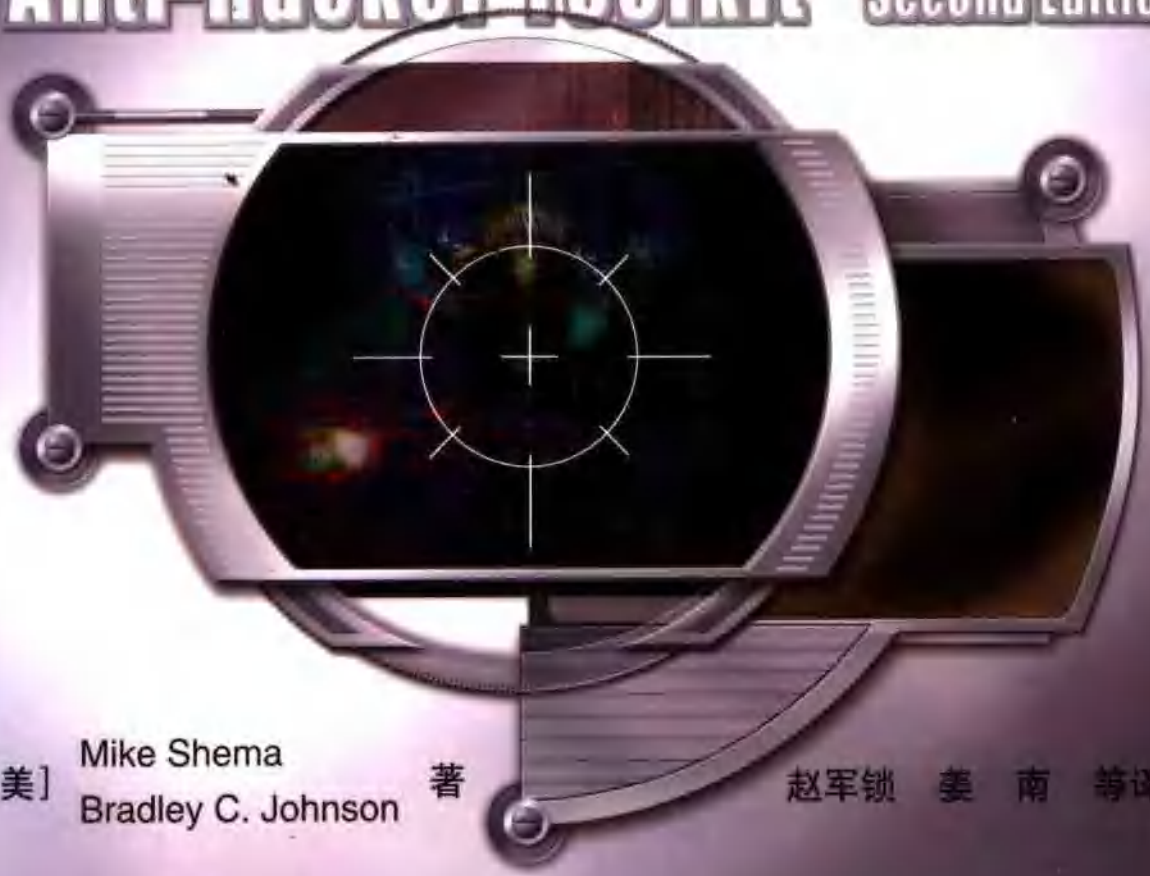
信息安全丛书

《黑客大曝光》姊妹篇

反黑客工具包

（第二版）

Anti-Hacker Toolkit Second Edition



[美] Mike Shema
Bradley C. Johnson 著

赵军锁 姜南 等译

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

信息安全丛书

反黑客工具包

（第二版）

Anti-Hacker Toolkit

Second Edition

[美] Mike Shema
Bradley C. Johnson 著

赵军锁 姜 南 等译

電子工業出版社
Publishing House of Electrouics Industry
北京 · BEIJING

内容简介

本书分类介绍了当前的IT专家所使用的计算机及网络安全工具，旨在帮助读者熟悉各种黑客与反黑客工具，使其能够更加高效且有效地选择合适的工具并最终出色地完成任务。每一章都以该章要讨论的工具的概述作为开始；然后详细描述了工具及其使用技术，包括如何用这些工具进行测试；最后，根据作者的实际经验在章节末尾给出“案例学习”，用以说明这些安全工具在现实世界中的使用，这也是本书的一个侧重点。全书共分四部分（即多功能工具、审计工具和主机防护工具、用于攻击和审计网络的工具以及用于取证和事件响应的工具），是对其第一版内容的更新和增强：更新了一些安全工具；增加了THC-Amap、THC-Hydra、Trinux、Kismet、Ettercap、Wellenreiter、WinHex、X-Ways Trace等一些新的工具；增加了Netcat、tcpdump、Ethereal、nmap、hping等工具的例子；增加了有关防火墙的内容。此外，本书的Web站点提供了关于最新的工具、工具信息、本书勘误和内容更新的链接。

作为一本面向安全的优秀技术图书，本书的特点是实用性强且技术含量高。适合安全管理员、网络管理员以及系统管理员阅读，也可作为网络和计算机安全专业相关技术人员的参考书。

Mike Shema, Bradley C. Johnson: **Anti-Hacker Toolkit, Second Edition.**

ISBN 0-07-223020-7

版权贸易合同登记号　图字：01-2004-2713

图书在版编目（CIP）数据

反黑客工具包（第二版）/（美）施玛（Shema, M.）等著；赵军锁等译. -北京：电子工业出版社，2005.6
（信息安全丛书）
书名原文：Anti-Hacker Toolkit, Second Edition
