7	使用RBL检查IP信誉
117	
5.4	fast flux域名
118	
5.4.1	使用被动DNS和TTL检测
	fast flux网络
119	
5.4.2	跟踪fast flux域名
121	
5.5	IP地址地理映射
123	
	第6章 文档、shellcode和URL
127	
6.1	分析JavaScript
127	

6.1.1 使用SpiderMonkey分析JavaScript	127
6.1.2 使用Jsunpack自动解码JavaScript	130
6.1.3 优化Jsunpack-n的解码速度和完整性	133
6.1.4 通过模拟浏览器DOM元素触发漏洞利用	134
6.2 分析PDF文档	138
6.2.1 使用pdf.py从PDF文件中提取JavaScript	138
6.2.2 伪造PDF软件版本触发漏洞利用	142
6.2.3 利用Didier Stevens的PDF工具集	145
6.2.4 确定利用PDF文件中的哪些漏洞	148
6.2.5 使用DiStorm反汇编shellcode	154
6.2.6 使用libemu模拟shellcode	159
6.3 分析恶意Office文档	161
6.3.1 使用OfficeMalScanner分析Microsoft Office文件	161
6.3.2 使用DisView和MalHost-Setup调试Office shellcode	167
6.4 分析网络流量	170
6.4.1 使用Jsunpack从报文捕获文件中提取HTTP文件	171
6.4.2 使用Jsunpack绘制URL关系图	173
第7章 恶意软件实验室	

177	
7.1 网络互联	
179	
7.1.1 实验室中TCP/IP路由 连接	
180	
7.1.2 捕获、分析网络流量	
182	
7.1.3 使用INetSim模拟 Internet	
185	
7.1.4 使用Burp套件操作 HTTP/HTTPS	
188	
7.2 物理目标机	
191	
7.2.1 使用Joe Stewart开发的 Truman	
191	
7.2.2 使用Deep Freeze保护物理 系统	
192	
7.2.3 使用FOG克隆和映像 磁盘	
194	
7.2.4 使用MySQL数据库自动调度 FOG任务	
197	
第8章 自动化操作	
201	
8.1 恶意软件分析周期	
201	
8.2 使用Python实现自动化 操作	
203	
8.2.1 使用VirtualBox执行自动化 恶意软件分析	
203	
8.2.2 分析VirtualBox磁盘以及 内存映像	
208	
8.2.3 使用VMware执行自动化 恶意软件分析	
210	
8.3 添加分析模块	
213	
8.3.1 在Python中使用tshark捕获 报文	
214	

8.3.2 在Python中使用INetSim收集网络日志	216
8.3.3 使用Volatility分析内存转储	217
8.3.4 组合所有的沙箱块	219
8.4 杂项系统	229
8.4.1 使用ZeroWine和QEMU执行自动化分析	229
8.4.2 使用Sandboxie和Buster执行自动化分析	233
第9章 动态分析	237
9.1 变化检测	237
9.1.1 使用Process Monitor记录API调用	239
9.1.2 使用Regshot进行变化检测	240
9.1.3 接收文件系统变化通知	242
9.1.4 接收注册表变化通知	245
9.1.5 句柄表的差异比较	246
9.1.6 使用HandleDiff研究代码注入	250
9.1.7 观察Bankpatch.C禁用Windows文件保护的活动	252
9.2 API监视/钩子	253
9.2.1 使用Microsoft Detours构建API监视器	254
9.2.2 使用API监视器追踪子进程	260
9.2.3 捕获进程、线程和映像加载事件	

263	
9.3 数据保护	
267	
9.3.1 阻止进程终止	
268	
9.3.2 阻止恶意软件删除文件	
270	
9.3.3 阻止加载驱动程序	
272	
9.3.4 使用数据保护模块	
273	
9.3.5 使用ReactOS创建定制命令 shell	
276	
第10章 恶意软件取证	
283	
10.1 The Sleuth Kit(TSK)	
283	
10.1.1 使用TSK发现备用 数据流	
283	
10.1.2 使用TSK检测隐藏文件 和目录	
286	
10.1.3 使用Microsoft脱机API 查找隐藏注册表数据	
293	
10.2 取证/事件响应混合	
298	
10.2.1 绕过Poison Ivy锁定 的文件	
298	
10.2.2 绕过Conficker文件系统的 ACL限制	
302	
10.2.3 使用GMER扫描 rootkit	
306	
10.2.4 通过检查IE的DOM 检测HTML注入	
309	
10.3 注册表分析	
318	
10.3.1 使用RegRipper插件对注册 表取证	
319	
10.3.2 检测恶意安装的PKI 证书	
325	

