Google核心技术丛书

Google Hacking 技术手册

Google Hacking for Penetration Testers

（美） Johnny Long 等著
李 静 等译

机械工业出版社
China Machine Press

Google Hacking技术手册

Google Hacking for Penetration Testers

由全球最大的Google Hacking数据库的维护者“The Google Guy”撰写
抢在那些坏人行动之前，通过Google查找到你自己的敏感数据
特殊的Google Hacking案例分析，列举了最让人惊慌的Google黑客举措

Google Hacking技巧的全新版

自尊的Google黑客会花上好几个小时在互联网上查找丰富的资源。通过一轮又一轮的搜索，他们会因找到清晰易读的、有意义的、精简的搜索结果而欣喜若狂。作为Google Hacking Database（GHDB）和http://johnny.ihackstuff.com网页上的搜索引擎攻击论坛的创建者，我经常会为Google hacking社团提出的内容而惊愕不已。这表明传闻是真的——具有创造力的Google搜索可以泄露医药、金融、财产甚至是机密信息。尽管政府颁布了法令、规章及保护法案，且不断设置安全装置，这一问题仍然存在。信息仍会驱使网络中这一问题的发生，而这正好被Google黑客逮个正着。请阅读本书了解这些新的内容，以保护你免受Google黑客的攻击。

——Johnny Long

- 学习Google搜索的基础知识。
- 使用高级的操作符执行高级查询。
- 学习Google黑客的多种手法。
- 回顾文档研磨以及数据库挖掘的知识。
- 理解Google在信息收集框架中扮演的角色。
- 搜索漏洞利用程序并查找目标。
- 学习10种简单的安全性搜索。
- 搜索Web服务器。
- 了解坏家伙们是如何搜索数据的。
- 了解黑客如何攻击Google服务。

投稿热线：(010) 88379604
购书热线：(010) 68995259, 68995264
读者信箱：hzjsj@hzbook.com
华章网站 http://www.hzbook.com
网上购书：www.china-pub.com
封面设计：王建敏

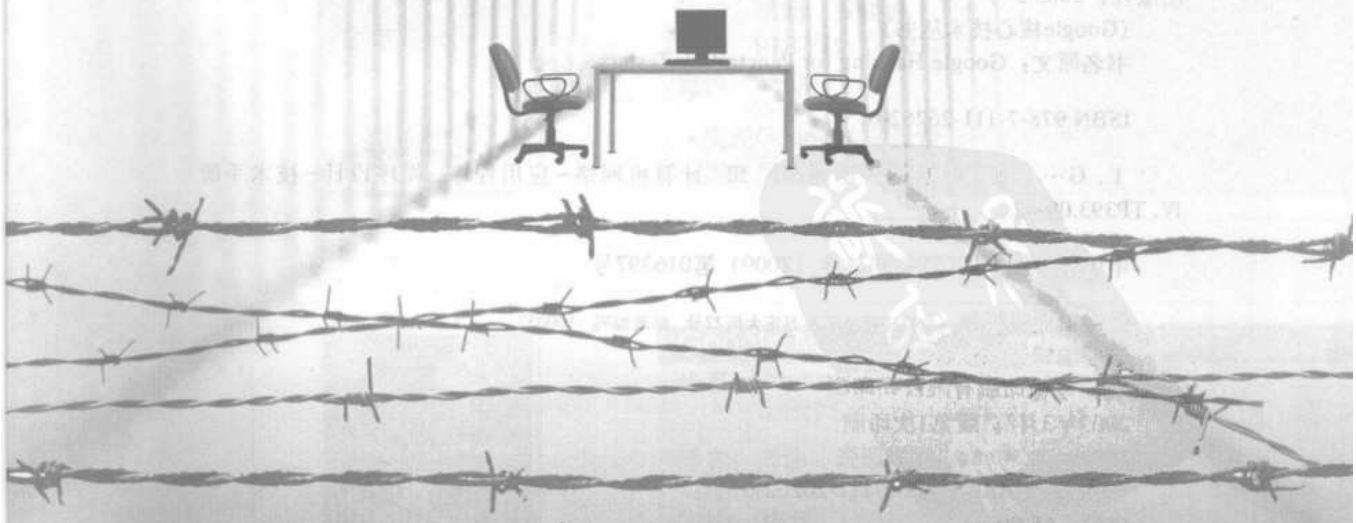
上架指导：计算机/网络安全
ISBN 978-7-111-26262-6
9 787111 262626
定价：65.00 元

Google核心技术丛书

Google Hacking 技术手册

Google Hacking for Penetration Testers

（美） Johnny Long 等著
李 静 等译



机械工业出版社
China Machine Press

译者序

人类的每一个进步都悄然而巨大地改变了人们的生活，计算机的出现使得信息化办公成为可能，网络的出现使得地球成为了“村庄”，搜索引擎更是让网络无秘密可言。查找网页，我们用搜索；查找信息，我们用搜索；查找朋友，我们用搜索。离不开计算机、离不开网络的人们都无一例外地爱上了搜索。一时间，网络拉近了全世界各地的人们之间的距离，搜索改变了人们的生活，让人不禁感慨：“夜夜思君不见君，共用Google搜索。”

然而任何事物都有两面性，你可以使用它来美化生活，也可以用它来破坏世界。Google也不例外，它的强大功能可以为黑客中的“黑帽”所用，也可以为“白帽”工作。在黑客日益增长且黑客技术日益强劲的网络时代，你是否已经察觉到了身边的危机？在网络如此盛行的时代，你不可能脱离网络生活，也正是因为离不开网络，才使得你时刻身陷网络危机之中。你的重要个人信息不但用于电子邮箱注册、论坛注册，还经常在聊天时被你不经意地传送给朋友，你开通了网上购物功能，开通了银行卡的网上购物功能，甚至把家人的敏感信息也登在了自己的博客之中，你可知道这些信息对于黑客来说是多么大的馈赠吗？难道你一点也不担心黑客冒用你的名义做坏事，用不正当的方式刷你的银行卡吗？

现在是潜心了解Google Hacking的时候了！在这么恶劣的网络环境中，维护个人正当权益不能仅仅依靠法律、道德、自觉，更重要的是自己要有防患意识，知道黑客们如何“黑”到你的信息。

本书是一本告诉你黑客是如何通过Google这一强大的引擎在线精确搜索敏感信息并据为己有。阅读本书后，不仅可以大大强化你的搜索知识和能力，还可以将了解到的黑客知识用于防范。书中讲解了很多搜索的技巧和案例，相信这些技巧和案例会使黑客技术爱好者、专业的安全测试或渗透测试人员，以及网络管理人员获益匪浅。总而言之，Hacking的最高境界就是强化搜索别人的功能并避免泄露自己的敏感信息。想要一试究竟的读者不妨仔细地学习并实践一下，跟随本书的步骤，你一定会搜索到让你惊讶的东西。不信？那你就试试看吧！

参与本书翻译的全体人员有：李静、贺倩、李凌燕、贺强、梁晓琴、陈平锋、吴启文、卢祖英、幸慧、陈爱萍、马睿倩、翁子扬、苏建忠和穆陟晅。译者水平有限，书在难免存在瑕疵，敬请广大读者朋友不吝赐教。

译　者

2008年12月

前言

我是Johnny。我爱好黑客技术。

你是否曾有一种会改变你的生活的业余爱好？Google Hacking就成了我的一种爱好，不过在2004年，它成了一份意想不到的礼物。那一年，我参加了Defcon会议并发表了演讲，这是我事业的顶峰。那一年我处于世界之巅，并且有些飘飘然——我实际上是一个微不足道的小人物。我发表了有关Google Hacking的演讲，并在演讲的时候模仿了我的偶像。演讲进行得很顺利，确实引起了诸多评论，并预示着我辉煌演说生涯的开始。前景一片光明，但是到了周末，我却感到空虚。

在两天的时间里，一连串不幸随之而来，把我从成功的顶峰无情地拽到了绝望低谷的峭壁上。过分？有点，不过这是我的真实感受，而且我甚至没有从中得到任何收获。我不确定是什么原因导致我这样，但是我向上张开双臂，把我工作中所有的烦恼，我的事业，我的500位网站用户以及我刚刚开始的演讲生涯，都交给了上帝。

那时，我并不是很清楚那意味着什么，不过我是认真的，我渴望能实现巨大的改变，并且莫名地希望为了更高的目标而生活。我有生以来第一次看到了我生命的浅薄和自私，而这让我感到了震惊。我希望得到更多，而且是希望真的得到。可笑的是，我得到的要比我要求的多。

Syngress出版公司找到了我并问我是否愿意写一本Google Hacking方面的书，也就是你现在看到的这本书的第1版。我极其希望我能掩饰自己经验的缺乏以及对写作的厌恶，因此我接受了这份我后来视为“原创的礼物（original gift）”的邀请。如今，Google Hacking已经成为了畅销书之一。

我的网站用户从500名上升至近8万名。Google图书出版计划衍生了10个左右的额外图书出版计划。媒体如潮汐般涌来，给我留下了极深的印象，首先是Slashdot，紧随其后的是在线媒体、出版界、电视台和有线电台。随着邀请我参加的会议逐渐增多，我得以很快进行了全球旅行。我特别希望成为Hacking社区中的一员，很庆幸他们无条件地接收了我，尽管我最近持有很多保守观点。他们从我的网站购买图书，并将由此产生的收入用于公益事业，他们甚至全额资助我和我妻子的非洲乌干达之旅。这一系列的事件改变了我的生活，并且为ihackcharities.com做好了铺垫，这是一个旨在Hacking社区技术与需要这些技术的公益事业间搭建连接桥梁的组织。我“真正的”生活也得到了改变，我与妻子和孩子的关系比任何时候都要好。