ISBN 7-121-01265-0

Ⅰ.反… Ⅱ.①施… ②赵… Ⅲ.计算机网络-安全技术 Ⅳ.TP393.08

中国版本图书馆CIP数据核字（2005）第049861号

责任编辑：杜闽燕
印　　刷：北京天竺颖华印刷厂
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
经　　销：各地新华书店
开　　本：787×1092　1/16　　印张：35.25　字数：902千字
印　　次：2005年6月第1次印刷
定　　价：55.00元

译 者 序

当网络渗透到我们生活和工作中的方方面面时，也带来了一个棘手的问题——“黑客”问题。由于Internet本身设计的缺陷及其开放性，使其极易受到黑客的攻击。根据美国安全部门统计，Internet上有98%的计算机曾遭受过黑客的攻击性分析，50%的机器被黑客成功入侵过，而被入侵的机器中有20%的管理员并未发现自己曾经被入侵。因此，网络的安全性已经成为阻碍Internet发展的重要因素之一。

事实上，“黑客”并不全是以恶意破坏为目的的“罪犯”，其中有许多高手只是出于好奇。他们掌握着网络世界中许多先进的工具和技术，可以攻击网站、获取情报，从而在人们心中蒙上了一层神秘的面纱。如果你对神秘的黑客和他们的各种手段感兴趣，并热衷于探索其内幕；如果你有志于成为一名伸张正义的黑客；如果你是系统管理员或对安全问题敏感，希望了解敌人可能的进攻手段以提高系统的安全性；那么恭喜你，这本书无疑是你的最佳选择。俗话说，“知己知彼，百战不殆”。如果要更好地保护自己的系统免受攻击、追踪黑客线索，就必须要对黑客的种种攻击手段有详尽的了解；此外，“工欲善其事，必先利其器”，我们必须要熟练掌握如何应用各种安全工具，才能事半功倍。从论述的角度看，本书作者可谓是另辟蹊径，全面深入地介绍了Windows、Linux、FreeBSD、X Window等各种操作系统，以及与其相关的黑客与反黑客工具，并详细介绍了各种攻击方法的原理和应对措施。这就是来自美国的著名黑客给我们带来的丰厚礼物！

作为翻译人员，我们深切地体会到翻译过程也是学习的过程，而且是一个比学习要求更高的过程，深感语句优美的重要；同时，作为一直在计算机安全领域从事研究工作的博士、科研人员，我们感到了责任之重大——即尽快以最高的质量促使本书与读者见面，希望这些巧妙的工具和方法能够对读者的反黑客工作有所帮助。

最后，我们就有关术语问题做一点说明。实际上，当我们在日常工作中与同事、同学交流时，一些术语都是用英文表达的，甚至一些常用的短语短句也是如此。因此在翻译这些术语时，往往理解其含义，但不知如何更加严谨地表达。尽管查阅了各种词典，并重点参考了网上用法，但不可否认，这些词语可能会有更好的译法，我们也渴望得到您的批评指正。

