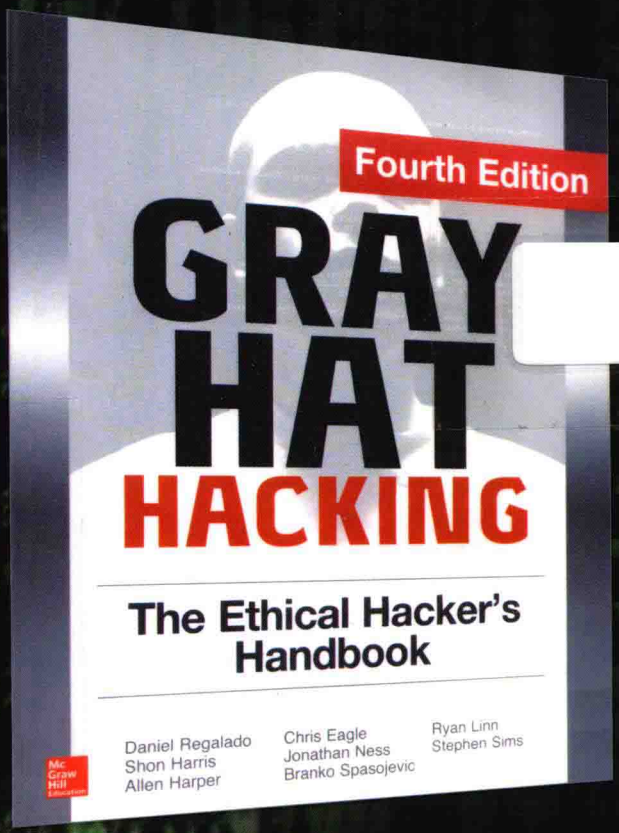
Mc Graw Hill Education

安全技术经典译丛

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

Gray Hat Hacking, The Ethical Hacker's Handbook, Fourth Edition



[美] Daniel Regalado
Shon Harris
李枫
等著
译

清华大学出版社

安全技术经典译丛

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

[美] Daniel Regalado Shon Harris 等著

李 枫 译

清华大学出版社

北 京

Daniel Regalado, Shon Harris, Allen Harper, Chris Eagle, Jonathan Ness, Branko Spasojevic, Ryan Linn, Stephen Sims
Gray Hat Hacking, The Ethical Hacker's Handbook, Fourth Edition
EISBN: 0-07-183238-6

图书在版编目(CIP)数据

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 /(美)里加拉多(Regalado, D.)，哈里斯(Harris, S.) 等著；李枫 译. —北京：清华大学出版社，2016
(安全技术经典译丛)
书名原文：Gray Hat Hacking, The Ethical Hacker's Handbook, Fourth Edition
ISBN 978-7-302-42867-1

Ⅰ. ①灰… Ⅱ. ①里… ②哈… ③李… 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028859号

责任编辑：王 军 韩宏志
装帧设计：孔祥峰
责任校对：成凤进
责任印制：刘海龙

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦A座 邮 编：100084
社 总 机：010-62770175 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经 销：全国新华书店
开 本：185mm×260mm 印 张：34.25 字 数：861千字
版 次：2016年3月第1版 印 次：2016年3月第1次印刷
印 数：1～3500
定 价：79.80元

产品编号：063324-01

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
6.1 用户空间 shellcode ····· 125
6.1.1 系统调用 ····· 125
6.1.2 基本 shellcode ····· 126
6.1.3 端口绑定 shellcode ····· 126
6.1.4 反向 shellcode ····· 128
6.1.5 查找套接字 shellcode ····· 129
6.1.6 命令执行代码 ····· 130
6.1.7 文件传输代码 ····· 130
6.1.8 多级 shellcode ····· 130
6.1.9 系统调用代理 shellcode ····· 131
6.1.10 进程注入 shellcode ····· 131
6.2 其他 shellcode 考虑因素 ····· 132
6.2.1 shellcode 编码 ····· 132
6.2.2 自我破坏 shellcode ····· 133
6.2.3 反汇编 shellcode ····· 134
6.3 内核空间 shellcode ····· 135