如你所见，对我来说它不仅是一本书。它实际上是一份原创的礼物，我非常认真地进行了本书的升级改版工作。我亲自审核了每一句话和每一幅图片（特别是我编写的那一部分）以保证它的正确性。我以本书的第2版为荣，我很感激各位读者，感谢你支持那些为本书倾注了很

多心血的人们。谢谢你！

欢迎你访问我们的网站http://johnny.ihackstuff.com，感谢你阅读本书。欢迎你链接Google Hacking Database，通过单击我们的Amazon链接来资助公益事业。感谢你为我们提供一个影响现实变化的平台——不只是在安全社区，而是在更大的全球范围。我真诚感谢你的支持。

—Johnny Long

2007年10月

撰稿作者介绍

Roelof Temmingh出生于南非，就读于比勒陀利亚大学，并于1995年获得电子工程技术学位。从那时起，他就一直对计算机安全充满了激情。他曾经是一名开发人员，随后在1995年到2000年之间，在一个信息安全工程公司做一名系统架构师。2000年初，他与一些安全评估和咨询领域中的引领者一同创建了安全评估和咨询公司SensePost。在SensePost工作的期间，他出任评估团队技术主管，之后，他又负责公司的技术革新中心的管理。Roelof曾经在多个国际会议上发表演说，例如Blackhat、Defcon、Cansecwest、RSA、Ruxcon和FIRST会议。本书中由他撰写的内容包括：窃取网络、如何拥有一块陆地，渗透测试者的开放源代码工具箱等，同时他还是“通过数字攻击”训练课程的首席培训师之一。Roelof编写了几个非常有名的安全测试应用程序，如Wikto、Crowbar、BiDiBLAH和Suru。2007年初，他创建了Paterva以便能自主从事研发工作。Roelof在Paterva期间开发了一个称为Evolution（现在称为Maltego）的应用程序，该程序在信息收集和信息关联领域展现了巨大的潜力。

Petko “pdp” D. Petkov是英国伦敦的一名高级IT安全咨询师。他每天的工作就是识别漏洞，构建攻击策略，创建攻击工具和渗透测试基础结构。私底下，人们都知道Petko是pdp或者架构设计师，但是在IT安全行业，他的名字却是因为他强大的技术背景和富有创造性的思维而广为人知。

他最近的一个项目，GNUCITIZEN（gnucitizen.org）是一个领先的在线Web应用程序安全资源，为公众利益而公开了其部分工作。Petko将他自己定位为安全圈内的“酷”猎人。

他与他心爱的女友Ivana一起居住，如果没有他的女友，他将不可能参与本书的撰写。

CP是GHDB和位于http://johnny.ihackstuff.com的论坛的仲裁员，他是Advanced Dork:等许多开源的工具的研发者，是Google站点索引编著者，是http://tankedgenius.com的合作创立者，是自由安全咨询顾问，是DC949（http://dc949.org）的积极分子——在DC中，他参与了一年一度的名为“Amateur/Open Capture the Flag”的hacking竞赛以及各种搜索项目。

“我的身份众多，但是最重要的身份是黑客。”——CP

Jeff Stewart（Jeffball55）正在East Stroudsburg大学主修计算机科学、计算机安全和应用数学。他是johnny.ihackstuff.com论坛的活跃分子，经常在这个论坛中编写与Google服务相关的程序以及Firefox扩展程序。他当前从事的所有项目都可以在http://www.tankedgenius.com上找到。最近，他接了一个关于FD软件计划（FD Sofware Enterprise）的工作：帮助几家医院创建

事故管理系统（Incident Management System）。

Ryan Langley是加利福尼亚人，目前居住在洛杉矶。身为兼职程序员以及安全评估员的Ryan一直都在坚持不懈地研究和学习有关IT安全的知识以及新的评估技术。Ryan拥有五年的系统维修和管理员经验。他经常与CP和Jeffball合作项目。

致谢

此时此刻，我要感谢很多人，不过我不会全部列出他们的名字，但是我会尽最大的努力。

感谢我的妻子和三个可爱的孩子。无法用言语来表达你们对于我的意义。感谢你们包容“真实的”j0hnny。

感谢这本书的团队：CP、Seth Fogie、Jeffball55、L0om、pdp、Roelof Temmingh、Rar、Zanthas。感谢我的朋友Nathan、Mike “Corn” Chaney、Seth Fogie、Arun、@tlas和Apu。感谢Shmoo组中我众多的知己和支持者，ihackcharities志愿者以及支持者，Malcolm Mead和Pat、Predestined（David、Em、Isaac、Josh、Steve、Vanessa）、Tushabe家庭、Dennis以及所有AOET家庭成员。

我还要借此机会感谢Google Hacking社区的成员。正是他们的付出，才让此书以及Google Hacking的运转如期进行。以下列出了这些成员的名字，根据他们对GHDB的发贴数量排序。

Jimmy Neutron（107）、rgod（104）、murfie（74）、golfo（54）、Klouw（52）、CP（48）、L0om（32）、stonersavant（32）、cybercide（27）、jeffball55（23）、Fr0zen（22）、wolveso（22）、yeseins（22）、Rar（21）、ThePsyko（20）、MacUk（18）、crash_monkey（17）、MILKMAN（17）、zoro25（15）、digital.revolution（15）、Cesar（15）、sfd（14）、hermes（13）、mlynch（13）、Renegade334（12）、urban（12）、deadlink（11）、Butt-Pipe（11）、FiZiX（10）、webby_guy（10）、jeffball55+CP（8）、James（7）、Z!nCh（7）、xlockex（6）、ShadowSpoof（6）、noAcces（5）、vipsta（5）、injection33（5）、Fr0zen+MacUK（5）、john（5）、Peefy（4）、sac（4）、sylex（4）、dtire（4）、Deakster（4）、jorokin（4）、Fr0zen rgod（4）、zurik6am（4）、brasileiro（4）、miss.Handle（4）、golfo42（3）、romosapien（3）、klouw（3）、MERLiiN（3）、Darksun（3）、Deeper（3）、jeffball55+klouw（3）、ComSec（3）、Wasabi（3）、THX（3）、putsCTO（3）。

以下所列主体为GHDB提供了两个附件：HaVoC88、ToFu、Digital_Spirit、CP and golfo、ceasar2、namenone、youmolo、MacUK / CP / Klouw、242、golfo、CP and jeff、golfo and CP、Solereaper cp、nuc、bigwreck_3705、ericf、ximum、/iachilles、MacUK/CP、golfo and jeffball55、hevnsnt、PiG_DoG、GIGO、Tox1cFaith、strace、dave@cirt.net、murk、klouw & sylex、NRoberts、X-Ravin、ZyMoTiCo、dc0、Fr0zen jeffball55、Rar CP、rgod jeffball55、vs1400、pitt2k、John Farr、Kartik、QuadsteR、server1、rar klouw、Steve Campbell。

以下主体为GHDB提供了一个附件：Richie Wolk、baxter_jb、D3ADLiN3、accesspwd1、darkwalk、bungerScorpio、Liqdfire、pmedinua、WarriorClown、murfie & webbyguy、stonersavant、klouw、thereallinuxinit、arrested、Milkman & Vipsta、Jamuse and Wolveso、

FiZiX and c0wz、spreafd、blaqueworm、HackerBlaster、FiZiX and klouw、Capboy118、Mac & CP、philY、CP and MacUK、rye、jeffball55 MacUK CP9、rgod + CP、maveric、rar、CP、rgod + jeffball55、norocosul_alex R00t、Solereaper、Daniel Bates、Kevin LAcroix、ThrowedOff、Apoc、mastakillah、juventini、plaztic、Abder、hevensnt、yeseins & klouw、bsdman & klouw & mil、digital.ronin、harry-aac、none90810、donjoe145、toxic-snipe、shadowsliv、golfo and klouw、MacUK / Klouw、Carnage、pulverized、Demogorgo、guardian、golfo、macuk、klouw、Cylos、nihil2006、anonymous、murfie and rgod、D. Garcia、offset、average joe、sebastian、mikem、Andrew A. Vladimirov、bullmoose、effexca、kammo、burhansk、cybercide cybercide、Meohaw、ponds、blackasinc、mr.smoot、digital_revolution、freeeak、zawa、rolf、cykyc、golfo wolveso、sfd wolveso、shellcoder、Jether、jochem、MacUK / df、tikbalang、mysteryman0122、irn-bru、blue_matrix、dopefish、muts、filbert、adsl3000、FiNaLBeTa、draino、bARDO、Z!nCh & vs1400、abinidi、klouw & murfie、wwooww、stonersavant、jimmyn、linuxinit、url、dragg、pedro#、jon335、sfd cseven、russ、kg1、greenflame、vyom、EviL_Phreak、golfo、CP、klouw、rar murfie、Golem、rgod +murfie、Madness!、de Mephisteau、gEnTi、murfie & wolveso、DxM、l0om wolveso、olviTar、digitus、stamhaney、serenh、NaAcces、Kai、goodvirus、barabas、fasullo、ghooli、digitalanimal、Ophidian、MacUK / CP / Jeffb、NightHacker、BinaryGenius、Mindframe、TechStep、rgod +jeffball55 +cp、Fusion、Phil Carmody、johnny、laughing_clown、joenorris、peefy & joenorris、bugged、xxC0BRAxx、Klouw & Renegade334、Front242、Klouw & digital.revo、yomero、Siress、wolves、DonnyC、toadflax、mojo.jojo、cseven、mamba n*p、mynewuser、Ringo，Mac / CP，MacUK / golfo，trinkett，jazzy786，paulfaz，Ronald MacDonald，.-DioXin-.，jerry c、robertserr、norbert.schuler、zoro25 / golfo、cyber_、PhatKahr4u2c、hyp3r、offtopic、jJimmyNeutron、Counterhack、ziggy1621、Demonic_Angel、XTCA2S、m00d、marcomedia、codehunter007、AnArmyOfNone、MegaHz、Maerim、xyberpix、D-jump Fizix、D-jump、Flight Lieutenant Co、windsor_rob、Mac、TPSMC、Navaho Gunleg、EviL Phreak、sfusion、paulfaz、Jeffball55、rgod + cp clean +、stokaz、Revan-th、Don、xewan、Blackdata、wifimuthafucka、chadom、ujen、bunker、Klouw & Jimmy Neutro、JimmyNeutron & murfi、amafui、battletux、lester、rippa、hexsus、jounin、Stealth05、WarChylde、demonio、plazmo、golfo42 & deeper、jeffball55 with cle、MacUK / CP / Klou、Staplerkid、firefalconx、ffenix、hypetech、ARollingStone、kicktd、Solereaper Rar、rgod + webby_guy、googler。