本书是在第一版的基础上，由姜南组织翻译的，赵军锁、宋震、易晓东、彭智、杨征、岳科峰、张煜、邓波、王东霞、张巧利、孙文明、李化、田丽韫、崔桐等人共同完成了本书的翻译、录排、校对等工作，全书最后由姜南负责统稿。本书涉及内容极广，不当之处，请广大读者不吝赐教。

关于作者和合著者

Mike Shema

Mike Shema 是 Foundstone 公司的主要顾问和教员，曾经执行过多种类型（范围甚广）的安全测试，包括网络渗透、防火墙、VPN 检查和 Web 应用程序检查。Shema 熟悉当前最新的安全工具及其漏洞与发展趋势。基于在 Web 应用程序测试中积累的丰富经验，他还为 Buqtraq 发现并提交了几个零天（zero-day）探测。

加盟 Foundstone 公司之前，Shema 在一家产品开发公司工作，其间他为许多 Internet 客户配置并开发了高性能的 Apache Web 服务器和 Oracle 数据库服务器。Schema 以前在 Booz 的 Allen & Hamilton（National Security Team 的一部分）工作，并曾为几个政府和军用站点做过安全评估；此外，他还着手编写了一些安全培训材料。

Shema 在宾夕法尼亚州立大学获取电子工程系和法语系的学士学位。他是 McGraw-Hill/Osborne 出版的“Incident Response: Investigating Computer Crime”一书的技术评论。

Bradley C. Johnson

Bradley C. Johnson 目前在马里兰的盖瑟斯堡仿生 IT 提供商处任网络和安全小组经理，负责其公司内部网络以及客户网络的安全和功能。Johnson 对用于保护和监视计算机网络的工具（如防火墙、入侵检测系统、漏洞和端口扫描工具、网络范围的系统监视程序以及日志档案工具）有着丰富的经验。值得一提的是，Johnson 具有深厚的 C/C++ 和 Perl 编程背景，曾经协助开发了定制的网络安全工具，并予以实现。他毕业于巴尔的摩陶森大学的计算机科学专业。

Keith J. Jones

Keith J. Jones（合著者，也是本书第一版的第一作者）是 Foundstone 公司的计算机取证顾问，主要的研究领域为事件响应程序开发与计算机取证。Jones 专工日志分析、计算机犯罪调查、取证工具分析并专门研究攻击与渗透试验。在 Foundstone，Jones 已经调查了多种不同类型的案例，包括知识产权偷窃、财务贪污、过失以及外部攻击。Jones 是 Foundstone 公司的讲师，同时也是“Incident Response and Computer Forensics”课程的开发者。作为计算机取证方向的专家证人，Jones 已经在美国联邦法庭注册，并参与调查过多宗国际犯罪案件。他曾与人合作编写了“Hacker's Challenge”一书（由 McGraw-Hill/Osborne 出版）。

Jones 拥有两个学士学位（计算机工程与电子工程）和一个硕士学位（电子工程），并且涉猎软件开发（中段规模的项目、开放源代码和专用安全工具）和图像分析（小玻璃瓶和隐写术/水印）领域。

前　言

最近我有一个朋友举办晚宴，当天他为宴会准备的酒的瓶子看上去都非常可爱，但在他倒酒时才发现没有起瓶器。于是他找来刀子和几把剪子费了好大力气才打开瓶子，可又把瓶塞弄到酒瓶里了。不仅如此，他还弄碎了瓶塞，溅出来的酒喷了我一身。后来又有一天，我在女友的公寓里吃晚餐，也是要打开一瓶酒。虽然这次我们有起瓶器，可那是她女伴的，那种起瓶器我用着不大习惯，也可以说我根本就不会用这种工具。我想方设法开启瓶塞，但瓶塞同样是在打开酒瓶前裂开了，起瓶器还扎伤了我的右手。后来我用自己还算完好的手抓起刀子，把另外三分之一的塞子弄到了瓶子里。不过第二天，我又用勺子挖出了碎了的塞子。