6.4 本章小结……136
6.5 参考文献……136
6.6 扩展阅读……137
第 7 章 编写 Linux shellcode……139
7.1 基本的 Linux shellcode……139
7.1.1 系统调用……139
7.1.2 使用 C 进行系统调用……140
7.1.3 使用汇编语言进行系统调用……141
7.1.4 exit 系统调用……141
7.1.5 setreuid 系统调用……143
7.1.6 利用 execve 实现创建 shell 的 shellcode……144
7.2 实现端口绑定 shellcode……147
7.2.1 Linux 套接字编程……147
7.2.2 用汇编程序创建套接字……150
7.2.3 测试 shellcode……152
7.3 实现反向连接 shellcode……155
7.3.1 反向连接的 C 代码……155
7.3.2 反向连接的汇编程序……156
7.4 shellcode 编码……158
7.4.1 简单的异或编码……158
7.4.2 编码后 shellcode 的结构……158
7.4.3 JMP/CALL XOR 解码器示例……159
7.4.4 FNSTENV XOR 示例……160
7.4.5 将代码整合起来……162
7.5 利用 Metasploit 自动生成 shellcode……164
7.5.1 利用 Metasploit 生成 shellcode……164
7.5.2 利用 Metasploit 对 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 ProSSHD 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：利用多字节编码……358
15.4 搜捕跨站脚本攻击(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
17.1 释放后重用概述……391
17.2 分析释放后重用攻击(UAF)…394
17.3 利用 UAF 漏洞……402
17.4 本章小结……407
17.5 参考文献……407
17.6 扩展阅读……408
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
19.1 有关二进制比较的介绍……433
19.1.1 应用程序比较……433
19.1.2 补丁比较……434
19.2 二进制比较工具……434
19.2.1 BinDiff……435
19.2.2 turbodiff……436
19.2.3 实验 19-1：首次文件比较……438
19.3 补丁管理流程……440
19.3.1 微软周二补丁……440
19.3.2 实验 19-2：获得并提取微软补丁……441
19.3.3 检查补丁……443
19.3.4 实验 19-3：使用 turbodiff 比较 MS14-006……445
19.3.5 内核调试……447
19.3.6 实验 19-4：内核调试 MS14-006……451
19.4 本章小结……454
19.5 参考文献……454
19.6 扩展阅读……454
第III部分 高级恶意软件分析
第 20 章 剖析 Android 恶意软件……457
20.1 Android 平台简介……457
20.1.1 Android 应用程序包……457
20.1.2 应用程序清单……459
20.1.3 分析 DEX……460
20.1.4 Java 反编译……462
20.1.5 DEX 反编译……463
20.1.6 DEX 反汇编……465
20.1.7 练习 20-1：在模拟器中运行 APK……466
20.2 恶意软件分析……468
20.2.1 恶意软件分析入门……468
20.2.2 练习 20-2：运用 Droidbox 进行黑盒 APK 监控……471
20.3 本章小结……472
20.4 扩展阅读……473
第 21 章 剖析勒索软件……475
21.1 勒索软件的历史……475
21.2 赎金支付选项……476
21.3 剖析 Ransomlock……476
20.2.1 实验 21-1：动态分析……477

20.2.2 实验 21-2：静态分析 ……479
21.4 CryptoLocker……491
21.5 本章小结 ……493
21.6 扩展阅读 ……493
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
23.1 著名的 IDA 插件……513
23.1.1 IDAscope ……513
23.1.2 IDA Toolbag ……519
23.1.3 协作……522
23.2 基于 TrapX 的蜜罐和沙箱技术……523
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……527

第Ⅰ部分

速成课：备战

第1章　道德黑客和法律制度
第2章　编程技能
第3章　静态分析
第4章　使用IDA Pro进行高级分析
第5章　模糊测试的世界
第6章　shellcode策略
第7章　编写Linux shellcode