最后，我想重申一下我对第1版中所提到的所有人的谢意，这些人仍与我有关：

感谢我的母亲和父亲，感谢你们允许我对数字生活的废寝忘食。感谢这本书的团队：Alrik“Murf”van Eijkelenborg、James Foster、Steve、Matt、Pete和Roelof。Cooper先生、Elliott夫人、Athy C、Vince Ritts、Jim Chapple、Topher H、Mike Schiffman、Dominique Brezinski和rain.forest.puppy都停下手头的工作来帮助我成就未来。没有我的那些亲密的朋友们的帮助，就没有这本书，他们是Nathan B、Sujay S、Stephen S。感谢Mark Norman让一直保持生活在现实中。来自Google Hacking论坛的Google大师们对论坛和GHDB做了许多的贡献，我很荣幸能在

这里把他们列出来（以发帖数降序排列）：murfie、jimmyneutron、klouw、l0om、ThePsyko、MILKMAN、cybercide、stonersavant、Deadlink、crash_monkey、zoro25、Renegade334、wasabi、urban、mlynch、digital.revolution、Peefy、brasileiro、john、Z!nCh、ComSec、yeseins、sfd、sylex、wolveso、xlockex、injection33、Murk。特别感谢Murf在我写作此书时维护网站的运行，同时也要感谢版主团队：ThePsyko、l0om、wasabi和jimmyneutron。

StrikeForce总是难以描述，但是它占据了我生活的大部分，所以即使我只能玩一小部分也是非常感谢：Jason A、Brian A、Jim C、Roger C、Carter、Carey、Czup、Ross D、Fritz、Jeff G、Kevin H、Micha H、Troy H、Patrick J、Kristy、Dave Klug、Logan L、Laura、Don M、Chris Mclelland、Murray、Deb N、Paige、Roberta、Ron S、Matty T、Chuck T、Katie W、Tim W、Mike W。

感谢CSC和许多我曾经的帅气的老板。你们很牛："FunkSoul"、Chris S、Matt B、Jason E和Al E。感谢'TIP成员让生活充满乐趣。'TIP成员有许多人，但是我只记得一些与我合作比较多的：Anthony、Brian、Chris、Christy、Don、Heidi、Joe、Kevan、The 'Mikes'、"O"、Preston、Richard、Rob、Ron H、Ron D、Steve、Torpedo、Thane。

在写作本书的过程中，我听了许多音乐来掩盖那些噪声。感谢P.O.D（感谢Sonny）、Pillar、Project 86、Avalon O2 remix、D.J. Lex、Yoshinori Sunahara、Hashim and SubSeven（很棒的名字！）（第2版的更新：Green Sector、Pat C.、Andy Hunter、Matisyahu、Bono和U2）。Shouts to securitytribe、Joe Grand、Russ Rogers、Roelof Temmingh、Seth Fogie、Chris Hurley、Bruce Potter、Jeff、Ping、Eli、Blackhat的Grifter，以及作者的整个Syngress家族。我非常荣幸能成为这一团队的一员，尽管在你们面前我很自惭形秽！感谢Andrew和Jaime。你们很牛！

感谢苹果计算机公司制造出如此帅气的笔记本（和操作系统）。

—Johnny Long

当人们的信息来源渠道日益扩大时，搜索、过滤信息已成为网民必不可少的日常操作，因此Google自然而然地成为了人们常用的工具之一。本书采纳了详尽的Google语法与工具。在发挥Google的最大搜索效用后，你又将如何反其道，让自己的信息无法被别人搜索到呢？只要你细细品读书中的讲解，顺着作者的逆向思维提示，便可轻松地找到甩掉中低级黑客的方法。

本书内容详实，通俗易懂，可作为技术人员的参考用书。

Google Hacking for Penetration Testers
Johnny Long
ISBN:978-1-59749-176-1

ISBN:978-981-272-105-1

本书版权登记号：图字：01-2009-1622

图书在版编目（CIP）数据

Google Hacking技术手册/（美）朗格（Long, J.）等著；李静等译. —北京：机械工业出版社，2009.3
（Google核心技术丛书）
书名原文：Google Hacking for Penetration Testers

ISBN 978-7-111-26262-6

Ⅰ. G…　Ⅱ. ①朗…　②李…　Ⅲ. 计算机网络-应用程序-程序设计-技术手册
Ⅳ. TP393.09-62

中国版本图书馆CIP数据核字（2009）第016397号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：陈佳媛
北京瑞德印刷有限公司印刷
2009年3月第1版第1次印刷
186mm × 240mm · 23印张
标准书号：ISBN 978-7-111-26262-6
定价：65.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

目　录

第1章 Google搜索基础知识

1.1 简介

Google的Web界面很清爽。它的“观感效果（look and feel）”[⊖]是受版权保护的。其界面清新而简约。然而大多数人都没有意识到这个界面的功能也是十分强大的。在本书中，我们将了解到怎样利用Google来展示那些真正让人感到惊奇的事。但是，就像日常生活一样，在开始跑之前，要先学会走。

本章简要介绍Google搜索的基础知识。Google功能强大的基于Web的界面，让它成为一个家喻户晓的词，我们就从探索它开始。甚至许多高级Google用户仍在使用基于Web的界面来完成他们的日常查询工作。在掌握了怎样浏览和解释各种界面生成的结果之后，我们将开始探索基本的搜索技巧。

掌握基本的搜索技巧是学习高级查询技巧的基础。你将学到怎样合理地使用Boolean（布尔）操作符（AND、NOT以及OR），同时探索强大和灵活的群搜索。我们也将学到Google几种独特的通配符实现。

最后，我们将了解Google的URL（Uniform Resource Locator，统一资源定位符）结构的语法。学习Google URL的输入和输出能够让你以更快的速度和更大的灵活性来提交一系列相关的Google搜索。Google URL结构为朋友和同事之间交流有趣的搜索提供了一种极棒的简约形式。

1.2 探索Google的Web界面

1.2.1 Google的搜索页面

Google的主页面可以在www.google.com上看到，如图1-1所示。这个界面以其简洁的线条、令人愉悦的整洁感觉和友好的接口而广为人知。尽管这个界面初看起来似乎功能有些少，但是我们将看到许多不同的搜索功能正是从此页开始执行的。

如图1-1所示，用户只可以在一个地方输入内容。这是搜索域（search field）。如果想问Google一个问题或者进行一次查询，你只需简单地把要查找的东西输入进去，然后敲一下Enter（回车键）（前提是你的浏览器支持这种操作），或者点击“Google搜索”（Google Search）按钮，Google就可以给出你想要的查询结果了。

⊖ look and feel（外观和感觉）通常是指程序的图形用户界面的形式与功能，它包含的元素有颜色、形状、布局以及字体等，同时也包含按钮、输入框以及菜单等动态元素的行为。这个术语可用于软件以及网站。由于人们逐渐认识到了look and feel的重要性，所以一些国家制定了相关的版权保护法律。——译者注

图1-1 Google的主页面

在界面顶部的那些链接包括：Web（网页），Images（图片），Video（视频）等。用于打开表1-1所示的其他搜索区域。每一部分的基本搜索功能均相同。我们将在第2章中看到，Google网页界面的每个搜索区域都有不同的功能，并能接受各种不同的查询操作符。例如，author操作符专门设计用于Google Groups搜索领域。表1-1描述了Google主页面的各种不同搜索区域的功能。

表1-1 Google主页面的链接及其功能

界面部分	描 述
Google工具栏	我正在使用的浏览器安装了一个Google“工具栏”，并把它放在了地址栏的后面。下一节将介绍各种不同的Google工具栏
网页，图片，视频，新闻，地图，Gmail和更多标签	这些标签支持用户分别搜索网页，照片，群组发布的消息，Google地图，Google邮件。如果用户初次使用Google，则要知道这些标签并不总能代替“提交搜索”（Submit Search）按钮。这些标签只是简单地切换到其他Google搜索应用
iGoogle	该链接将返回到用户的个性化Google主页
（Signin）登录	该链接支持用户通过登录到个人的Google账户（Google Account）来注册访问额外的功能
搜索项输入域	该文本域位于那些备用的搜索标签的正下方，允许用户输入一个Google搜索项。我们将在整本书中讨论Google搜索的语法
Google Search（Google搜索）按钮	这个按钮是用来提交用户搜索项的。对于大多数浏览器而言，在输入搜索项之后简单地敲击Enter/Return键（回车键）就可以激活这个按钮
I'm Feeling Lucky（手气不错）按钮	与列出搜索结果列表不同的是，这个按钮会挑选出针对输入的搜索项而言最佳的页面。通常这个页面是和输入的搜索项最相关的页面