发生的这些尴尬的事说明了一个重要的道理。并不是所有人都知道如何打开酒瓶。其次，如果没有合适的工具，那么完成一项任务将会非常地困难。最后，如果不了解如何正确地使用工具，要完成任务同样困难。这些经验教训可以充分应用于计算机和网络安全领域。在进行漏洞扫描的时候，如果没有合适的工具或者不知道如何来正确使用该工具，将很难完成工作。或者你也许能够完成，但是结果并不理想。

本书完整地对当前IT专家所使用的计算机和网络安全工具进行了分类描述。在选择正确的工具前，你必须知道可以使用的工具有哪些，并要对这些工具有个基本的认识。你需要了解这些工具都是如何使用的。

本书旨在提供使用安全工具的实践经验，不仅介绍了如何使用一种工具，而且解释了使用这种工具的原因，以及何时使用哪一种工具。如果对工具的安全原理和概念缺乏理解，而只是单单知道有这样一种工具和几项命令行选项，那么根本就不够。通过屏幕抓图、代码列举、示例工具和“案例学习”的使用，本书向读者展示了在现实工作中是如何用这些工具的。虽然命令行标志和配置选项的讲解可使本书作为读者的案头手册来阅读，但书中各章涵盖的其他信息和基本概念又使得本书不仅仅是一本使用手册。它可以帮助你熟悉各种工具，以便能够更加高效且有效地选择合适的工具并最终出色地完成任务。

本书由四部分内容组成：即多功能工具、在网络上审计系统的工具、审计网络的工具以及用于调查取证的工具。通过本书的学习，读者应当能够掌握一些恰当的、可进行实地试验的工具。

- 审计与预防
- 事件检测
- 调查与响应
- 补救

我们发现，这些任务代表了大多数现实世界中一个安全/网络/系统管理员所要完成的工作。由于书中包含了所有这些任务（从始至终的安全过程），因此出现了“反黑客”一词。

本书各章都遵循一个连续的主题。每章都以一个该章要讨论的工具开始，接下来详细描述每一种工具。但本书不只是工具清单以及介绍工具的参考书；每种工具都包含着深入的使用

技术，依次解释了如何利用工具进行测试。同时，基于作者在实际工作中使用这些工具的一些发现，本书还为读者提供了一些建议。章节中的“案例学习”，用于示范说明现实世界中这些工具的使用，某些章节中的“案例学习”还是对本章讨论的多个工具的组合使用（并包含尽可能多的工具）。针对某些主题，本书为每种工具提供了特定的“案例学习”。在使这些“案例学习”尽可能真实的同时，作者又用了文学用语使故事阅读起来更加有趣。在某些“案例学习”的示范中，可能讲述了系统管理员对其网络中发生的事件所做出的反应，这种讨论或许会引起争议。因此这里要说明的是，在安全交战或事件中，本书决不是提供应当采取何种行动的方法和建议，只不过是希望在章节的末尾奉上有趣的案例分析供您阅读，或者编写一节的内容来强调安全工具的用法。

有些读者曾经阅读过本书的第一版，而本版中增加并更新了一些内容，使得本书介绍的工具能够跟随科学的进步。更新的内容包括：

- 重新组织了章节的结构。
- 更新了一些工具。
- 为 Netcat、tcpdump、Ethereal、nmap、hping 等工具增加了新的案例和示例。
- 增加了 THC-Amap、THC-Hydra、Trinux、Kismet、Ettercap、Wellenreiter、WinHex、X-Ways Trace 等新的工具。
- 增加了一节关于防火墙知识的介绍，讲述了防火墙的概念、ipchains、iptables、ipfw、Cisco PIX 等。

这里要再次强调的是，本书的侧重点是讲述工具的使用，而不是保护网络的方法。因此，本书与 Stuart McClure、Joel Scambray 和 George Kurtz 合著的“Hacking Exposed”，以及 Chris Prosise 和 Kevin Mandia 编写的“Incident Response: Investigating Computer Crime”有很大区别，那两本书构建了这些工具茁壮发展所依赖的方法。所以建议你在学习工具之前，首先了解一下其实现方法。如果你对这些方法已经有了一个全面的了解，那么在阅读本书时将更加容易。