## 10.3.3 检查泄露数据到注册表的 的恶意软件

328

## 第11章 调试恶意软件

335

### 11.1 使用调试器

335

#### 11.1.1 打开和附加到进程

336

#### 11.1.2 为shellcode分析配置JIT 调试器

337

#### 11.1.3 熟悉调试器的图形用户 界面

339

#### 11.1.4 检查进程内存和资源

344

#### 11.1.5 控制程序执行

347

#### 11.1.6 设置和捕获断点

348

#### 11.1.7 使用有条件的日志记录 断点

351

### 11.2 Immunity Debugger的Python API接口

352

#### 11.2.1 使用Python脚本和 PyCommand调试

353

#### 11.2.2 在二进制文件中检测 shellcode

356

#### 11.2.3 调查Silentbanker木马的 API钩子

360

### 11.3 WinAppDbg Python调试器

363

#### 11.3.1 使用WinAppDbg工具操作 进程内存

364

#### 11.3.2 使用WinAppDbg工具设计 一个Python API监视器

366

## 第12章 反混淆

373

### 12.1 解码常见算法

373

#### 12.1.1 Python中的逆向XOR

算法	
373	
12.1.2 使用yaratize检测XOR	
编码的数据	
378	
12.1.3 使用特殊字母解码	
base64	
379	
12.2 解密	
382	
12.2.1 从捕获的数据包中隔离	
加密数据	
382	
12.2.2 使用SnD反向工具、	
FindCrypt和Kanal	
搜索加密机制	
384	
12.2.3 使用zynamics BinDiff移植	
Open SSL的符号	
386	
12.2.4 在Python中使用PyCrypto	
解密数据	
387	
12.3 恶意软件脱壳	
389	
12.3.1 查找加壳恶意软件	
的OEP	
390	
12.3.2 使用LordPE转储进程	
内存	
392	
12.3.3 使用ImpREC重建	
导入表	
394	
12.4 与脱壳有关的资源	
400	
12.5 调试器脚本	
401	
12.5.1 破解域名生成算法	
402	
12.5.2 使用x86emu和Python解码	
字符串	
406	
第13章 处理DLL	
411	
13.1 枚举DLL的导出函数	
411	
13.1.1 CFF Explorer	
412	

13.1.2 Pefile	412
13.1.3 IDA Pro	413
13.1.4 常见和不常见的导出名	414
13.2 使用rundll32.exe执行DLL	415
13.3 绕过宿主进程的限制	416
13.4 使用rundll32ex远程调用DLL导出函数	418
13.4.1 创建新工具的原因	418
13.4.2 使用rundll32ex	420
13.5 使用LOADDLL.EXE调试DLL	421
13.5.1 将DLL加载到调试器中	421
13.5.2 找到DLL的入口点	422
13.6 捕获DLL入口点处的断点	422
13.7 执行作为Windows服务的DLL	423
13.7.1 服务DLL的入口点	424
13.7.2 服务初始化	424
13.7.3 安装服务DLL	425
13.7.4 传递参数给服务	425
13.8 将DLL转换成独立的可执行文件	428
第14章 内核调试	431
14.1 远程内核调试	431
14.2 本地内核调试	431
14.3 软件需求	

432	
14.3.1	使用LiveKd进行本地调试
432	
14.3.2	启用内核调试启动开关
433	
14.3.3	调试VMware工作站客户机 (在Windows系统中)
436	
14.3.4	调试Parallels客户机 (在MAC OS X上)
438	
14.3.5	WinDbg命令和控制简介
439	
14.3.6	探索进程和进程上下文
445	
14.3.7	探索内核内存
451	
14.3.8	在驱动程序加载时捕捉断点
456	
14.3.9	脱壳驱动程序
463	
14.3.10	转储和重建驱动程序
469	
14.3.11	使用WinDbg脚本检测rootkit
474	
14.3.12	使用IDA Pro进行内核调试
479	
第15章	使用Volatility进行内存取证
483	
15.1	内存获取
483	
15.1.1	使用MoonSols Windows内存工具箱转储内存
483	
15.1.2	使用F-Response获取远程、只读内存
486	
15.1.3	访问虚拟机的内存文件
487	
15.2	准备安装Volatility