（续）

界面部分	描　述
Advanced Search（高级搜索）	这个链接可转到高级搜索页面。第2章将介绍这些高级选项
Preferences（使用偏好）	这个链接允许用户设置某些选项（这些设置会保存在用户机器里的cookies中，以便下次访问时使用）。可选的选项包括语言选择，父辈过滤器，每页列出的结果数量以及结果视窗选项
Language tools（语言工具）	这个链接允许你设置许多不同的语言选项以及各种语言之间的翻译

1.2.2 Google的查询结果页面

在处理完一次搜索查询后，Google显示出一个结果页面。如图1-2所示，这个结果页面列出查询结果并且提供包含查询内容的网页。

图1-2 典型的网页查询结果页面

查询结果页面的顶部与主搜索页面相同。注意该页面顶部的Image（图片）、Video（视频）、News（新闻）、Maps（地图）和Gmail（邮件）链接。如果点击搜索页面的这些链接，那么就可以自动地将搜索作为另一种搜索类型重新提交，而不用重新输入一遍。

结果行显示出当前页列出了哪些结果（图中是1—10），约有多少项符合（这里是800万以上），搜索查询自身（包括可以在字典中查找每个单词的链接）以及执行搜索所用的时间。人们通常不会关注查询速度，不过它确实非常迅速。即使有上百万项结果的大量查询也能够在不到一秒的时间内返回！

对于结果页面上的每一项，Google都列出了网站的名称，网站的介绍（通常是该网站内容的前几行），符合查询的页面URL，页面的大小和上次抓取[⊖]该页面的日期，一个显示Google上次抓取的时候页面内容的经过缓存的链接，以及一个到类似网页的链接。如果查询结果页面未采用你的本地语言，那么Google还支持把该页面翻译成该语言（可以在使用偏好中设置），这时将会出现一个标题为“翻译此页”（Translate this page）的链接，它允许用户能够使用自己的语言来近似阅读该页面（如图1-3所示）。

图1-3 Google翻译

Google搜索背景知识

翻译代理

通过翻译服务，可以把Google当作一个透明的代理服务器来使用。当你点击“翻译此页”（translate this page）链接时，会转到一个位于Google的服务器上的该页面的翻译副本。它可以替你获取相关页面，这可以看作是一种代理服务器。即使你想要查看的页面并不需要翻译，通过修改URL中的hl变量值来匹配该页的本地语言这种办法，你仍然可以把翻译服务作为一种代理服务器来使用。注意，图片并不能以这种方式作为代理。

1.2.3 Google Groups

由于基于Web的论坛、博客、邮件列表和实时消息技术的风靡，作为最原始的公众论坛讨论形式的USENET新闻组已经被人们忽视。但是仍然有许多用户每天都在USENET上发布消息。关于USENET所包含内容的完整介绍可以在www.faqs.org/faqs/usenet/what-is/part1/中找到。DejaNews（www.deja.com）过去曾被认为是收集过去和现在的新闻组消息的权威站点，而它在2001年2月被Google收购（参见www.google.com/press/pressrel/pressrelease48.html）。这个收购使得用户能够通过简单明了的Google搜索界面来查询自1995年以来发布的全部USENET消息存档。Google把USENET新闻组称作Google Groups。今天，全球范围内的互联网用户都转向Google Groups进行讨论和解决某些问题。IT从业人员从Google Groups获得各种技术相关问题的答案已经是非常普遍的了。在Google Groups搜索引擎漂亮的界面背后，古老的USENET交流方式仍然散发着其旺盛的生命力。

Goolge Groups搜索可以通过点击主页面中的Groups标签或者通过浏览http://groups.google.com来访问。虽然它的查询界面（如图1-4所示）看起来和其他的Google查询页面有点儿不同，但是搜索操作方法都是一样的。网页搜索页面和Groups搜索页面的主要不同之处在于新闻组浏览链接。

⊖ 抓取（crawl），又译作爬行，搜索引擎术语，是对搜索引擎访问网站的一个形象说法。是指搜索引擎派出机器人程序（又称为spider，智能代理等）对网页进行抓取并分析，建立索引的过程。——译者注

图1-4 Google Groups搜索页面

首先在入口域中输入搜索项，然后点击“搜索”（Search）按钮，就会转到Groups查询结果页面，它与网页搜索结果页面非常相似。

1.2.4 Google图片搜索

Google图片搜索功能可以搜索出超过10亿个（在本书写作时）符合搜索条件的图片文件。Google会尝试在图片的文件名、图片的标题、图片附近的文字以及其他Google没有公布的地方中查找搜索项，并返回一个符合搜索条件的无重复的图片列表。Google图片搜索的操作和Web搜索的操作除了一些高级搜索项不同之外，其他操作大致相同，我们将在下一章中讨论这些高级搜索项。另外，这两种搜索的结果页面也有些不同，如图图1-5所示。

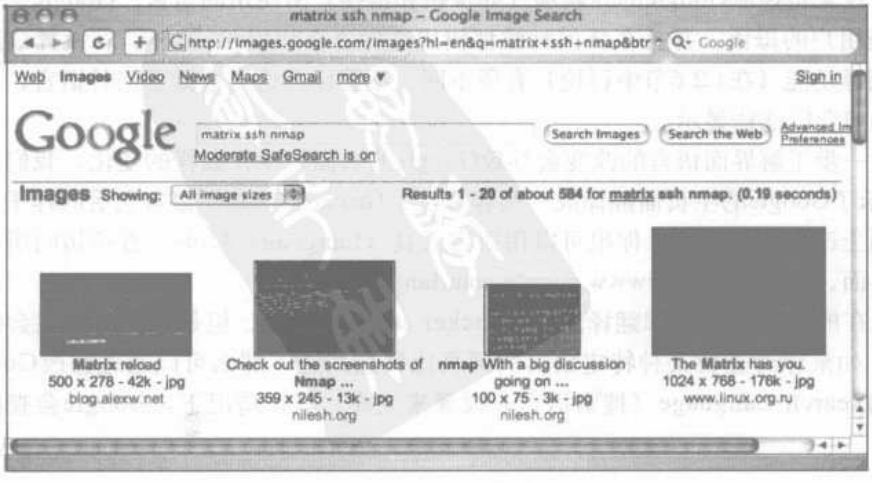


图1-5 Google图片搜索结果页面

页面页眉和Web搜索结果页面相似，只是还包括一些特定于这种页面的内容。搜索域下方的Moderate SafeSearch链接允许用户显示或者隐藏那些有可能是色情内容的图片。Showing（显示）下拉框（位于Results（结果）行中）支持用户通过尺寸来筛选显示的搜索结果图片。页眉下方，所有匹配的图片都将以缩略图形式显示，同时分别注明原始分辨率、图片大小以及图片所在的站点名。

1.2.5 Google使用偏好

你可以通过在任一Google搜索页面中点击“偏好”（Preferences）链接，或者通过浏览www.google.com/preferences来访问偏好页面。使用偏好中的这些选项主要与语言及位置设置有关，如图1-6所示。

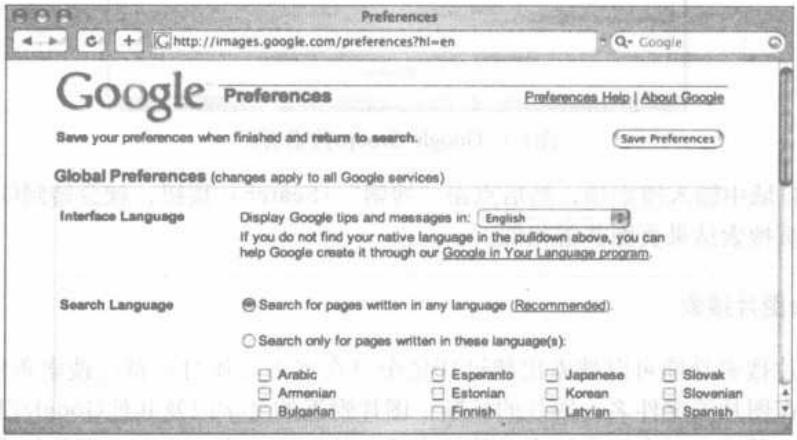


图1-6 Google使用偏好界面

Interface Language（界面语言）选项描述了Google用来显示提示和信息消息的语言。除此之外，这个设置也控制Google的导航项（如按钮和链接）所使用的语言。Google会假定用户选择的语言是用户的母语，并且在任何时候都使用这种语言和用户“对话”。设置该选项与使用Google的翻译功能（在1.2.6节中讨论）有所不同。不管你在这里选择了何种语言，用法语所写的网页都仍旧会以法语显示。

为了进一步了解界面语言的改变会导致Google的页面发生什么样的变化，我们来看一看图1-7，它显示了Google的主页面翻译成“黑客方言”（hacker speak）语言之后的样子。除了在使用偏好界面上改变设置之外，你也可以用语言工具（Language Tools）直接访问所有特定语言的Google界面。语言工具位于www.google.com/language_tools。

虽然现在的Google页面都翻译成了“hacker speak”语言，但是Google仍然会搜索各种语言的网页。如果你对查找某种特定语言的网页比较感兴趣，那么可以通过修改Google使用偏好页面上的Search Language（搜索语言）设置来完成。默认情况下，Google会查找各种语言的网页。

