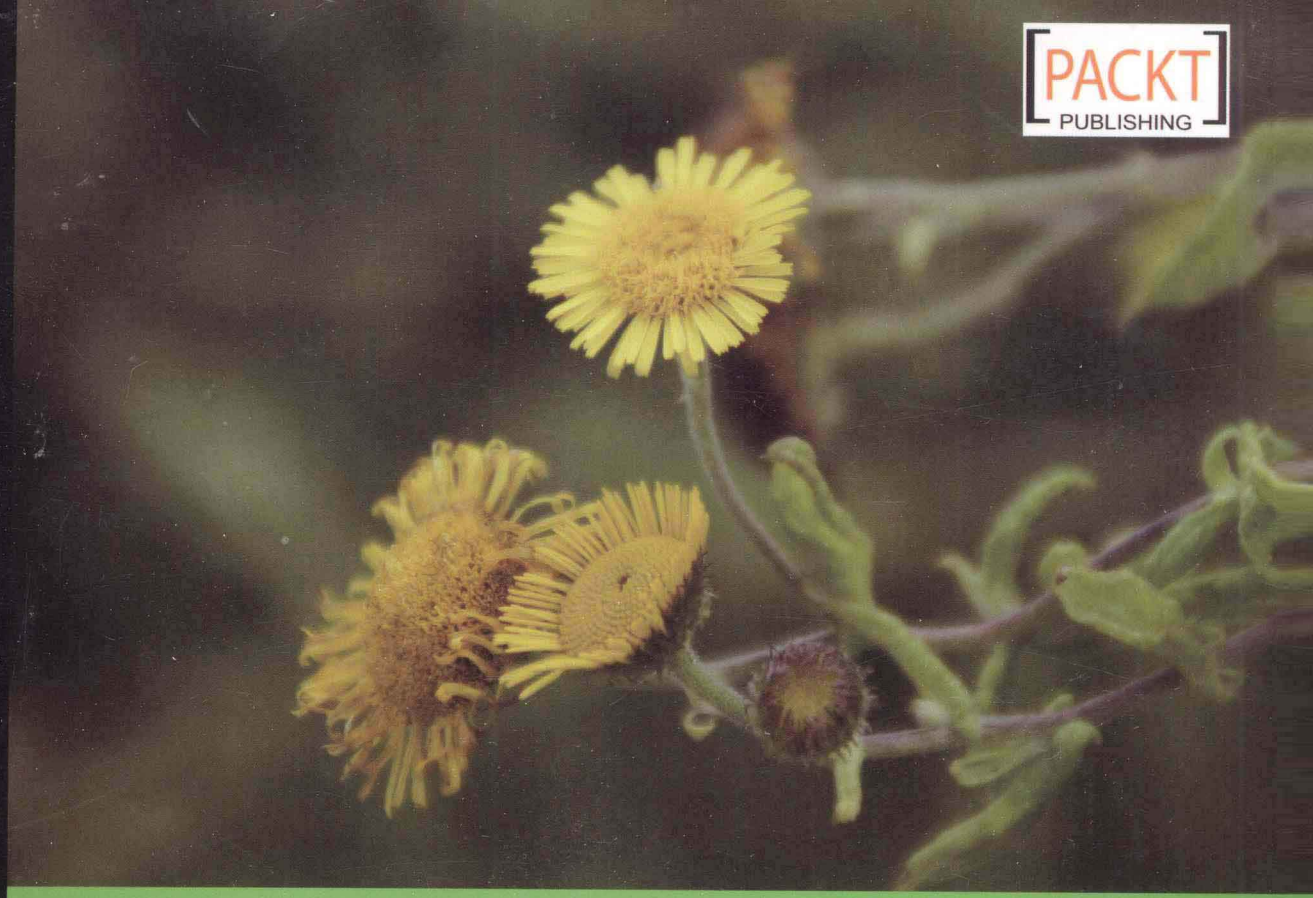


Metasploit
渗透测试手册

Metasploit Penetration Testing Cookbook

[印度] Abhinav Singh 著
王一 译

人民邮电出版社
POSTS & TELECOM PRESS

Metasploit
渗透测试手册

[印度] Abhinav Singh 著
王一 译

人民邮电出版社
北京

图书在版编目（CIP）数据

Metasploit渗透测试手册 / （印）辛格（Singh, A.）著 ; 王一译. -- 北京 : 人民邮电出版社, 2013.9
ISBN 978-7-115-32383-5

