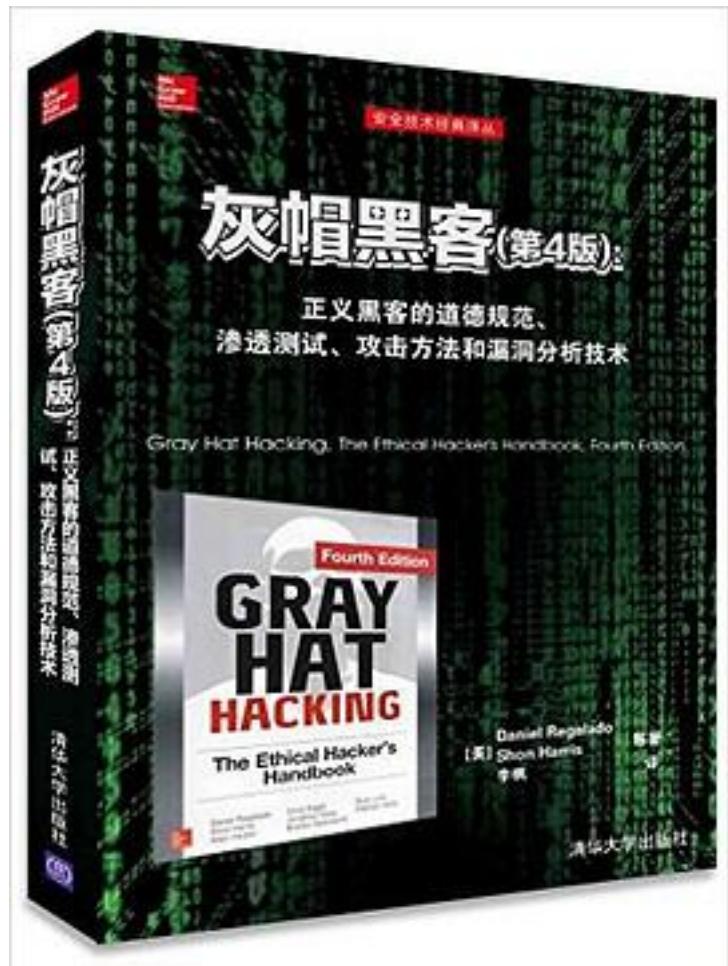


灰帽黑客(第4版)



[灰帽黑客\(第4版\) 下载链接1](#)

著者:[美] Daniel Regalado 里加拉多

出版者:计算机·安全·清华大学出版社

出版时间:2016-3

装帧:平装

isbn:9787302428671

《灰帽黑客(第4版):正义黑客的道德规范、渗透测试、攻击方法和漏洞分析技术》在上一版本的基础上做了全面细致的更新，新增了12章内容，分析敌方当前的武器、技能和

战术，提供切实有效的补救措施、案例研究和可部署的测试实验，并揭示黑客们如何获取访问权限、攻击网络设备、编写和注入恶意代码及侵占Web应用程序和浏览器。这个与时俱进的知识宝库也透彻讲解Android漏洞攻击、逆向工程技术和网络法律等主题。

作者介绍：

Daniel

Regalado，又名Danux，是一名墨西哥裔的高级恶意软件和漏洞研究员，目前在FireEye工作，之前是赛门铁克(Symantec)的逆向工程师。他在安全领域拥有12年以上的丰富经验，并获得包括CISSP、ISO

27001首席作者、OSCP、OSCE和CREA在内的多个认证证书，这使得他具备剖析高级持续性威胁(Advanced Persistent Threat，APT)所需的高超渗透测试和逆向工程的技巧。他乐于分享自己的研究成果——新的演讲是在2014年度的BsidesLV大会上。他也喜欢在个人博客(danuxx.blogspot.com)上记录有趣的研究发现。

Shon Harris，Logical

Security公司创始人兼首席执行官，同时也是一名作家、教育工作者和安全顾问。她曾是美国空军信息战(U.S. Air Force Information Warfare)部队的工程师，并出版过数本信息安全领域不同学科的书籍。并被Information