488	
15.2.1	Volatility概览
488	
15.2.2	在内存转储中研究 进程
491	
15.2.3	使用psscan检测DKOM 攻击
496	
15.2.4	研究csrss.exe的备用进程 列表
499	
15.2.5	识别进程上下文的技巧
501	
第16章	内存取证：代码注入与 提取
507	
16.1	深入研究DLL
507	
16.1.1	搜寻已加载的可疑 DLL
508	
16.1.2	使用ldr_modules检测 未链接的DLL
510	
16.2	代码注入和VAD
514	
16.2.1	研究VAD
514	
16.2.2	转换页面保护
517	
16.2.3	在进程内存中搜索 证据
520	
16.2.4	使用malfind和YARA 识别注入代码
522	
16.3	重建二进制文件
527	
16.3.1	从内存中重建可执行 文件的映像
529	
16.3.2	使用impscan扫描导入 函数
530	
16.3.3	转储可疑的内核模块
533	
第17章	内存取证：rootkit

537	
17.1	检测IAT钩子
537	
17.2	检测EAT钩子
539	
17.3	检测内联API钩子
540	
17.4	检测IDT钩子
543	
17.5	检测驱动程序的IRP钩子
544	
17.6	检测SSDT钩子
547	
17.6.1	SSDT的角色
548	
17.6.2	钩子和钩子检测
548	
17.7	使用ssdt_ex自动研究
551	
17.8	根据附加的内核线程搜索 rootkit
552	
17.8.1	使用线程在内核中 隐藏
553	
17.8.2	在内存转储中检测分 离线程
554	
17.9	识别系统范围的通知 例程
555	
17.9.1	找出检查的位置
555	
17.9.2	使用notifyroutines 插件
556	
17.10	使用svscan定位恶意的服务 进程
557	
17.10.1	恶意软件如何滥用 服务
558	
17.10.2	SCM的服务记录 结构
558	
17.10.3	枚举进程内存中 的服务
560	
17.10.4	Blazgel木马的例子

560	
17.10.5	使用Volatility的svcsan 插件
561	
17.11	使用mutantscan扫描互斥体 对象
564	
第18章	内存取证：网络和注册表
567	
18.1	探索套接字和连接对象
567	
18.1.1	套接字和连接证据
567	
18.1.2	套接字和连接对象
569	
18.2	分析Zeus留下的网络 证据
570	
18.3	检测企图隐藏TCP/IP 的活动
572	
18.3.1	扫描套接字和连接 对象
572	
18.3.2	其他项目
574	
18.4	检测原始套接字和混杂模式 的网络接口
574	
18.4.1	混杂模式的套接字
574	
18.4.2	检测混杂模式
575	
18.5	注册表分析
575	
18.5.1	使用内存注册表工具分析 注册表证据
576	
18.5.2	通过最后写入时间戳排序 注册表项
580	
18.5.3	使用Volatility和Reg- Ripper
582	

## 章节摘录

版权页：插图：日常生活中，我们喜欢拥有某种程度的隐私。窗户上有窗帘，办公室有门，甚至计算机都有特制的屏保以防止窥视。对隐私的需求也延伸到Internet的使用上。我们不希望其他人知道我们在Google中输入的内容、即时通信对话的内容，以及访问的网站。但是，如果有人监视，那么他们往往可以看到您的私密信息。访问Internet时有许多选择匿名方式的理由。而匿名并不意味着是您在做违法或错误的事情。调查恶意软件和追踪别有用心者时，进行匿名的理由非常直接。您不希望信息显示在日志或其他记录中，因为这可能暴露自己或自己所在公司的信息。例如，假设您在金融公司工作，最近检测到银行特洛伊木马感染了系统中的部分计算机。您收集恶意域名、IP地址和恶意软件相关的其他数据。在调查中，随后采取的步骤是找到罪犯拥有的网站。结果，如果未提前采取措施进行匿名访问，那么您的IP地址会被记录到各种日志中，还会被罪犯看到。