Ⅰ. ①M… Ⅱ. ①辛… ②王… Ⅲ. ①计算机网络－安全技术－应用软件－手册 Ⅳ. ①TP393.08-62

中国版本图书馆CIP数据核字(2013)第135566号

版 权 声 明

◆ 著 [印度] Abhinav Singh
译 王 一
责任编辑 傅道坤
责任印制 程彦红 杨林杰

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：15
字数：305 千字 2013 年 9 月第 1 版
印数：1－3 000 册 2013 年 9 月北京第 1 次印刷

著作权合同登记号 图字：01-2012-7222 号

定价：49.00 元

读者服务热线：(010)67132692 印装质量热线：(010)67129223
反盗版热线：(010)67171154

内容提要

本书是一本介绍渗透测试的安全类技术书籍，全书以 Metasploit 这一最流行的渗透测试框架为演示和实验工具，内容由浅入深，易于理解，同时具有极强的可操作性与实用性。

本书总共分为 10 章，前两章对 Metasploit 及信息收集与扫描进行简单介绍；第 3 章介绍使用 Metasploit 对操作系统漏洞进行攻击渗透；第 4 章介绍使用 Metasploit 进行客户端漏洞攻击和防病毒软件规避；第 5 章、第 6 章介绍非常重要的 Meterpreter，并演示了利用该工具探索已攻陷目标机器的情况；第 7 章、第 8 章分别介绍框架中模块和漏洞利用代码的使用问题；第 9 章介绍 Armitage；第 10 章介绍社会工程工具包的使用问题。

本书作为 Metasploit 渗透测试技术手册，适合于渗透测试人员、网络安全管理人员、信息安全专业的学生及对信息安全感兴趣的读者阅读。

关于作者

Abhinav Singh 是来自印度的一位信息安全专家，年轻有为。他对破解和网络安全领域有着浓厚的兴趣。他以自由职业者的身份积极服务于多家安全公司，为它们提供咨询服务。当前，他是印度 Tata Consultancy Services 公司的一名系统工程师。他因其博客（http://hackingalert.blogspot.com）而被人们所熟知，他在其博客中与他人分享了解决破解和网络安全问题的经验。Abhinav 的文章已经被多家技术杂志和门户网站所引用。

我要感谢我的父母，谢谢他们一直以来对我的支持和信任。我还要感谢我的姐姐，作为我的医生，她对我悉心照料。谢谢 Sachin Raste 先生，他为审校本书付出了宝贵的精力。谢谢 Kanishka Khaitan，他是我最完美的榜样。我还要感谢我的博客读者们，他们向我提供了宝贵的建议和意见。最后，我要感谢 Packt Publishing 出版社，与他们的默契合作，让我终生难忘。

关于审稿人

Kubilay Onur Gungor 当前以 Web 应用安全专家的身份供职于 Sony 欧洲公司，他还是 Sony 欧洲和亚洲地区的事件经理。

他在 IT 安全类领域已经工作了 5 年多的时间。独立在安全领域研究了一段时间以后，他凭借图像的密码分析（即使用混乱的逻辑图来加密）开始了其安全职业生涯。通过在 Isik 大学数据处理中心的工作，他在网络安全领域积累了大量经验。在 Netsparker 担任 QA 测试人员工作期间，他开始进入渗透测试领域，并为土耳其的一家领先的安全公司工作。他曾经为很多大型客户（例如银行、政府机构、电信公司）的 IT 基础设施进行多次渗透测试。他还为多家软件厂商提供了安全咨询服务，以帮助他们维护软件安全。