Security Magazine评为信息安全领域25位杰出的女性精英之一。

目录: 第Ⅰ部分 速成课：备战

第1章 道德黑客和法律制度 3

1.1 理解敌方策略的意义 3

1.2 正义黑客过程 4

1.2.1 渗透测试过程 5

1.2.2 不道德黑客的做法 7

1.3 网络法的兴起 8

1.3.1 了解各种网络法 8

1.3.2 关于“黑客”工具的争论 13

1.4 漏洞披露 13

1.4.1 各方看待问题的不同角度 14

1.4.2 个中缘由 14

1.4.3 CERT目前采取的工作流程 15

1.4.4 Internet安全组织 16

1.4.5 争议仍将存在 17

1.4.6 再没有免费的bug了 18

1.4.7 bug赏金计划 19

1.5 本章小结 19

1.6 参考文献 20

1.7 扩展阅读 21

第2章 编程技能 23

2.1 C编程语言 23

2.1.1 C语言基本结构 23

2.1.2 程序范例 27

2.1.3 使用gcc进行编译 28

2.2 计算机内存 29

2.2.1 随机存取存储器(RAM) 29

2.2.2 字节序 29

2.2.3 内存分段 30

2.2.4 内存中的程序 30

2.2.5 缓冲区	31
2.2.6 内存中的字符串	31
2.2.7 指针	31
2.2.8 内存知识小结	32
2.3 Intel处理器	32
2.4 汇编语言基础	33
2.4.1 机器指令、汇编语言与C语言	33
2.4.2 AT&T与NASM	33
2.4.3 寻址模式	36
2.4.4 汇编文件结构	37
2.4.5 汇编过程	37
2.5 使用gdb进行调试	37
2.5.1 gdb基础	38
2.5.2 使用gdb进行反汇编	39
2.6 Python编程技能	40
2.6.1 获取Python	40
2.6.2 Python的Hello World程序	40
2.6.3 Python对象	41
2.6.4 字符串	41
2.6.5 数字	42
2.6.6 列表	43
2.6.7 字典	44
2.6.8 Python文件操作	45
2.6.9 Python套接字编程	46
2.7 本章小结	47
2.8 参考文献	47
2.9 扩展阅读	47
第3章 静态分析	49
3.1 道德的逆向工程	49
3.2 使用逆向工程的原因	50
3.3 源代码分析	51
3.3.1 源代码审计工具	51
3.3.2 源代码审计工具的实用性	53
3.3.3 手工源代码审计	54
3.3.4 自动化源代码分析	59
3.4 二进制分析	60
3.4.1 二进制代码的手工审计	60
3.4.2 自动化的二进制分析工具	72
3.5 本章小结	74
3.6 扩展阅读	74
第4章 使用IDA Pro进行高级分析	75
4.1 静态分析难点	75
4.1.1 剥离的二进制文件	75
4.1.2 静态链接程序和FLAIR	77
4.1.3 数据结构分析	83
4.1.4 已编译的C++代码的怪异之处	87
4.2 扩展IDA Pro	89
4.2.1 IDAPython脚本	90
4.2.2 执行Python代码	98
4.3 本章小结	98
4.4 扩展阅读	98
第5章 模糊测试的世界	101

5.1 模糊测试简介	101
5.2 选择目标	102
5.2.1 输入类型	102
5.2.2 易于自动化	102
5.2.3 复杂性	103
5.3 模糊器的类型	104
5.3.1 变异模糊器	104
5.3.2 生成模糊器	105
5.4 开始	105
5.4.1 寻找模糊测试模板	106
5.4.2 实验 5-1: 从互联网档案馆获取样本	107
5.4.3 利用代码覆盖率选取最优模板集	108
5.4.4 实验 5-2: 为模糊测试选取最优样本	109
5.5 Peach模糊测试框架	110
5.5.1 Peach模糊测试策略	115
5.5.2 速度的重要性	116
5.5.3 崩溃分析	116
5.5.4 实验5-3: Peach变异模糊测试	120
5.5.5 其他变异模糊器	121
5.6 生成模糊器	121
5.7 本章小结	122
5.8 扩展阅读	122
第6章 shellcode策略	125
第7章 编写Linux shellcode	139
shellcode进行编码	166
7.6 本章小结	167
7.7 扩展阅读	167
第 II 部分 漏洞攻击	
第8章 基于欺骗的攻击	171
8.1 什么是欺骗	171
8.2 ARP欺骗	172
8.2.1 实验8-1: 使用Ettercap的ARP欺骗	173
8.2.2 查看网络流量	174
8.2.3 修改网络流量	175
8.3 DNS欺骗	181
8.3.1 实验8-2: 使用Ettercap进行DNS欺骗	182
8.3.2 执行攻击	183
8.4 NetBIOS名称欺骗和LLMNR欺骗	184
8.4.1 实验8-3: 使用Responder攻击NetBIOS和LLMNR	185
8.4.2 破解NTLMv1和NTLMv2哈希	188
8.5 本章小结	188
8.6 扩展阅读	189
第9章 攻击Cisco路由器	191
9.1 攻击团体字符串和密码	191
9.1.1 实验9-1: 使用Ncrack和	