此外，为了使用这些工具，我们必须讨论目前在市面上最流行的操作系统，除非另有说明，否则当提到“Windows”时，指的都是由 Microsoft 发行的操作系统，如 Windows 95/98/Me/NT/2000 和 XP。当提及“UNIX”时，指的是任何版本的 UNIX 操作系统，而不只是来自 Bell Labs 的原版 UNIX。可以在其上使用这些工具的 UNIX 操作系统包括：Solaris（i386 和 Sparc 版本）、Linux、FreeBSD、NetBSD、OpenBSD 等。如果某个工具只能在一个 UNIX 版本上运行，我们就会提到该工具的使用场合。

由于本书所涉及的工具在将来可能会发生变化（尤其是开放源代码或黑客工具），所以我们提供了许多屏幕抓图和输出，这样可以帮助读者将工具的后续版本与本书中提到的信息做个对照。

如前所述，为了追随先进的技术和潮流，计算机及网络安全工具的发展异常迅速。新的工具将不断涌现，旧的工具也会不断增加新的特性。本书的侧重点是网络安全工具，所以本书的 Web 站点提供了这些工具的最新链接、最新的工具信息、本书勘误和内容更新。通过每一条链接，读者可以获取如何安装该工具的信息，还能下载工具的最新版本。网址是 http://www.antihackertoolkit.com。

目　录

第一部分　多功能工具

第二部分　审计工具和主机防护工具

第一部分
多功能工具

第1章 Netcat和Cryptcat

在本书中可以看到很多可选用的网络安全工具和黑客工具。大多数情况下，一种黑客工具往往专用于某一个目的。举例来说，有些黑客工具用于收集网络及其内部主机的信息，另一些则直接搜寻易受攻击的系统。然而，最有用和最常用的工具通常是那些具有多种功能并且适用于不同场合的工具，例如Netcat和Cryptcat。

1.1 Netcat

Netcat能够建立并接受传输控制协议（Transmission Control Protocol，TCP）和用户数据报协议（User Datagram Protocol，UDP）连接。Netcat可在这些连接上读写数据，直到连接关闭为止。它提供了一个基本的TCP/UDP网络子系统，使用户可以手工或者通过脚本与应用层的网络应用程序或服务进行交互。在被下一种诸如文件传输协议（File Transfer Protocol，FTP）、简单邮件传输协议（Simple Mail Transfer Protocol，SMTP）或者超文本传输协议（Hypertext Transfer Protocol，HTTP）的最高层协议封装之前，可以使用该工具来查看原始的TCP及UDP数据。

> **注意**：从技术上讲，Netcat并不能产生UDP连接，因为UDP是一种无连接的协议。就本章而言，每当谈到使用Netcat建立一个UDP连接的时候，都是指在UDP模式中使用Netcat向可能运行在接收端的某个UDP服务发送数据。

Netcat并不会做什么很奇特的事情，它没有漂亮的图形用户界面（GUI），也不按照什么恰当的报告形式输出结果。它很粗糙、原始、丑陋，但是由于它在一个非常基础的层次上工作，所以这个工具在许多情况下都很有用。因为Netcat如果不与其他工具和技术进行结合就得不到任何有用的结果，所以没有经验的用户可能认为Netcat只是一个Telnet客户端工具，而另一些用户则可能很难从冗长的Readme文件所详述的众多命令行参数中看出它会是一个强大的工具。但是，阅读完本章，你将了解到为什么Netcat会成为你的工具包中最有用的工具之一。

1.1.1 使用

由于使用这个工具的用户众多，所以Netcat通常被称为TCP/IP及UDP的“瑞士军刀”。在开始学习如何使用之前，需要下载并安装。

下载

Netcat可以通过多种途径获得，尽管许多UNIX在发布的时候就已经安装了Netcat二进制代码，但最好的方法还是先获取Netcat的源代码，然后进行编译。因为在默认情况下，Netcat源代码可能并未按照用户所需要的选项进行编译。因此，通过下载源程序并自己重新编译，就可以依据需要完全控制Netcat的功能。

UNIX和Windows平台下的Netcat官方下载站点是http://www.atstake.com/research/tools/network_utilities。

安装

本书中并不详细讨论工具的下载、解包及安装，但由于Netcat是我们所介绍的第一个工具，而且它具有一些可能令你感兴趣的编译选项，所以在这里有必要进行一些比较深入细致的讨论。