Kubilay 还一直在研究多学科的网络安全方法，其中包括犯罪学、冲突管理、感知管理、恐怖主义、国际关系和社会学。他还是 Arquanum 多学科网络安全研究学会（Arquanum Multidisciplinary Cyber Security Studies Society）的创始人。

Kubilay 经常以发言人的身份参与安全会议。

Kanishka Khaitan 是印度普纳大学计算机应用专业的一名硕士研究生，她在瓦拉纳西印度大学获得了数学专业的荣誉学位。在过去的两年里，她一直在 Amazon 的 Web 领域工作。而在此之前，她参与了 Infibeam（一家位于印度的在线零售公司）为期 6 个月的实习生项目。

Sachin Raste 是一位著名的安全专家，在网络管理和信息安全领域有 17 年以上的工作经历。他与其团队为印度的一些大型商业机构设计过网络和应用，并将它们与 IT 流程以流水化的形式集成起来，从而实现业务的连贯性。

他当前以自身安全研究人员的身份与 MicroWorld（信息安全解决方案电子扫描范围[eScan range]的开发团队）一起工作。他设计并开发了一些开创性的算法用来检测和预防恶意软件和数字欺诈，从而保护网络免于黑客和恶意软件的攻击。Sachin Raste 在其专业领域内还发表了多篇白皮书，并出席了许多以“宣传防止数字欺诈，增强防范”为主题的电视

节目。

与 MicroWorld 一起工作的经历也提升了 Sachin 的技术技能，从而使其可以跟上信息安全业界的当前趋势。

首先，我要特别感谢我的妻子和儿子，以及为我提供帮助的密友们。正是因为你们的存在，世间一切之事才有了可能。谢谢来自MicroWorld及其他单位的同事们，谢谢你们能够耐心地聆听，并帮助我成功完成了许多复杂的项目；与你们的合作令人愉快而难忘。感谢我的老板——MicroWorld的MD——他给了我足够的自由和时间来探索自己的未知。

感谢你们！

献词

谨将本书献给我的祖父母，感谢他们的祝福。将本书献给我的父母和姐姐，感谢他们的支持和鼓励。还要将本书献给我的密友 Neetika，他是我永不止步的动力。

前言

对当前环境下的网络安全而言，渗透测试是核心工作之一。渗透测试通过进行实质意义上的入侵式安全测试，对目标的安全性进行完全分析，这有助于识别目标系统主要组件中硬件或软件方面的潜在弱点（即安全漏洞）。渗透测试之所以重要，是因为其有助于从黑客的视角来识别目标系统的威胁与弱点，并且在发现目标中存在的安全漏洞之后，可以实时地对其进行渗透利用以评估漏洞的影响，然后采用适当的补救措施或打补丁，以便保护系统免遭外部攻击，从而降低风险因素。

决定渗透测试可行性的最大因素是对目标系统相关信息的了解情况。在不具备目标系统先验知识的情况下，就只能实施黑盒测试。在黑盒测试工作中，渗透测试人员只能“白手起家”，一点一滴地收集目标系统的相关信息。而在白盒测试中，测试人员已全面掌握目标系统的相关信息，此时需要做的工作是识别目标系统中存在的已知（或未知）弱点。这两种渗透测试方法都有相当的难度，并且每种环境都会有特定的需求。业界专家提炼了一些关键步骤，这些步骤对几乎所有形式的渗透测试都是至关重要的，包括以下几点。

- **目标发现与枚举**：识别目标，收集目标相关的基本信息，但不与目标建立任何形式的物理连接。
- **漏洞识别**：通过扫描、远程登录、网络服务等多种方法，统计出目标系统中运行的软件和提供的服务。
- **漏洞利用**：对目标系统软件或服务中存在的已知或未知漏洞进行利用。
- **漏洞利用后的控制程度**：成功地进行漏洞利用后，攻击者在目标系统中具备的访问控制权限级别。
- **报告**：针对发现的漏洞及其可能的应对措施提出建议。