Metasploit来猜测凭据	191
9.1.2 实验9-2：使用onesixtyone和Metasploit猜测团体字符串	193
9.2 SNMP和TFTP	195
9.2.1 实验9-3：使用Metasploit 下载配置文件	195
9.2.2 实验9-4：使用SNMP和 TFTP修改配置	197
9.3 攻击Cisco密码	199
9.3.1 攻击CiscoType 7密码	199
9.3.2 实验9-5：使用Cain破解 Type 7密码	200
9.3.3 实验9-6：使用Metasploit 解密Type 7密码	200
9.3.4 攻击CiscoType 5密码	201
9.3.5 实验9-7：使用John the Ripper 攻击CiscoType 5密码	201
9.4 使用隧道中转流量	202
9.4.1 实验9-8：建立GRE隧道	203
9.4.2 实验9-9：在GRE隧道上 路由流量	205
9.5 漏洞攻击和其他攻击	209
9.5.1 Cisco漏洞攻击	209
9.5.2 保持对Cisco设备的访问	210
9.6 本章小结	210
9.7 扩展阅读	211
第10章 基本的Linux漏洞攻击	213
10.1 栈操作	213
10.2 缓冲区溢出	214
10.2.1 实验10-1：meet.c 溢出	216
10.2.2 缓冲区溢出的后果	219
10.3 本地缓冲区溢出漏洞攻击	220
10.3.1 实验10-2：漏洞攻击的 组件	220
10.3.2 实验10-3：在命令行上 进行栈溢出漏洞攻击	222
10.3.3 实验10-4：使用通用漏 洞攻击代码进行栈溢出 漏洞攻击	224
10.3.4 实验10-5：对小缓冲区 进行漏洞攻击	225
10.4 漏洞攻击的开发过程	228
10.4.1 实验10-6：构建定制 漏洞攻击	228
10.4.2 确定偏移	229
10.4.3 确定攻击向量	231
10.4.4 生成shellcode	232
10.4.5 验证漏洞攻击	233
10.5 本章小结	234
10.6 扩展阅读	234
第11章 高级Linux漏洞攻击	235
11.1 格式化字符串漏洞攻击	235
11.1.1 问题描述	235

11.1.2 实验11-1：从任意内存读取 238
11.1.3 实验11-2：写入任意内存 241
11.1.4 实验11-3：改变程序执行 242
11.2 内存保护机制 245
11.2.1 编译器的改进 245
11.2.2 实验11-4：绕过堆栈保护 247
11.2.3 内核补丁和脚本 249
11.2.4 实验11-5：“Return to libc”漏洞攻击 250
11.2.5 实验 11-6：使用ret2libc保持权限 254
11.2.6 结论 258
11.3 本章小结 259
11.4 参考文献 259
11.5 扩展阅读 259
第12章 Windows漏洞攻击 261
12.1 Windows程序编译与调试 261
12.1.1 实验12-1：在Windows上编译程序 261
12.1.2 在Windows上使用Immunity Debugger进行调试 263
12.1.3 实验12-2：程序崩溃 265
12.2 编写Windows漏洞攻击程序 268
12.2.1 漏洞攻击程序开发过程回顾 268
12.2.2 实验12-3：攻击ProSSH服务器 268
12.3 理解结构化异常处理(SEH) 277
12.4 本章小结 279
12.5 参考文献 279
12.6 扩展阅读 279
第13章 绕过Windows内存保护 281
13.1 理解Windows内存保护(XP SP3、Vista、Windows 7/8、Server 2008和Server 2012) 281
13.1.1 基于栈的缓冲区溢出检测(/GS) 281
13.1.2 SafeSEH 282
13.1.3 SEHOP 283
13.1.4 堆保护 283
13.1.5 DEP 283
13.1.6 ASLR 284
13.1.7 EMET 285
13.2 绕过Windows内存保护 285
13.2.1 绕过/GS 285
13.2.2 绕过SafeSEH 286
13.2.3 绕过ASLR 287