图1-7 翻译成“Hacker Speak”语言之后的Google主页

Google搜索背景知识

代理服务器所导致的语言恶作剧

从后面几章将会看到，当你上网时，可以使用代理服务器来隐藏你的位置和身份。Google会根据代理服务器的地理位置把Google主页改为和该服务器所位于的国家相匹配的语言。如果你的语言设置莫名其妙地改变了，那么请务必确认你的代理服务器设置。即便是经验丰富的代理用户也有可能忘记什么时候使用了代理，什么时候没有使用代理。正如我们稍后将看到的那样，语言设置可以直接通过某个URL进行修复。

使用偏好界面也可以用来更改其他的搜索参数，如图1-8所示。

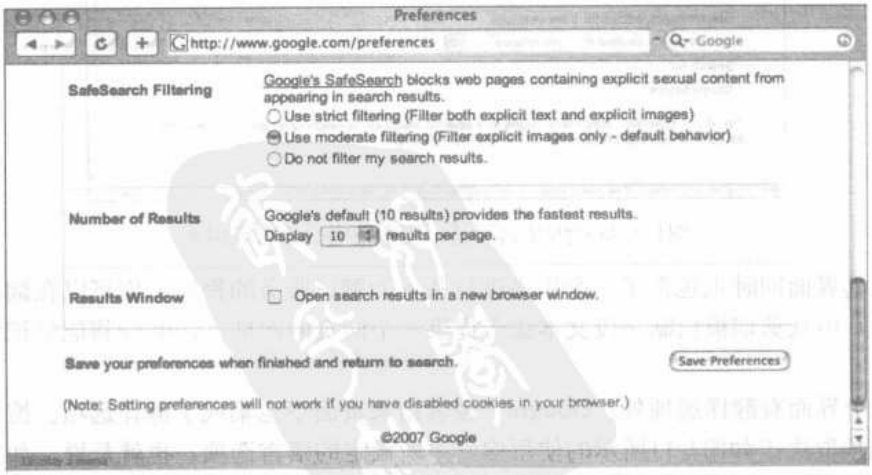


图1-8 其他的使用偏好设置

“安全搜索过滤”（SafeSearch Filtering）用来屏蔽Web搜索结果中所出现的色情内容。尽管

这在日常Web搜索中是一个很有用的选项，但是当你在执行漏洞评估搜索时，应当取消这一选项。如果色情内容所在的Web站点中的主要内容实质上并非是色情的，那么也许仅仅是网站所有人对这种内容感兴趣而已。

“结果数量”（Number of Results）设置描述了在每个搜索结果页面上显示了多少项结果。这个选项很主观，即它取决于你的喜好和网络连接速度。但是，你很快就会发现默认的每页显示10项搜索结果还不太够。如果你的网络连接相对较快，那么可以考虑选择将该项设为100，即每页显示的最大结果数。

当选择了“结果视窗”（Results Window）设置时，Google会打开一个新的浏览器窗口显示搜索结果。这个设置也由个人喜好而定。该选项的选择与否都不会有什么不好的影响，除非你的浏览器（或者其他的软件）可能会把这个新窗口当成一个弹出广告而屏蔽掉。如果你注意到在点击“搜索”（Search）按钮后，Google的搜索结果页面并没有出现，那么你应该在Google的使用偏好中取消该设置。这些更改不会被保存，除非你在浏览器中启用了cookies。

1.2.6 语言工具

可以从Google主页访问到的语言工具界面提供了各种不同的程序用来定位和翻译以不同语言书写的网页。如果你很少查找以其他语言（Language Tools）编写的网页，那么在执行此类搜索前修改使用偏好就比较麻烦了。语言工具界面的第一部分（如图1-9所示）可以让你快速搜索用其他语言编写的文档和位于其他国家的文档。

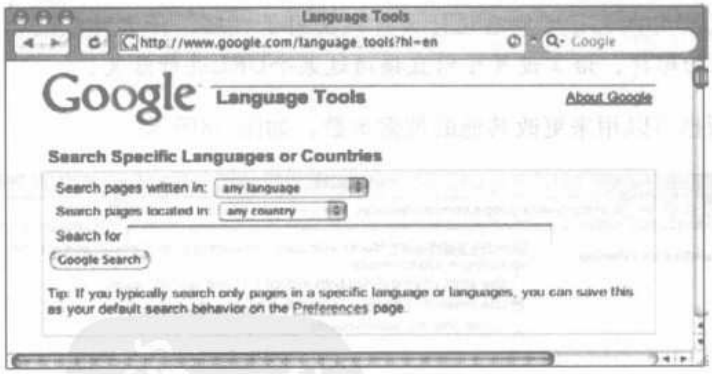


图1-9 Google语言工具：搜索特定的语言或国家

语言工具界面同时也包含了一个用来进行基本的翻译服务的程序。你可以在翻译窗体（如图1-10所示）中从剪切板粘贴一段文本或者提供一个网页的网址，Google将能够把它们翻译成各种语言。

除了这个界面有翻译选项外，Google在搜索结果页面中也集成了翻译选项。搜索结果页面中的翻译选项取决于如图1-11所示的使用偏好中所设定的语言选项。也就是说，如果你的界面语言设置为英语（English），而搜索结果页面为法语（French），那么Google将会给出把该页翻译为你的母语——英语的选项。可选的语言翻译列表如图1-11所示。

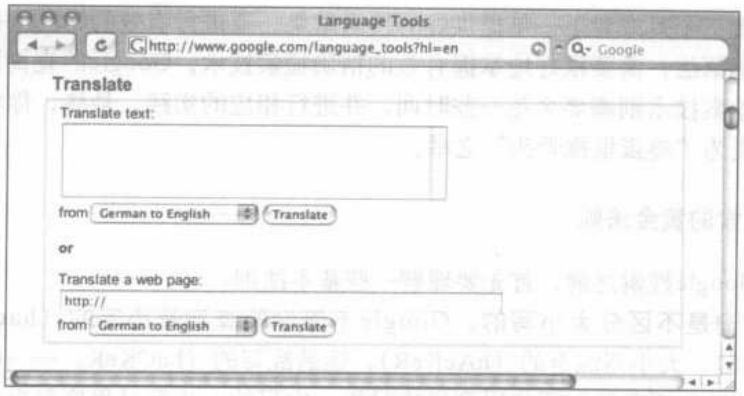


图1-10 Google翻译工具

图1-11 Google的翻译语言

Google搜索背景知识

Google工具栏

不要因为Google的“辅助”程序（如浏览器工具栏）的吸引而分散了你的注意力。你会发现完全可以从Google搜索主页面访问到所有的重要功能。每个工具栏都提供了少许的便利，如“一键目录遍历”或者“选择及搜索”功能。但是，目前有如此众多的工具栏，你不得不决定哪一种适合你的操作环境。

1.3 建立Google查询

Google查询的建立是一个过程。并没有所谓的不正确的查询。你有可能会创建一个效率低下的查询，但是，随着互联网的飞速发展以及Google缓存的不断增大，一个效率低下的查询可

能会在明天，或者下个月或者下一年提供好的搜索结果。要进行有效的Google搜索，你需要扎实地掌握基本搜索语法，需要很好地掌握有效的精确检索技术。Google的查询语法很简单，而学习有效的精确检索技术则需要多花一些时间，并进行相应的实践。最终，你将了解Google查询语法，它也将成为“鸡蛋里挑骨头”之举。

1.3.1 Google搜索的黄金法则

在我们讨论Google搜索之前，首先要理解一些基本法则：

- **Google查询是不区分大小写的。**Google不管你的查询是小写的（hackers），大写的（HACKERS），大小写混合的（hAcKeR），还是乱写的（haCKeR）——Google都同样对待它们。当你在搜索类似源代码列表的时候，字母的大小写对程序员来讲具有不同的意义，那么此时这种处理就非常重要。其中，值得注意的一个例外是单词or。当它作为布尔操作符时，or必须为大写，即OR。
- **Google通配符。**Google的通配符概念和程序员的通配符概念不同。大多数程序员认为通配符可以是任一单一字母的字符表示（UNIX爱好者可能会联想到问号）或者是用星号表示的一系列字母。这种技术称为词干提取。在Google的通配符中，星号（*）仅代表搜索词组中的一个词。在一个词的开始或者结尾使用星号和直接使用这个单词的效果相同。
- **Google保留忽略查询关键字的权利。**Google会忽略一个搜索中的某些常用单词、字母和一些单独的数字。这些词汇常称为“停用词”（stop word）。Google的基本搜索文档（www.google.com/ help/basics.html）指出这些停用词包括where和how，如图1-12所示。不过，Google似乎会在搜索中包括这些词。例如，搜索WHERE 1=1的结果要少于搜索1=1的结果。很明显搜索中包括了WHERE这个词。搜索where pig返回的结果明显少于简单地搜索“pig”，这再次表明：事实上，Google不会忽略类似how和where此类的词。有时，Google会一声不响地忽略那些停用词。例如，搜索HOW 1 = WHERE 4时返回的结果数便与搜索1 = WHERE 4的结果数相同。这似乎表明单词HOW与搜索的结果无关，Google会默认地将该单词忽略掉。忽略单词看似没有什么明显的规则，但是有时候当Google忽略一个查询项时，它会在搜索结果页面的查询输入框的下方提醒用户。