# 《恶意软件分析诀窍与工具箱》

## 媒体关注与评论

“ 本书是我今年所阅读过的最有用的安全技术书籍，对于所有希望保护其系统免受恶意软件威胁的人员来说，这是一本必备书籍。 ” ——Lenny Zeltser, Savvis公司的安全业务主管和SANS机构的高级教师 “ 每个恶意软件分析爱好者的终极行动指南。 ” ——Ryan, Olson VeriSign iDefense公司快速响应部门的主管 “ 每一页都充满了实用的恶意软件知识、创新的理念、有用的工具，非常值得购买！ ” ——Aaron Walters Terremark公司的Volatility和VP安全研究项目的领导者

# 《恶意软件分析诀窍与工具箱》

## 编辑推荐

《恶意软件分析诀窍与工具箱:对抗"流氓"软件的技术与利器》编辑推荐：“一本极好的恶意软件书籍” August14, 2011By Ashraf Aziz “ Ash Aziz ” “我乐意向在计算机取证领域工作的任何人甚至是桌面支持人员推荐《恶意软件分析诀窍与工具箱:对抗"流氓"软件的技术与利器》，不要忘记使用配书光盘中的命令和示例，它们也很有帮助。” “恶意软件分析人员（包括反应人员和CERT）必备书籍” January1, 2011By Aaed Salah Nemer “我已经阅读过很多有关恶意软件从概念到实验分析的安全书籍，但从来没有一本书籍像《恶意软件分析诀窍与工具箱:对抗"流氓"软件的技术与利器》这样给出了大量的技术细节。我进行的大量的安全活动，需要我具有最新的恶意软件的坚实背景，并提升我的事件响应技能和分析技术。” “一本优秀的书籍” December14, 2010By ShaWn “这是我读过的有关恶意软件分析的最好、最容易的书籍。《恶意软件分析诀窍与工具箱:对抗"流氓"软件的技术与利器》包含了有用的代码、巧妙的方法以及其他实用的信息。这并不是是一本随便拼凑编写几个工具的典型的大杂烩书籍。显而易见，作者在内容和组织上花费了大量心思。如果您也认真对待恶意软件分析，那么《恶意软件分析诀窍与工具箱:对抗"流氓"软件的技术与利器》值得您拥有。”

# 《恶意软件分析诀窍与工具箱》

## 精彩短评

- 1、内容全面丰富，极具实践性！
- 2、新的思路，想法，很详细！
- 3、恶意软件分析入门级的书
- 4、好书，做病毒分析，买来当参考书
- 5、对于恶意软件分析技术，有较强的实用性，在中国很少有看到这种实践针对性强的书了
- 6、有朋友说这本书写的比较浅。但对我来说，已经足够深了！深浅是相对的，但作为一个不想成为病毒专家的逆向票友来说。只有浅薄过，才能深
- 7、非常不错恶意软件分析书籍
- 8、针对多种常见威胁的强大而循序渐进的解决方案  
我们将《恶意软件分析诀窍与工具箱——对抗“流氓”软件的技术与利器》称为工具箱，是因为每个  
诀窍都给出了解决某个特定问题或研究某个给定威胁的原理和详细的步骤。在配书光盘中提供了补充资  
源，您可以找到相关的支持文件和原始程序。您将学习如何使用这些工具分析恶意软件，有些工具是作者  
自己开发的，另外数百个工具则是可以公开下载的。如果您的工作涉及紧急事件响应、计算机取证、系统  
安全或者反病毒研究，那么本书将会为您提供极大的帮助。
- 9、很多干货嘛
- 10、比较不错，实用性不强，但是很有意思。

# 《恶意软件分析诀窍与工具箱》

## 精彩书评

1、我看过了这本书，内容相当的全面，这本书确实有很多进行恶意软件分析的，平时想问而又不知道到哪里问的知识！尤其是后面的几个章节，详细的步骤和代码，绝对可以帮我们入门。翻译的质量也还是不错的。推荐网络安全的同学，尤其是分析恶意代码的，值得读一下

# 《恶意软件分析诀窍与工具箱》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)