- 13.2.4 绕过DEP 287
 - 13.2.5 绕过EMET 293
 - 13.2.6 绕过SEHOP 294
 - 13.3 本章小结 300
 - 13.4 参考文献 300
 - 13.5 扩展阅读 301
- 第14章 攻击Windows访问
- 控制模型 303
 - 14.1 为何黑客要攻击访问
 - 控制机制 303
 - 14.1.1 多数人并不理解访问
 - 控制机制 303
 - 14.1.2 访问控制漏洞易于攻击 304
 - 14.1.3 访问控制漏洞的数量巨大 304
 - 14.2 Windows访问控制的工作机制 304
 - 14.2.1 安全标识符 304
 - 14.2.2 访问令牌 305
 - 14.2.3 安全描述符 308
 - 14.2.4 访问检查 311
 - 14.3 访问控制配置的分析工具 314
 - 14.3.1 转储进程令牌 314
 - 14.3.2 转储SD 317
 - 14.4 特殊SID、特殊访问权限和“禁止访问” 318
 - 14.4.1 特殊的SID 318
 - 14.4.2 特殊访问权限 320
 - 14.4.3 剖析“禁止访问” 321
 - 14.5 分析访问控制引起的提权漏洞 327
 - 14.6 各种关注的对象类型的攻击模式 328
 - 14.6.1 针对服务的攻击 328
 - 14.6.2 针对Windows注册表DACL的攻击 334
 - 14.6.3 针对目录DACL的攻击 337
 - 14.6.4 针对文件DACL的攻击 342
 - 14.7 其他对象类型的枚举方法 346
 - 14.7.1 枚举共享内存段 346
 - 14.7.2 枚举命名管道 347
 - 14.7.3 枚举进程 347
 - 14.7.4 枚举其他命名的内核对象(信号量、互斥锁、事件、设备) 348
 - 14.8 本章小结 349
 - 14.9 扩展阅读 349
- 第15章 攻击Web应用程序 351
- 15.1 概述十大Web漏洞 351
 - 15.2 MD5哈希注入 352
 - 15.2.1 实验15-1：注入哈希 352
 - 15.3 多字节编码注入 357
 - 15.3.1 理解漏洞 357
 - 15.3.2 实验15-2：利用 357

多字节编码 358
15.4 捕捉跨站脚本攻击(XSS) 362
15.4.1 实验15-3：JavaScript块
中的基本XSS注入 363
15.5 Unicode规范化形式攻击 364
15.5.1 实验15-4：利用
Unicode规范化 364
15.5.2 Unicode规范化简介 365
15.5.3 规范化形式 366
15.5.4 准备好测试的环境 367
15.5.5 通过x5s插件执行
XSS测试 368
15.5.6 手动发起攻击 369
15.5.7 添加自己的测试用例 370
15.6 本章小结 371
15.7 参考文献 372
15.8 扩展阅读 372
第16章 攻击IE：堆溢出攻击 373
16.1 设置环境 373
16.1.1 WinDbg配置 373
16.1.2 将浏览器附加到
WinDbg 374
16.2 堆喷射简介 374
16.3 使用HTML5喷射 376
16.3.1 实验16-1：使用HTML5
执行堆喷射 377
16.4 DOM元素属性喷射(DEPS) 379
16.4.1 实验16-2：使用DEPS
技术的堆喷射 380
16.5 HeapLib2技术 382
16.5.1 通过耗尽缓存块来
强制执行新的分配 383
16.5.2 实验16-3：HeapLib2
喷射 383
16.6 使用字节数组的Flash喷射 384
16.6.1 实验16-4：使用Flash
执行基本的堆喷射 385
16.7 使用整数向量的Flash喷射 386
16.7.1 实验16-5：使用Flash
向量的堆喷射 385
16.8 利用低碎片堆(LFH) 388
16.9 本章小结 389
16.10 参考文献 389
16.11 扩展阅读 389
第17章 攻击IE：释放后重用技术 391
第18章 使用BeEF进行高级
客户端攻击 409
18.1 BeEF基础 409
18.1.1 实验18-1：设置BeEF 409
18.1.2 实验18-2：使用BeEF
控制台 411
18.2 挂钩浏览器 414
18.2.1 实验18-3：基本的
XSS挂钩 414