这些步骤看起来很简单，但事实上，要对运行着大量服务的高端系统进行全面的渗透测试，需要花费数天甚至数月的时间才能完成。渗透测试之所以是一项耗时的任务，原因

在于渗透测试以“试错法”技术作为基础。对漏洞的渗透与利用依赖于大量的系统配置要素，如果不去实践尝试，就不可能确定某一个特定的漏洞是否能够成功利用。试想一下，以对运行着 10 项服务的 Windows 操作系统进行漏洞利用为例，渗透测试人员必须对这 10 种不同服务中是否存在已知漏洞进行全面的分析与识别。而且在识别之后，才能开始漏洞利用的过程。这还只是仅需要考虑一个系统的小型场景，如果面对的是包含大量类似系统的整个网络，我们又该怎样逐一地对其进行测试呢？

这就是渗透测试框架发挥作用的地方。渗透测试框架可以将多个测试过程进行自动化实现，例如网络扫描、基于可用服务及其版本信息的漏洞识别、自动式漏洞利用等。渗透测试框架为测试人员提供了一个全面的控制面板，测试人员可以借助控制面板对所有测试活动进行有效的管理，同时还可以对目标系统进行有效监控，从而加快渗透测试进程。渗透测试框架的另一个优势是报告生成。利用渗透测试框架，可以自动保存渗透测试结果，并生成测试报告以备后续使用，或者与远程工作的其他人员共享。

本书旨在帮助读者掌握当前应用最为广泛的测试框架之一——Metasploit。Metasploit 框架是一个开源平台，有助于创建实用型漏洞利用工具，并提供了渗透测试需要的其他核心功能。本书将带领读者畅游 Metasploit 世界，并介绍怎样使用 Metasploit 进行有效的渗透测试。此外，本书还将涉及 Metasploit 框架之外的其他一些扩展工具，并讨论怎样提高其功能以便提供更好的渗透测试体验。

本书内容

第 1 章，给安全专业人员的 Metasploit 快速提示，将带领读者初探 Metasploit 与渗透测试，对 Metasploit 框架及其体系结构、库等内容进行初步认识。要使用 Metasploit 框架进行渗透测试，需要先对其进行安装，本章将介绍怎样使用虚拟机构建自己的渗透测试环境。然后讨论怎样在不同的操作系统上进行安装，最后对 Metasploit 的使用进行初步尝试，并对其使用界面进行介绍。

第 2 章，信息收集与扫描，这是渗透测试的第一步，本章从最传统的信息收集方式开始，然后介绍怎样使用 Nmap 进行高级扫描。本章在内容上还涵盖了一些其他工具，例如 Nessus 与 NeXope。与 Nmap 相比，NeXope 提供了一些额外的信息，从而弥补了 Nmap 的不足。最后，讨论 Dradis 框架，渗透测试人员广泛使用这一框架与远程的其他测试人员共享测试结果和报告。

第 3 章，操作系统漏洞评估与利用，主要讨论目标系统中运行的、尚未打补丁的操作

系统中漏洞的发现与利用。利用操作系统漏洞成功率高，并且操作简便。还讨论对几种流行的操作系统的渗透测试，例如 Windows XP、Windows 7 及 Ubuntu 等，包括这些操作系统中常见的、已知的一些漏洞，以及怎样在 Metasploit 中利用这些漏洞来突破目标机器。

第 4 章，客户端漏洞利用与防毒软件规避，讨论怎样使用 Metasploit 进行客户端漏洞利用的主题。本章在内容上涉及一些流行的客户端软件，例如 Microsoft Office、Adobe Reader 及 IE 浏览器。本章还进一步讨论如何规避或关闭客户端防病毒软件，以防止目标系统产生告警信息。

第 5 章，使用 Meterpreter 探索已攻陷目标，讨论漏洞利用成功后的下一个步骤。Meterpreter 是一款在漏洞利用成功之后使用的工具，包含一些功能，有助于在攻陷的目标机器中获取更多信息。本章还包括一些有用的渗透测试技术，例如权限提升、文件系统访问、键盘截获窃听等。