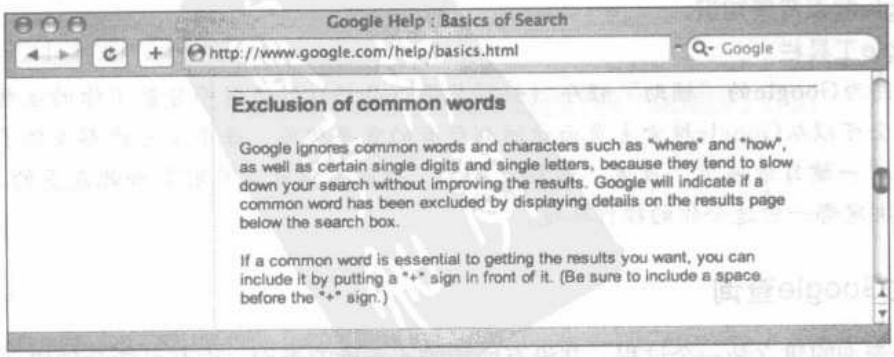


图1-12 忽略查询中的单词

强制Google使用常见词汇的一种方法是把它们用引号引起来。这样即可将该查询作为一个短语提交，并且查询结果中会包括短语中的所有单词——不管这些词是如何的常见。另外，你还可以在查询项之前添加一个+号，例如查询+and。不使用引号，同时注意不要在+号和单词and之间添加空格，那么这个查询会返回将近50亿条结果。

Google搜索背景知识

扩大搜索的规模

一个非常有趣的搜索便是查询*。这个查询能产生大约180亿条结果，号称当今产生结果最多的查询。你能打破这个记录吗？

- **32个单词的限制。**Google限制搜索关键字最多为32个单词，这与先前10个单词的限制相比已有所提升。这也包含高级操作符，我们将在稍后讨论。对于大多数用户来说这个限制已经足够用了，不过当用户需要时，还是有办法突破这一限制的。一种非常的方法便是使用用通配符（*）来代替某些单词。Google不认为通配符是一个查询项，这使得你能够稍稍扩展查询。下面来看一下如何搜索美国宪法的前几个单词：

```
we the people of the united states in order to form a more perfect union
establish justice
```

这组查询有17个单词之长。如果我们用星号（通配符）来替换其中的一些单词，然后提交为：

```
"we * people * * united states * order * from * more perfect * establish *"
```

包括引号在内，Google将会把该查询看作是一个包含9个单词的查询（如果不算通配符的话，只有8个）。也可以进一步扩展我们的查询，即只使用两个以上的实词和任意数量的通配符。

1.3.2 基本搜索

Google搜索是一个过程，其目标是查找关于某个主题的信息。这个过程以基本搜索开始。Google只会在返回相关信息的时候，才用各种方式对基本搜索进行修改。在这一过程中，Google使用网站排名（ranking）技术把排名最高的页面放到搜索结果的第一页上。排名系统的细节很复杂，并且略有水份，但就我们的目的而言，Google很少能够精准地给出我们所需要的信息。

最简单的Google查询由一个单词或者一组单词组成。一些基本的单词搜索包括：

- hacker
- FBI hacker Mitnick
- mad hacker dpak

比单个单词搜索稍复杂一点的是词组（或称“短语”）搜索。词组是由封装在双引号内的一组单词所组成。当Google碰到一个词组时，它会按照你提供的单词顺序对词组中的所有单词进行搜索。Google不会忽略词组中的常见词。词组搜索包括：

- “Google hacker”

- “adult humor”
- “Carolina gets pwnt”

我们将在下一章中看到，词组和单词搜索可以和高级操作符一起组合使用。

1.3.3 使用布尔操作符和特殊字符

虽然与基本单词搜索相比，词组搜索更为高级，但它仍然是Google的一种查询形式。为了执行高级查询，有必要先了解布尔操作符AND、OR和NOT。为了适当地对高级Google查询中各个部分进行分段，我们也必须研究使用括号的可视化分组技术。最后，我们将利用某些特殊字符将这些技术组合起来。这些特殊字符可以是某些操作符的缩写，也可以是通配符或者占位符。

如果你曾使用过其他的Web搜索引擎，那么你很可能接触过布尔操作符。布尔操作符可以用来限定查询返回的结果。由于许多搜索引擎以不同的方式来处理布尔操作符，因此即使你已经熟悉布尔操作符，也请你花点时间读一读这一节，以帮助你理解Google是如何处理这些操作符的。对这些操作符的不正确使用会完全改变返回的结果。

最常用的布尔操作符是AND。这个操作符用于在查询中包含多个关键字。例如，一个简单的查询hacker可以通过一个布尔操作符扩展为hacker AND cracker。第二个查询不仅包含关于黑客的网页，也包含黑客以及他们的“食物”的网站。一些搜索引擎要求必须使用这个操作符，而Google则不要求。AND关键字对Google来讲是多余的。默认情况下，Google会自动搜索查询中的所有关键字。事实上，当你使用了明显多余的关键字时，Google会给出警告，如图1-13所示。

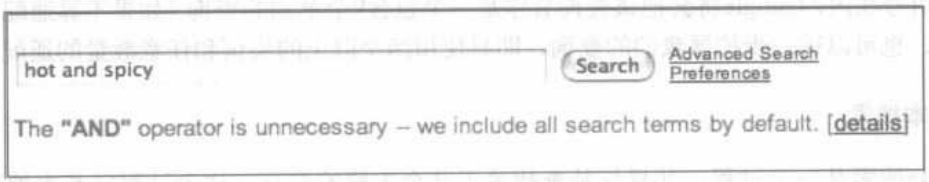


图1-13 Google的警告

注意

在你刚开始学习Google-fu[⊖]的方法时，要时刻注意Web搜索界面上的查询输入框下面所显示的信息。你可以从中学习到许多技巧提示，以改进你的查询语法。

加号（+）强制Google搜索它后面的单词。在加号后面不得有空格。例如，你打算把and、justice、for和all作为单独的、不同的单词进行搜索，那么Google会给出其中的一些单词太常用并已被忽略的警告。为了强制Google搜索这些常见的单词，需要在它们前面添加加号。在查询中过度使用加号不会产出问题。为了执行包含所有单词的搜索，可考虑使用查询+and justice for +all。除此之外，还可以把这些单词封装到双引号中。一般情况下，这可以强制Google包含词组中所有的常见词。用一个词组来表示这个查询为“and justice for all”。

⊖ Google-fu是指使用Google进行快速的、简便的、精确的搜索，“fu”取自于“Kong-fu”（功夫）。——译者注

另一个常见的布尔操作符是NOT。在功能上，它和AND操作符恰好相反，NOT操作符从查询中忽略一个单词。一种使用这个操作符的办法是在搜索单词前面加上一个减号（-）。在减号和搜索关键字之间不能有空格。考虑一个简单的查询，如hacker。这个查询非常通用，它会返回各种职业的查询结果，例如，打高尔夫的人、伐木人、连环杀手以及那些患有慢性支气管炎的人。对于这个查询，你很可能并不是对单词hacker的每种解释都感兴趣，而是想得到这个关键字更为特殊的解释。为了让查询更为精确，你可以包含更多的关键字，Google会自动地把它们AND在一起，或者也可以使用NOT来删除查询中不需要的关键字。考虑使用如hacker -golf或者hacker -phlegm这样的查询来从你的查询中删除那些令人厌恶的字符。这会让搜索的结果更加接近你所查找的hacker的解释：伐木人。或者干脆在Google Vide）中查找lumberjack song。这似乎有点跑题了。

OR是一个很少使用而且有时会让人感到迷惑的布尔操作符。OR操作符可以用管道符号（|）或者大写的单词OR来表示。它命令Google查找搜索中的一个或者另外一个关键字。尽管这看起来相当直接，例如当考虑一个简单的查询时，如hacker或者“evil cybercriminal”，但是当你使用多个AND、OR和NOT时，情况就变得很糟了。为了消除这种疑惑，你只需要把一个查询看作是一个从左到右的句子即可。忘记那些你在高中代数里所学到的运算顺序。对我们而言，AND操作符，OR操作符以及高级操作符都是具有相同的优先级的。这些因素可能会影响排名或者搜索结果页面的显示顺序，但对Google是如何处理这些搜索查询却没有任何影响。

我们来看一个非常复杂的例子。这正是我们将要在第2章中讨论的机制。

```
intext:password | passcode intext:username | userid | user filetype:csv
```

这个例子混合使用了高级操作符和OR布尔操作符来创建一个查询，读起来就像是一个书写礼貌的请求语句一样。Google将会这样来读这个请求，“查找所有文档的文本中包含password或者passcode的页面。在那些页面中，只要求显示出那些包含username、userid或者user的页面。对于那些页面，只需要显示CSV文件”。从技术上来说，那些OR符号把查询分成各种可能的查询解释，而Google并不会被这个事实所迷惑。从代数学的观点来看，这个查询在语法上是错误的，但是Google并不会为此所迷惑。针对我们学习怎样创建查询这一目的而言，所有我们要记住的就是Google是从左向右来读取查询的。

Google的那种呆板的组合布尔操作符的方法对读者而言仍然是容易迷惑的。幸运的是，Google对括号并不“感冒”（或者不受其影响）。前面的查询也可以提交为

```
intext:(password | passcode) intext:(username | userid | user) filetype:csv
```

这个查询对我们人类而言更容易阅读，而且它确实能够和那个缺少括号的令人迷惑的查询产生同样的结果。

1.3.4 搜索缩简

为了找到最精确的结果，常常需要修改搜索查询关键字来达到更为精确的搜索。虽然Google对大多数的基本搜索都努力提供非常精确的结果，我们仍然需要使用非常复杂的查询来查找包含非常精确结果的网站页面。查找这些页面需要掌握搜索缩简的技巧。即使这本书关注