第 1 章　道德黑客和法律制度

本书的写作目的不是为了介绍如何进行恶意的破坏性活动，而是为了教会你如何抵御此类攻击和破坏性行为，以帮助你扩大知识范围，完善自己的技能。

本章将讨论以下主题：

- 理解敌方策略的意义
- 道德黑客
- 网络相关法律的兴起
- 发掘漏洞

1.1　理解敌方策略的意义

了解攻击模式是防务安全最具挑战性的方面。通过了解黑客的思考和操作方式，可以更好地调整组织以防范新近兴起的威胁和趋势。若不测试攻击防御系统，破坏分子将成为唯一测试你网络的人。通过学习进攻安全，就能够测试防御系统，并找出问题和差距所在。

关于黑客群体，值得注意的一点在于他们总是在变化。在过去几年中，黑客的作案动机已经从找出利用漏洞的方法从而获得兴奋刺激感变成了通过入侵或者利用自己的入侵技能而获利。在很大程度上，只是为了好玩而没有特定目标的黑客已经被想要专门通过入侵行为获得物质利益的黑客取代了。攻击不仅变得更具针对性，而且日趋复杂。在这种趋势下，世界上发生了越来越多的类似事件，以下是几个比较典型的案例：

- 2013年10月，黑客入侵Adobe盗取3800万份账户凭据及经过加密的信用卡号码，部分账户信息泄露于互联网上[1]。
- 2013年7月，Harbor Freight遭受恶意软件攻击，超过400家门店的信用卡数据被窃取，这是一起恶意软件在电商平台上窃取大量信用卡信息的典型案例。[2]
- 2013年5月，Ponemon Institute发布了由赛门铁克赞助的报告，指出在美国，每个网络攻击可造成每家公司平均$188的损失[3]。报告还指出，网络攻击造成超过 28 000条信息泄露，也就意味着黑客谋取的利益，受害公司要花更多的钱来弥补损失。
- 2013年圣诞购物狂欢季，Target公司遭受到迄今为止最大的网络攻击，使得大约40 000至70 000人员遭受财产损失。Target公司赶在新闻报道前积极做出响应，帮助客户了解网络攻击以及公司的应对方案。Target公司还专门设立在线网站介绍新的安全措施及如何防范信用卡诈骗。[4]

据Gartner的保守估计，计算机网络的瘫痪可导致平均每小时$42 000的损失[5]。如果全年遭受175小时以上的停工瘫痪，则会使公司蒙受超过700万美元的损失。即使攻击的影响不够大，新闻上没有报道，安全业界也没有讨论，但它们仍然会对公司的利润造成影响。

黑客并不都是出于获利目的发动网络攻击，有些出于政治目的，这部分攻击称为黑客行

为(hactivism)。合法和非法的方法都可以用来描述政治思想。试图通过技术来影响社会改革合法吗？在言论自由的幌子下，Web是不是变得面目全非？在提供非法内容的网站上进行虚拟的“静坐抗议”是错误行为吗？2009年伊朗大选，个人建立网站对潜在腐败的政府选举发泄不满不道德吗？以色列入侵加沙，很多网站都遭到攻击、拒绝服务(Denial-of-Service，DoS)攻击及网页劫持的事件。对正义与否的判断可能与每个人所处的利益位置有关。

一些黑客还会创建和销售零日漏洞攻击(zero-day attack)。零日漏洞攻击当前还没有对应的修复补丁。无论是谁，只要运行包含可攻击漏洞的软件，就会被暴露，此时，保护微乎其微，甚至没有保护。创建这类攻击的黑客会在一些网站上做广告，并将其销售给其他黑客或者有组织的犯罪团伙。