第 6 章，高级 Meterpreter 脚本设计，通过介绍构建自己的 Meterpreter 脚本、使用 API 组合工作等高级主题，本章使读者对 Metasploit 知识的认识进入一个新高度。通过本章的学习，读者可以更灵活地运用 Metasploit，因为可以根据渗透测试的实际场景，自己设计实用脚本，并将其融入到 Metasploit 框架中使用。本章还包括一些高级的“后渗透”概念，例如劫持、哈希注入及持续连接等内容。

第 7 章，使用模块进行渗透测试，本章将读者的注意力转移到 Metasploit 中另一个重要方面：模块。Metasploit 框架中收集整合了大量模块，不同的模块适用于不同的特定场景。本章包括 Metasploit 中一些重要的辅助性模块，也包括怎样构建自己的 Metasploit 模块。需要注意的是，准确理解本章内容需要一些关于 Ruby 脚本的基本知识。

第 8 章，使用漏洞利用代码，通过讨论怎样将任意的攻击代码转换为 Metasploit 模块，本章将终极武器加入到 Metasploit 库中。本章涉及一些高级主题，将向读者讲解怎样构建自己的 Metasploit 攻击代码，并在框架中进行使用。由于本章不可能涉及 Metasploit 框架中的所有漏洞利用代码，建议读者可以将本章作为手册，以便为 Metasploit 库之外的漏洞利用代码进行测试时提供参考。本章还涉及模糊测试模块，该模块可用于对任何漏洞构建自己的概念性验证代码。最后，本章以一个完整的实例作为结尾，包括怎样对一个应用程序进行模糊测试、怎样发现缓冲区溢出漏洞，以及怎样构建针对该漏洞的 Metasploit 模块。

第 9 章，使用 Armitage，简单讨论 Armitage，它是最流行的 Metasploit 模块之一。Armitage 为 Metasploit 框架提供了一个图形化界面，并提供一些点击式的漏洞利用选项增强 Metasploit 框架功能。本章重点关注 Armitage 的一些重要方面，例如快速发现漏洞、多目标处理、标签间移位，以及成功渗透后的处理等内容。

第 10 章，社会工程学工具包，这是本书的最后一章，介绍 Metasploit 框架中的另一个

重要扩展——社会工程学工具包（Social Engineer Toolkit，SET），用于生成利用目标用户的疏忽大意对目标进行渗透的测试用例。本章内容涉及 SET 相关的一些基本攻击手段，包括钓鱼攻击、网站攻击、USB 感染攻击等。

阅读本书的先决条件

为在阅读过程中重现和实践本书介绍的一些场景，读者需要准备两套系统，一套作为渗透测试实施系统，一套作为目标系统。另一种方法是，只需准备一套系统，之后使用虚拟化软件在其上建立测试环境。

此外，读者还需要准备一个 BackTrack 5 的 ISO 镜像文件，其中已包含预先安装的 Metasploit 和本书中讨论的其他工具。另一种方法是，从官方网站下载适合于读者的操作系统平台的 Metasploit。

本书读者对象

本书的目标读者既包括专业的渗透测试人员，也包括希望体验这一工具的 Metasploit 新手，书中包含了适合每个人的全部内容。本书采用了易于阅读、理解和场景再现的“食谱”结构，从初学者层次的渗透测试基础知识讲起，自然地过渡到专家级的高级知识和技能。因此，各个层次的读者都可以很容易地阅读和理解本书的内容。此外，本书需要读者具备扫描、漏洞利用和 Ruby 脚本的基本知识。