从@Stake web站点上下载文件nc110.tgz，接着将其解包：

```
[root@originix tmp]# ls
nc110.tgz
[root@originix tmp]# mkdir nc
[root@originix tmp]# cd nc
[root@originix nc]# tar zxf ../nc110.tgz
[root@originix nc]#
```

注意： 与大多数tar包（使用UNIX的tar工具创建的档案文件）不同的是，Netcat并不创建自己的子目录。这一点看起来似乎并不重要，但如果所有的tar包（tarball）和子目录都已下载到同一个目录，你就会发现Netcat把它所有的文件都放到了下载目录的根目录中。这样，清除这些文件将会是一件很烦人的工作。

现在，开始准备编译。下面是两个重要的编译时选项：

- **GAPING_SECURITY_HOLE**　正如其名，当用做恶意目的时，该选项可使Netcat成为一个非常危险的工具，但同时也使得Netcat的功能非常强大。激活这个选项后，Netcat可以运行一个外部程序，而该程序的输入/输出（I/O）将通过Netcat的数据管道流动，使Netcat看起来像一个欺诈性的inetd（端口监视程序）工具，只需要向相应的监听端口建立一个TCP或UDP连接，就可以执行远程命令（如启动一个Shell）。这个选项默认情况下并不启用，因为一旦启用，滥用或错误配置的可能性将很大。但如果正确使用，这个选项将是相当重要的一个特性。
- **TELNET**　通常，如果使用Netcat连接到一个Telnet服务器（使用nc servername 23命令），并不需要做太多的工作。在登录提示符出现之前，Telnet服务器和客户端将会自动协商服务选项。通过激活这个选项，Netcat可以对这些Telnet选项进行响应（通过对每个Telnet选项都回答no来进行），并且使得Telnet出现登录提示符。如果没有这个特性，那么当想要使用Netcat和Telnet做一些有用的工作的时候，就必须使用脚本来响应这些Telnet选项。

至此，这些选项可能对你来说还没有什么显而易见的用处，但在了解了本章末尾的一些例子之后，你就会明白为什么我们要把这些选项提出来进行讨论了。

要激活这些选项，需要在makefile的开头加上DFLAGS行。

```
# makefile for netcat, based off same ol' "generic makefile".
# Usually do "make systype" -- if your systype isn't defined, try "generic"
# or something else that most closely matches, see where it goes wrong, fix
# it, and MAIL THE DIFFS back to Hobbit.

### PREDEFINES

# DEFAULTS, possibly overridden by <systype> recursive call:
# pick gcc if you'd rather, and/or do -g instead of -O if debugging
```

```
# debugging
# DFLAGS = -DTEST -DDEBUG
DFLAGS = -DGAPING_SECURITY_HOLE -DTELNET
CFLAGS = -O
```

可以在DFLAGS行中包含上述两个选项。

如果想运行下面的例子，必须进行这些修改。然而在修改之前，要确保一点：你要么完全拥有正在使用的这个系统，要么可以完全限制其他用户访问将要建立的可执行文件。尽管对其他用户来说，下载一个Netcat的副本并使用这些选项进行编译也是很容易的，但如果某人使用的是你“专门编译”的Netcat作为进入你的系统的后门从而攻击了系统，那么你可能也会恨得咬牙切齿的吧。

准备好编译之后，只需要简单地在提示符下输入make systemtype命令就可以了，其中systemtype参数是类UNIX的各种系统（比如Linux、Freebsd、Solaris等，详见Makefile中对其他操作系统的定义）。完成后，你就会看见一个很小的“nc”二进制文件出现在目录中。

对于Windows用户而言，下载的Netcat压缩文件（nc11nt.zip）中也包含源程序文件，但是，由于大多数用户并没有Windows系统下的编译器，所以默认情况下，使用那两个选项进行编译的二进制文件是固有的。因此只需要简单地对这个文件进行解压，就可以得到nc.exe这个可执行文件了。

命令行

Netcat的基本命令行形式是nc [options] host ports，其中host是要扫描的主机名或IP地址，ports要么是一个单独的端口，要么是一个端口范围（用m-n的形式指定），要么是一系列用空格隔开的单个端口。