识别攻击

网络管理员、网络工程师和安全从业者需要有能力弄清楚是否有人正在进行攻击或即将进行攻击。当攻击发生时识别攻击看似简单。然而，它只对“动静非常大”或者影响比较严重的攻击，如拒绝服务(Denial-of-Service，DoS)攻击，才是成立的。许多攻击者都会隐藏自己，不被安全设备和负责安全的人员注意到。因此，知道如何进行不同的攻击是很重要的，这样就能正确地识别和阻止它们。

知道何时即将遭受攻击也很重要。如果网络管理人员接受过关于攻击技术的培训，在看到ping 扫描后的第二天又看到了端口扫描，那么他们就可以知道很可能不久后系统会遭到攻击。不同的活动会导致不同的攻击，所以理解这些活动可以帮助公司保护自身。有人可能辩称，现在有很多自动化的安全产品可以识别这类活动，所以我们不需要观察何时将发生攻击。但是软件没有能力将各种活动放到合适的情境中，也无法做出相应的决策，因此依赖软件是一种危险的做法。在计算和执行重复性任务方面计算机比人更有优势，但是因为生活中并不是非黑即白，也不会只是以1和0看待事物，所以能够根据自己的判断做出决定是人类强于计算机的优势所在。

因此，黑客工具只是一些软件工具，通过执行特定类型的过程实现预期的结果，理解这一点十分重要。工具既可以用于正当(防御)用途，也可以用于不当(攻击)用途。好人和坏人使用的是完全相同的工具集，区别在于使用这些工具时他们的意图是什么。如果安全从业者想让自己对客户和业界有用，就必须理解如何使用这些工具，以及如何执行攻击。

1.2　正义黑客过程

正义黑客过程组织为自保就应了解黑客的影响和能力，相应的，他们会雇佣正义黑客，也就是我们所熟知的渗透测试人员，模拟发动网络攻击。渗透测试人员则会利用他们所掌握的攻击技术，尽可能模仿真正的黑客，进行不造成任何损失的网络攻击。它们能使组织更好地保护自己免受攻击，客户和有抱负的黑客也应熟知整个流程。

通过定义渗透测试活动、阶段和步骤，你可以设置自己(即渗透测试人员)和客户之间的预期。客户既可以是组织外部客户，也可以是内部客户。无论测试对象和测试目的为何，测试范围的设定及通用语言的使用有助于减少误解、优化流程，更好地帮助相关方了解你的测

试内容及测试目的。

描述渗透测试过程前，我们需讨论渗透测试(penetration testing)和漏洞评估(Vulnerability assessment)的差异。这两种活动有着同样的目标很容易混淆。在漏洞评估期间，使用某些类型的自动化扫描产品可以探测某个IP 地址范围内的端口和服务。大多数这样的产品也可以测出所运行的操作系统和应用软件的类型、版本、补丁级别、用户账户和运行的服务。检测结果将与所用产品的数据库中的对应漏洞进行匹配。最终将得到一堆报告，其中列出了每个系统中存在的漏洞和可降低风险的相应措施。基本上，这个工具会这样声称：这里列出了你的系统中存在的漏洞，你可以采取这里列出的措施来修复这些漏洞。

漏洞扫描的缺点是即使知道漏洞的严重性，也无法预估其影响，因而有必要进行渗透测试。漏洞扫描可识别出某软件存在某些可利用的漏洞，随后渗透测试就能进一步利用这些漏洞，获取敏感信息。漏洞扫描大多基于版本和侵入式检查指出漏洞所在，而渗透测试则可指出漏洞扫描结果的正确与否。

当正义黑客执行渗透测试时，他们的最终目标是入侵一个系统，然后从一个系统入侵另一个系统，直到“占领”整个域或环境。与漏洞评估不同，渗透测试不会随漏洞的识别而中止。渗透测试人员会权衡查找到的漏洞直到他们“占领”整个域或环境。所谓“占领”，是指他们在最关键的Unix或Linux 系统上拥有root 权限，或者取得了可以访问和控制网络上的全部资源的管理员账户。这么做的目的是为了向客户(公司)展示在网络的当前环境和安全配置下，真正的攻击者可以采取哪些行动。