本书体例

提示框中的警告或重要提示以如此形式出现。

技巧与窍门则以这样的形式出现。

目录

第 1 章

给安全专业人员的 Metasploit 快速提示

本章讲解下述内容：

- 在 Windows 系统中配置 Metasploit；
- 在 Ubuntu 系统中配置 Metasploit；
- BackTrack 5 与 Metasploit 终极组合；
- 在单机上构建渗透测试环境；
- 在带有 SSH 连接的虚拟机上构建 Metasploit 环境；
- 从界面开始——Metasploit 的“Hello World”；
- 在 Metasploit 框架中建立数据库；
- 使用数据库存储渗透测试结果；
- 分析数据库中存储的渗透测试结果。

1.1 介绍

Metasploit 是当前信息安全与渗透测试领域最流行的术语，完全颠覆了已有的渗透测试方式。Metasploit 之所以如此受欢迎，是因为其所能执行的大部分任务可以简化渗透测试工作以使得系统更加安全。所有流行的操作系统都支持 Metasploit，并且 Metasploit 框架在这些系统上的工作过程也几乎是一样的。本书中的内容和示例主要以 BackTrack 5 操作系统为基础，因为该操作系统预装有 Metasploit 及在其上运行的其他第三方工具。

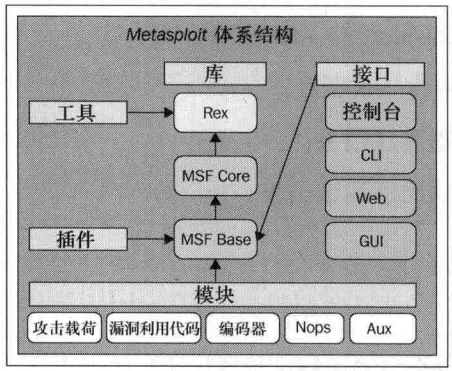
首先介绍 Metasploit 框架及与其相关的各种术语。

- **Metasploit 框架**：H.D.Moore 在 2003 年开发的一个免费的、开源的渗透测试框架，后来被 Rapid 7 公司收购。该框架目前的稳定版是使用 Ruby 语言开发的。Metasploit 框架包含了世界上最大且经过测试攻击的代码数据库，每年下载量超百万。该框架

也是迄今为止使用 Ruby 脚本语言构建的最复杂项目之一。

- **漏洞**：系统中存在的可能被攻击者或渗透测试人员用以破坏系统安全性的弱点。漏洞可能存在于操作系统中、应用软件中，甚至存在于网络协议中。
- **漏洞利用代码**：是攻击者或测试人员针对系统中的漏洞而设计的，用以破坏系统安全性的攻击代码。每个漏洞都有自己相应的攻击代码，Metasploit v4 中包含超过 700 个针对不同漏洞的漏洞利用代码。
- **攻击载荷**：完成实际攻击功能的代码，在成功渗透漏洞后会在系统上运行。攻击载荷最常见的用途是在攻击者和目标机器之间建立一个连接，Metasploit v4 中包含超过 250 个实现不同攻击功能的攻击载荷。
- **模块**：模块是组成完整系统的基本构建块。每个模块执行某种特定的任务，将若干模块组合成单独的功能主体可构成一个完整的系统。这种体系结构最大的优势在于，开发人员可以很容易地将新的漏洞利用代码和工具整合到 Metasploit 框架中。

Metasploit 框架采用的是模块式体系结构，漏洞利用代码、攻击载荷、编码器等都可以视为单独的模块。下图展示了 Metasploit 的体系结构。



进一步解释上图的内涵。

Metasploit 使用不同的库，这些库是保证 Metasploit 框架正确运转的关键。库实际上是预定义的任务、操作和功能的组合，框架中不同模块都可以使用这些库完成相应功能。Metasploit 框架最基本的组成部分是 Ruby 扩展库（Rex），Rex 提供的某些组件包含 wrapper socket 子系统、协议客户端与服务器、日志子系统、漏洞利用工具类及大量其他有用的类。Rex 本身在设计上是独立的组件，不像有些组件需要默认的 Ruby 安装。

MSF Core 库对 Rex 库进行了一些扩展，Core 主要负责实现所有与漏洞利用模块、会话和插件的接口。这一核心库由框架的基础库进行扩展，可提供简单的用于处理框架核心功