的大部分内容都是搜索缩简技术及其建议，但更为重要的是，你至少应该理解搜索缩简的基本知识。我们举一个简单的例子来说明这一点，比如管理基于TCP/IP的路由协议的免费软件GNU Zebra。GNU Zebra使用一个叫作zebra.conf的文件来存储包括接口信息和口令在内的配置信息。在从网上下载完最新版本的Zebra之后，我们可以看到其内含的zebra.conf.sample文件如下：

```
! -*- zebra -*-
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
!interface lo
! description test of desc.
!
!interface sit0
! multicast
!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!
!log file zebra.log
```

如果想用Google来查找这些文件，我们可以试试如下的这个简单的搜索：

```
"! Interface's description. "
```

这可以看作是基础搜索（base search）。基础搜索应该是唯一的，且尽可能和我们所需要的结果接近，牢记这句俗语“无用信息输入，无用信息输出”。好的基础搜索是成功实现复杂的搜索缩简技术的基本要求。我们的这个基础搜索之所以是唯一的，并不仅仅是因为我们注意到了单词Interface's和description，而是我们把跟在短语后的感叹号、空格以及最后的句号作为搜索的一部分。而这正是配置文件所用的语法，所以看起来我们这样做非常合适。但是，Google没有正确地解析这个查询，这致使我们没有得出恰当的结果，如图1-14所示。

这个结果并非毫无价值，而且这个查询相对而言比较简单，但是我们先来查看zebra.conf文件。在查询中添加该文件来帮助我们缩小查询结果，进行更精确的查询。这但是我们的下一个查询：

```
"! Interface's description. " zebra.conf
```

如图1-15所示，查询结果有些许不同，但是并非更好。

第一次搜索中的seattlewireless结果链接没有了。虽然这是一个有效结果链接，但是因为配

置文件没有被命名为zebra.conf（它被命名为ZebraConfig），所以“改良后的”搜索没再查看它。因此，搜索缩简是一个大学问：不要进行会过滤掉有效结果的缩简。

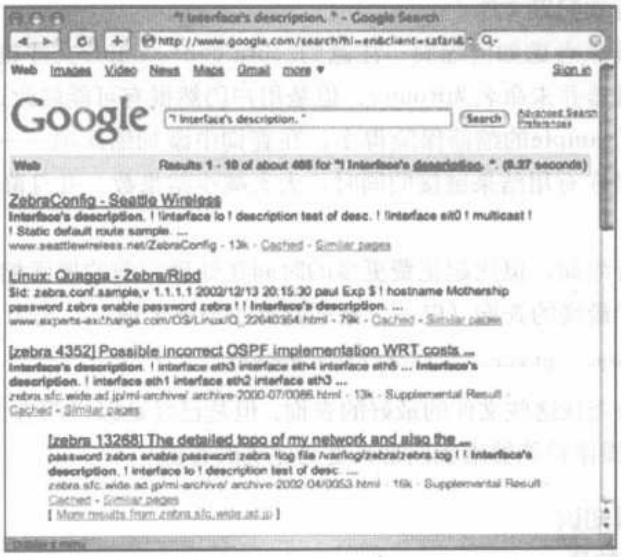


图1-14 基础搜索的处理

图1-15 搜索缩简结果

注意，图1-15中的第三个链接引用了zebra.conf.sample。这些示例文件会干扰有效查询结果，因此我们要添加到已有查询，减少包含该短语的结果链接。新的查询如下：

```
"! Interface's description. " -"zebra.conf.sample"
```

不过，这使得我们暂时步入了软件用户的后尘。此类软件安装通常会捆绑一个示例配置文件，以指导创建一个自定义配置。大多数用户只需要简单地编辑该文件，仅针对他们的环境更

改那些必须要更改的设置，并将该文件保存为.sample文件，而非.conf文件。在这种情况下，带有查询项“zebra.conf.sample”的有效配置文件仍然存在。基于该关键字的缩简查询可能会忽略以这种方式创建的有效配置文件。

还可以从另一角度来考虑如何缩简。注意到zebra.conf.sample文件包含hostname Router。虽然我们假定用户的机器并未命名为Router，但是用户仍然很有可能修改这一选项。即便如此，这仍比基于zebra.conf.sample的缩简保险得多。在查询中添加缩简项——“hostname Router”之后，可以在不牺牲潜在有用结果链接的同时，大大减少结果数，并且降低查找到潜在示例文件的结果链接的几率。

我们当然可以进行缩简，但比起花费更多的时间在处理完美的搜索缩简上，仅做一些简单的缩简就足够了。我们最终的查询（仅一个单词就有四个限定词！）变为：

```
"! Interface's description. "-"hostname Router"
```

虽然这并不是用来查找这些文件的最好的查询，但是已经足以说明查询缩简工作的原理了。在第2章中讨论的高级操作符能够给我们带来更为完美的查询！

Google搜索背景知识

故意制造不好的结果

在某些情况下，使用一些简单的Google查询语法是无可厚非的。即使Google确实具有从人性化的查询关键字忽略某些关键字的技术，但也请Google不要这么做。我们这些人类读者会感谢你！

1.4 使用Google URL

高级Google用户正是从Google的Web界面搜索域开始测试高级查询技巧的。他们不断地推敲查询关键字直到得到正确的结果为止。每个Google查询都可以用一个URL来指向搜索结果页面。Google的搜索结果页面不是静态的。它们会在你点击Search（搜索）按钮或者打开一个链接到结果页面的URL时动态创建。通过Web界面提交的查询能够打开可以用简单的URL来表示的结果页面。例如，考虑查询ihackstuff。当你输入这个查询之后，Google立即转向类似于下面的URL：

```
www.google.com/search?q=ihackstuff
```

如果你收藏了这个URL并在以后打开它，或者在你的浏览器地址栏中输入这个URL时，Google会处理你对ihackstuff的搜索并返回相应的结果。此时，这个URL不仅是一个连接到结果列表的链接，而且还是Google查询的一种简洁、美观的表达形式。任何经验丰富的Google搜索用户都能看懂这个URL并知道查询的主题。这个URL也可以相当容易地进行修改。通过把ihackstuff改成iwritestuff，Google查询就会变为查找关键字iwritestuff。这个简单的例子阐明了Google URL在高级搜索中的用处。URL的快速改变能够让变化来得更快！

Google搜索背景知识

简化URL的结构

大多数情况所需要的唯一的URL参数是一个查询（q参数）。这样就可以构造出最简单的Google URL：www.google.com/search?q=google。

1.4.1 URL语法

为了完全理解URL的强大，我们需要掌握其语法。URL的第一部分www.google.com/search是Google的搜索脚本。我把这个URL以及它后面的问号称作基本部分（base）或者叫起始URL。如果浏览这个URL，你将看到一个美观的、空白页面。search后面的问号表示参数即将传递给搜索脚本。参数是指命令搜索脚本实际所做的事情。参数之间用“&”符号进行分隔，每个参数由一个变量名、等于号(=)和该变量的值所组成。基本语法如下所示：

```
www.google.com/search?variable1=value&variable2=value
```

这个URL包含了非常简单的字符。更为复杂的URL将包含某些必须由等价的十六进制代码表示的特定字符。现在我们先来说是一下十六进制代码。

1.4.2 特殊字符

十六进制是一种低级进制。人们终究会需要在查询URL中包含一个特殊字符。当遇到需要包含特殊字符的情况时，最好是让浏览器来帮助解决此问题。大多数新型浏览器都可以自动校正输入的URL，使用等价的十六进制代码来替换特殊字符和空格。如果你的浏览器支持这个功能，那么URL构造的工作就变成非常容易了。让我们一起来做一个简单的测试。在你的浏览器的地址栏中输入如下的URL，务必在i、hack和stuff之间使用空格：

```
www.google.com/search?q="i hack stuff"
```

如果你的浏览器支持这种自动校正的功能，那么当你在地址栏中敲完回车键之后，这个URL应当会校正为www.google.com/search?q="i%20hack%20stuff"，或者校正为与其类似的URL。注意到空格字符被替换成为%20。百分号意味着接下来的两个数字表示空格字符的十六进制值，即20。有些浏览器则会做进一步转换，即把双引号改为%22。

如果你的浏览器拒绝转换这些空格，查询将无法进行。有可能你的浏览器中有一个用来修改该操作的设置，你不妨尝试着修改一下，并使用新型的浏览器。Internet Explorer、Firefox、Safari和Opera都是不错的选择。

Google搜索背景知识

快速进行十六进制转换

你可以在UNIX或者Linux机器中运行man ASCII，或者在Google中搜索“ascii table”来快速地查看字符的十六进制编码。

1.4.3 组合各个部分

构造Google搜索URL就像是拼接玩具一样。你可以从一个URL开始，然后按照你的需要来修改它以得到各种搜索结果。大多数情况下，紧接着URL基本部分的是通过Google Web界面所提交的查询。如果需要其他参数，你可以直接以任意的顺序把它们添加到URL中。如果你需要修改搜索中的参数，只需要修改参数的值并且重新提交查询即可。如果你需要删除一个参数，只需要在URL中把整个参数删除掉，然后重新提交查询。这个过程非常简单，你只需要在浏览器的地址栏中直接修改URL。所要做的只是简单地对URL进行修改，然后敲回车键。浏览器会自动地提取地址并转向一个更新后的搜索页面。你也可以通过浏览Google高级搜索页面（www.google.com/advanced_search，如图1-16所示），设置各种前面讨论过的参数来实现类似的结果，但是最后你将发现通过URL修改来完成快速的搜索调整更为迅速、简捷。