现在，差不多已经准备好了，看看使用Netcat都可以做什么令人惊讶的事情吧。但是，首先让我们深入地讨论一下每个命令行选项，从而对它们的用途有一个基本的了解：

- **-d** 只对Windows操作系统有用，该选项使Netcat以隐蔽（stealth）模式工作，从而脱离MS-DOS命令提示符环境运行，使得Netcat不需要保持打开命令窗口便可在监听模式下运行，也可以帮助黑客更好地隐藏监听的Netcat实例面不被系统管理员发现。
- **-e<command>** 如果Netcat使用GAPING_SECURITY_HOLE选项进行编译，那么只要某人在Netcat所监听的任何端口上建立连接，该Netcat都将执行<command>，且客户端Netcat会通过管道将I/O传输到在别处监听的另一个Netcat实例中。使用这个选项非常危险，除非你确实知道自己在做什么。这是在系统中建立后门Shell的一个非常快捷且容易的方法（下面将讲述相关的例子）。
- **-i <seconds>** 延时间隔，表示在两次数据发送之间Netcat等待的时间。例如，当通过管道传输一个文件到Netcat的时候，在传输输入的下一行之前，Netcat将等待<seconds>秒。当你使用Netcat在一台主机的多个端口之间进行操作时，在切换到下一个端口之前，Netcat将等待<seconds>秒，这样可以使用户在进行数据传输或对一个服务进行攻击的时候更加隐蔽，也可以帮助你的端口扫描不被人侵监测系统和系统管理员发现。

- **-g<route-list>**　这个选项具有欺骗性。Netcat 支持松散源路由（loose source routing）（将在本章后面的“IP 欺骗”一节中解释）。你可以在命令行中指定多达 8 个 -g 选项来强迫 Netcat 流量经过特定的 IP 地址，这在你为流量设置伪造的源 IP 地址（使用这种方法可通过防火墙过滤器或者允许访问主机列表）并且希望接收到从主机返回的响应的时候是很有用的。可以让源路由的报文经过你所能控制的机器，从而强制报文返回到你的主机地址而不是到真正的目标地址。注意，这个功能常常达不到预期的效果，因为大多数路由器可能会忽略源路由选项，并且大多数端口过滤器和防火墙将会把这种尝试记入日志。
- **-G<hop-pointer>**　该选项让用户在 -g 选项所指定的路由列表中指定一个地址作为当前的下一跳路由。由于 IP 地址为 4 字节大小，所以这个参数总是 4 字节的倍数；例如，4 代表路由列表中的第一个 IP 地址、8 代表第二个 IP 地址，依次类推。这在你试图伪造部分源路由列表使得报文看起来像是从别处来的时候非常有用。通过把假的IP地址放在前两个 -g 列表位置并指定一个跳步指针 12，报文将被直接路由到你的路由列表中的第三个 IP 地址。然而，实际报文中仍然包含这些假的 IP 地址，这样报文就像是来自于某个地方，而实际上这个报文是来自其他地方的。在进行欺骗和源路由时，这个功能有助于屏蔽你的位置，但你也许不一定能够接收到响应报文，因为响应报文将经由伪造的IP地址反向路由回来。
- **-l**　该选项切换 Netcat 的“监听”模式。该选项必须与 -p 选项一起使用以告诉 Netcat 绑定某个指定的 TCP 端口并等待到来的连接。如果增加 -u 选项，则应使用 UDP 端口而非 TCP 端口。
- **-L**　该选项只在 Windows 版中有用，是一个比 -l 选项的功能更强大的“监听”选项。它告诉 Netcat，当一个连接被关闭后，使用相同的命令行选项重启监听模式。这时，即使此时初始的连接已经结束，Netcat也能在不需要用户干涉的情况下接收后面新的连接。与 -l 一样，该选项也必须同 -p 选项一起使用。
- **-n**　该选项告诉 Netcat 不要做任何主机名查找工作。如果在命令行中使用该选项，一定要确保不能指定任何主机名作为参数。
- **-o<hexfile>**　对数据执行一次十六进制转储（hex dump）并将其存储在 hexfile 中。命令 nc -o hexfile 将把双向通信的数据记录下来，在每行的开始处会有一个“<”或“>”用来分别指示数据是进入数据还是外出数据。如果你只想得到对进入数据的十六进制转储，可以使用 nc -o<hexfile 命令；如果只想得到对外出数据的十六进制转储，则可以使用 nc -o >hexfile 命令。
- **-p<port>**　让用户指定一个 Netcat 应使用的本地端口号。当通过 -l 或 -L 选项来使用监听模式的时候，这个参数是必需的。如果没有为外出连接指定这个参数，则与大多数其他 TCP 或 UDP 客户程序的做法一样，Netcat 将使用系统分配给它的任何端口。记住，在 UNIX 系列系统中，只有根用户才能指定小于 1024 的端口号。
- **-r**　该参数使 Netcat 可以随机选择本地和远程端口。当使用 Netcat 在系统中范围很大的一批端口上获取信息时，要想混合源端口和目标端口的顺序使其看起来不怎么像端口扫描，这个选项是很有用的。当这个选项与 -i 选项和一个足够大的间隔结合使用的