很多时候，正义黑客在按自己的工作过程获得对网络的控制的同时，还会收获一些战利品。这些战利品包括CEO的密码、公司的商业机密文件、所有边界路由器的管理员密码、CFO和CIO的笔记本电脑中标记为“机密”的文档等。顺便收集这些战利品的目的是使决策者理解漏洞的危害并引起重视。否则，就算花费几个小时向CEO、CIO或COO解释有关服务、开放端口、不当配置和可能被攻击之处，他们也不能很形象地理解安全问题并引起足够的重视。但是一旦向CFO展示其下一年度的规划，向CIO展示其下一年度的生产线蓝图，或者说出CEO的密码“IAmWearingPanties”，他们自然而然就想更多地了解防火墙和其他应该就位的防护措施的重要性了。

警告：任何安全从业者都不应该嘲笑客户，或者让客户觉得自己对安全的认识太过匮乏，因为如果客户什么都懂，他们也就没必要寻求安全从业者的帮助了。安全从业者是来帮助解决问题的，而不是来指手画脚的。另外，在大多数情况下，负责渗透测试的团队都不应该读取任何敏感数据，以避免在将来发生由于使用公司的机密信息而导致的法律诉讼。

本书将介绍高级的漏洞修复工具和方法，以及复杂的渗透技术。然后将深入研究程序代码，以展示熟练的攻击者是如何找出漏洞并通过开发工具来利用这些漏洞的。接下来首先介绍一下正义黑客的渗透测试过程，以及该过程和黑客活动的区别。

1.2.1　渗透测试过程

一旦网络管理员、工程师和安全专家了解黑客的工作，就能模仿他们的活动，进行渗透

测试。但是，为什么会有人想模仿攻击？因为这是真正测试环境安全级别的唯一途径。当真正的攻击发生时，你必须知道应如何做出响应。

本书有助于帮助你分步骤理解许多攻击类型是如何发生的。它有助于开发模仿类似活动以测试公司安全状况的策略与方法。

万一你无意间运用本书中提供的信息从事恶意活动，本章后半部分所涉及的联邦法律会让你迷途知返，远离违法犯罪行为。如今的法律体系已将计算机犯罪列为重点犯罪领域，黑客们正在为他们的行为支付巨额罚款并面临牢狱之灾。希望你不要成为他们中的一员。人生充满了乐趣，拥有的聪明才智足以让你做个好人，完全没有坐牢的风险！为何要选择那样一条邪路？

渗透测试者的测试动机由客户所驱动。无论是访问敏感信息、为正在进行的项目提供额外的正当理由，抑或是测试组织的安全，启动测试前理解客户需求非常重要。一旦理解其目标，引导测试阶段的剩余部分就会变得容易得多。让我们先来看看渗透测试的典型步骤。

1. **基本规则**　建立基本规则：

- 设置测试人员与客户期望以及联系信息
- 确定测试相关方
- 设置开始、终止日期以及管制期
- 获取正式批准并签订关于适用范围、签名和法律要求的书面协议，通常称为工作申明(SOW)。

提示：在测试过程中将这份文件放在手边，必要时它可以作为“免罪金牌”。

2. **被动扫描**　收集尽可能多的关于目标的信息，同时使渗透测试人员和目标之间保持零接触。被动扫描过程(又可称为开源智能 (OSINT))中可包括：

- 社交网站
- 在线数据库
- 谷歌，Monster.com等
- 垃圾搜寻

3. **主动扫描和枚举**　使用扫描工具查找目标公共接口，这些工具包括：

- 商业扫描工具
- 网络映射
- 标志提取
- 战争拨号
- DNS 区域传送
- 流量嗅探
- 无线战争驾驶攻击

4. **指纹识别**　彻底探查目标系统以确定：

- 操作系统类型和补丁级别
- 应用程序和补丁级别