18.2.2 实验18-4：使用网站
欺骗挂钩浏览器 415
18.2.3 实验18-5：使用shank
自动注入挂钩 417
18.3 使用BeEF获得指纹 419
18.3.1 实验18-6：使用BeEF
获得浏览器指纹 419
18.3.2 实验18-7：使用BeEF
获得用户指纹 420
18.3.3 实验18-8：使用BeEF
获得计算机指纹 421
18.4 攻击浏览器 423
18.4.1 实验18-9：使用BeEF和
Java来攻击浏览器 423
18.4.2 使用BeEF和Metasploit
攻击浏览器 426
18.5 自动化攻击 430
18.6 本章小结 432
18.7 扩展阅读 432
第19章 基于补丁比较的1-day
漏洞开发 433
第III部分 高级恶意软件分析
第20章 剖析Android恶意软件 457
第21章 剖析勒索软件 475
第22章 分析64位恶意软件 495
22.1 AMD64架构概述 495
22.2 解密C&C服务器 498
22.3 本章小结 511
22.4 扩展阅读 511
第23章 下一代逆向工程 513
23.2.1 免费的动态分析工具 523
23.2.2 商业替代品：TrapX
Malware Trap 524
23.3 本章小结 527
23.4 参考文献 527
23.5 扩展阅读
• • • • • (收起)

[灰帽黑客\(第4版\) 下载链接1](#)

标签

黑客

信息安全

网络安全

安全

计算机

渗透测试

Security

灰帽黑客

评论

浏览了前半部分，书的作者貌似很牛逼，然后一查发现一位女性作者已经过世了...好吧，我老实评价一下，首先我是个门都没入的新手，这本书也没讲解啥先验知识，上来就是教你一套工具，原理都没说清，代码列出来解释也不多，每一个专题讲得零散空洞，本是很深奥的东西那么十多页讲完了，一看尽是些工具和一些不明所以的代码演示，不符合我深入浅出的预期。建议新手就别浪费时间看它了，我是实在忍不住才放弃看的，我也不知道这本书是否是新手定位，我甚至觉得拿来做科普书都不够格，至于其他人为什么给可以5分，我表示很好奇

不知攻，焉知防。技术的演进让攻防日益激烈，有如生物界病毒与免疫系统的交替进化。技术讲解很全面，值得一读。

非常不错

[灰帽黑客\(第4版\) 下载链接1](#)

书评

书中主要讲理论 还有一些相关法律 道德约束之类的东西 后半段讲的比较深
按作者的话说 “本篇假设读者已经具备相关经验 对各类工具比较熟悉的前提下” 所以
建议没基础的 只是对书名感兴趣的就不用读了

这书貌似2007年出版的，到2011年才绝版，这本书科海算是赚了，呵呵；
这本书其实很不错，比较容易上手，有能深入进去，可谓深入浅出；
这本书的第二版貌似也已经出了，不过技术性的篇幅下降不少；
科海现在好像不运营了？反正看不到出书了

没做过类似的事情，也不从事这样工作

前面有豆友的观点我同意，这本书理论性太强。总感觉作者是很长时间的经验积累总结出来的，而且不符合我们这边人的使用习惯 书还是不错，比较适合收藏

[灰帽黑客\(第4版\) 下载链接1](#)