时候，在不被注意的情况下进行端口扫描成功的可能性会增加，除非系统管理员仔细地分析了日志。

- -s 指定Netcat建立连接时所使用的IP地址，该选项允许黑客做一些相当卑鄙的勾当。首先，它允许黑客隐藏他们的IP地址或者假冒其他人的IP地址，但是要得到路由到他们所欺骗的地址的任何信息，都需要使用-g源路由选项。其次，当处于监听模式时，大多数情况下可以与一个已经监听的服务“预先绑定”。所有TCP和UDP服务都绑定到某一端口，但它们并不全都绑定到一个特定的IP地址。许多服务默认情况下监听所有可用端口。例如，Syslog将监听UDP端口514上的系统日志数据。然而，如果运行Netcat在514端口上监听，并且使用-s来指定一个源IP地址，那么所有到那个特定IP地址的流量都会先到达正在进行监听的Netcat。为什么呢？如果某个套接字既指定了端口又指定了IP地址，那么它将比那些没有绑定到特定IP地址的套接字具有更高的优先级。我们将在后面详细讨论这部分内容（参见“抢夺服务”一节），同时会告诉你如何鉴别系统中的哪些服务可以预先绑定。
- -t 如果使用TELNET选项编译，Netcat就可以与Telnet服务器进行Telnet选项协商，虽然它的响应是毫无意义的信息，但毕竟可以让你进入到提示符状态，该状态可能是当使用Netcat连接到TCP的端口23的时候你所希望见到的。
- -u 告诉Netcat使用UDP而非TCP。在客户端模式和监听模式下都可以起作用。
- -v 控制Netcat告诉你有关它将要做的事情所达到的程度。如果使用no -v，那么Netcat将仅吐出它所接收到的数据。一个单独的-v可以让你知道它连接或绑定的地址以及是否有问题发生。第二个-v可以让Netcat在一个连接结束时使你知道这个连接总共发送和接收了多少数据。
- -w<seconds> 控制在一个连接上Netcat放弃之前等待的时间。同时也告诉Netcat当在标准输入上接收到一个EoF（end-of-file）之后应该等待多长时间来关闭连接并退出。如果你通过Netcat向远程服务器发送命令并且期望大量的数据返回（例如，向一台Web服务器发送一个HTTP命令以下载一个大文件），则该选项是很有用的。
- -z 如果只准备查找哪个端口处于打开状态，可使用nmap（参见第4章）。但是该选项告诉Netcat只发送必要的数据来检测在指定范围内的哪些端口上存在程序监听。

在大致了解了Netcat的功能后，下面来看看实际应用中如何使用这一工具。

1.1.2 Netcat的用法

人们宣称已经找到几百种在日常任务中使用Netcat的方法，它们中的一些很相似，只有一点微小的差别。我们试图举出一些像Netcat本身一样通用并且覆盖范围最广的例子。下面是我们认为最重要的几种用法。

获得对Shell的远程访问

难道你不希望在世界上任何一个地方都能进入熟悉的DOS提示符状态吗？可以在一台Windows NT或Windows 200x系列系统的DOS提示符状态下运行nc.exe -l -p 4455 -e cmd.exe命令，这样，任何远程登录到端口4455的Telnet甚至不需要登录就可以看到一个DOS Shell。