图1-16 使用Google高级搜索页面

一个Google查询URL可以包含许多不同的参数。根据你选择的选项以及你提供的搜索关键字，你会看到如表1-2中所列的部分或者全部的变量。可以根据需要对这些参数进行添加或者修改。

表1-2 Google的查询参数

变量	值	说 明
q或者as_q	搜索查询	搜索查询
as_eq	搜索关键字	将从搜索中排除的关键字
start	0到最大结果数	用于显示结果页面。结果0是第一页结果页面的第一个结果
num maxResults	1到100	每页所显示的结果数（最大为100）
filter	0或1	如果filter为0，则显示可能重复的结果

（续）

变量	值	说　明
restrict	限定代码	限定位于某个特定的国家的结果
hl	语言代码	这个参数描述了Google用来显示结果所用的语言。可以设为你的母语。查找到的页面未经过翻译
lr	语言代码	语言限定。仅显示该种语言的页面
ie	UTF-8	Web搜索的输入编码。Google建议使用UTF-8
oe	UTF-8	Web搜索的输出编码。Google建议使用UTF-8
as_epq	搜索词组	这个值作为一个确切的词组进行提交。这样便不需要给词组加上引号
as_ft	i = 包含的文件类型 e = 排除的文件类型	用于包含或排除以as_filetype表示的文件类型
as_filetype	文件扩展名	包含或者排除由as_ft的值所指定的文件类型
as_qdr	all——所有的结果 m3 = 过去3个月 m6 = 过去6个月 y = 过去一年	查找在指定时间范围内更新的网页
as_nlo	最小数	查找as_nlo和as_nhi之间的数
as_nhi	最大数	查找as_nlo和as_nhi之间的数
as_oq	词的列表	查找这些词中间的至少一个
as_occt	any = 网页内的任何地方 title = 页面标题 body = 页面文本 url = 网页内的网址 links = 在网页的链接内	在指定位置查找搜索关键字
as_dt	i = 仅包含站点或者域 e = 不包含站点或者域	包含或者排除由as_sitesearch指定的域
as_sitesearch	域或站点	包含或者排除由as_dt指定的域或站点
safe	active = 启用安全搜索 off = 禁用安全搜索	使用或不使用安全搜索
as_rq	URL	查找与该URL类似的页面
as_lq	URL	查找链接到该URL的页面
rights	cc_*	用特殊的使用权限（政府、商业、非商业等）定位页面

一些参数的值为语言限定(lr)代码。lr的值命令Google只返回指定语言的页面。例如lr=lang_ar只返回以阿拉伯语所书写的页面。表1-3列出了lr的所有可能值：

表1-3 语言限定代码

lr语言代码	语 言	lr语言代码	语 言
lang_ar	阿拉伯语	lang_it	意大利语
lang_hy	亚美尼亚语	lang_ja	日语
lang_bg	保加利亚语	lang_ko	韩语
lang_ca	加泰罗尼亚语	lang_lv	拉脱维亚语
lang_zh-CN	中文（简体）	lang_lt	立陶宛语
lang_zh-TW	中文（繁体）	lang_no	挪威语
lang_hr	克罗埃西亚语	lang_fa	波斯
lang_cs	捷克语	lang_pl	波兰语
lang_da	丹麦语	lang_pt	葡萄牙语
lang_nl	荷兰语	lang_ro	罗马尼亚语
lang_en	英语	lang_ru	俄语
lang_eo	世界语	lang_sr	塞俄维亚语
lang_et	爱沙尼亚语	lang_sk	斯洛伐克语
lang_fi	芬兰语	lang_sl	斯洛文尼亚语
lang_fr	法语	lang_es	西班牙语
lang_de	德语	lang_sv	瑞典语
lang_el	希腊语	lang_th	泰国语
lang_iw	希伯来语	lang_tr	土耳其语
lang_hu	匈牙利语	lang_uk	乌克兰语
lang_is	冰岛语	lang_vi	越南语
lang_id	印度尼西亚语		

变量hl可改变Google的消息和链接的语言。它既和限定结果页面语言的变量lr不同，也和把页面由一种语言翻译成另外一种语言的翻译服务不同。

图1-17展示把变量hl的值设为DA（Danish，丹麦语）之后，搜索单词food的结果页面。注意到Google的消息和链接都是丹麦语，而搜索的结果却用英语显示。因为我们并没有要求Google限定或者修改查询。

图1-17 使用变量hl

为了更好地理解变量hl和lr之间的不同，可以尝试把查询food重新提交为一个lr搜索，如图

1-18所示。注意URL的不同：此时的结果比之前少了许多，结果是以丹麦语所写，Google增加了一个“丹麦语网页”（Search Danish）的按钮，且Google的消息和链接语言为英语。不同于选项hl（表1-4列出了hl域的值），选项lr可以改变我们的搜索结果。我们要求Google只返回用丹麦语书写的页面。

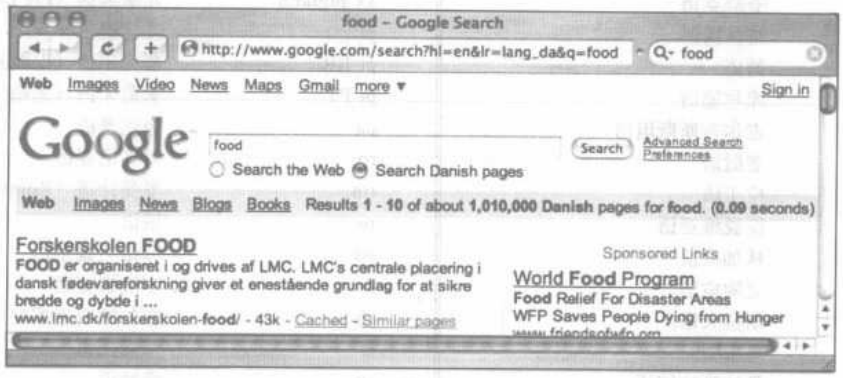


图1-18 使用语言限制

表1-4 hl语言域的值

lr语言代码	语　言	lr语言代码	语　言
af	布尔语（南非荷兰语）	en selected	英语
sq	阿尔巴尼亚语	eo	世界语
am	阿姆哈拉语	et	爱沙尼亚语
ar	阿拉伯语	fo	法罗语
hy	亚美尼亚语	tl	菲律宾语
az	阿塞拜疆语	fi	芬兰语
eu	巴斯克语	fr	法语
be	白俄罗斯语	fy	弗里斯兰语
bn	孟加拉语	gl	加利西亚语
bh	比哈里语	ka	格鲁吉亚语
xx-bork	七嘴八舌（Bork, bork, bork!）	de	德语
bs	波斯尼亚语	el	希腊语
br	布列塔尼语	gn	瓜拉尼语
bg	保加利亚语	gu	古吉拉特语
km	柬埔寨语	xx-hacker	黑客语
ca	加泰罗尼亚语	iw	希伯来语
zh-CN	中文（简体）	hi	北印度语
zh-TW	中文（繁体）	hu	匈牙利语
co	科西嘉语	is	冰岛语
hr	克罗埃西亚语	id	印度尼西亚语
cs	捷克语	ia	拉丁国际语
da	丹麦语	ga	爱尔兰语
nl	荷兰语	it	意大利语
xx-elmer	Elmer Fudd语（动画里的用语）	ja	日语

（续）

lr语言代码	语　言	lr语言代码	语　言
jw	爪哇语	ps	普什图语
kn	卡纳达语	fa	波斯语
kk	哈萨克语	xx-piglatin	儿童黑话（Pig Latin密语）
xx-klingon	格林岗语	pl	波兰语
ko	韩语	pt-BR	葡萄牙语（巴西）
ku	库尔德语	pt-PT	葡萄牙语（葡萄牙）
ky	吉尔吉斯斯坦语	pa	旁遮普语
lo	老挝语	ro	罗马尼亚语
la	拉丁语	rm	罗曼什语（Romansh）
lv	拉脱维亚语	ru	俄语
ln	林加拉语	gd	苏格兰盖尔语
lt	立陶宛语	sr	塞俄维亚语
mk	斯拉夫语	sh	塞尔维亚克罗地亚语
ms	马来语	st	塞索托语
ml	马拉雅拉姆语	sn	修纳语
mt	马耳他语	sd	信德语
mr	马拉地语	si	僧伽罗语
mo	摩尔达维亚语	sk	斯洛伐克语
mn	蒙古语	sl	斯洛文尼亚语
ne	尼泊尔语	so	索马里语
no	挪威语	es	西班牙语
nn	挪威语（尼诺斯克语）	su	巽他语
oc	奥克斯坦语	sw	斯瓦希里语
or	奥里雅语		

Google搜索背景知识

顽固的问题

hl的值是相当顽固的！这意思是说如果你在URL中更改了这个值，那么在以后的搜索中，它都一直是这个值而保持不变。把它改回去的最好的办法是通过Google使用偏好或者直接在URL中改变hl的代码。

变量restrict很容易和变量lr混淆，因为它限定你搜索特定的语言。但是，restrict和语言没有什么关系。这个变量能够把搜索结果限定在一个或几个国家，而国家的判断则是通过顶级域名（例如.us）和（或）通过服务器IP地址的地理位置。也许你会察觉到这种判断国家的方法有些不准确，是的，确实如此。尽管不准确，但是这个变量却相当有效。这次，让我们把搜索限定在JP（日语）中来搜索people，如图1-19所示。URL改变为包含限定值的值（参见表1-5），但是要注意的是，第二个结果来自www.unu.edu/，该URL的位置无从知晓。正如工具栏中显示的，实际上，主机看起来像是位于日本。

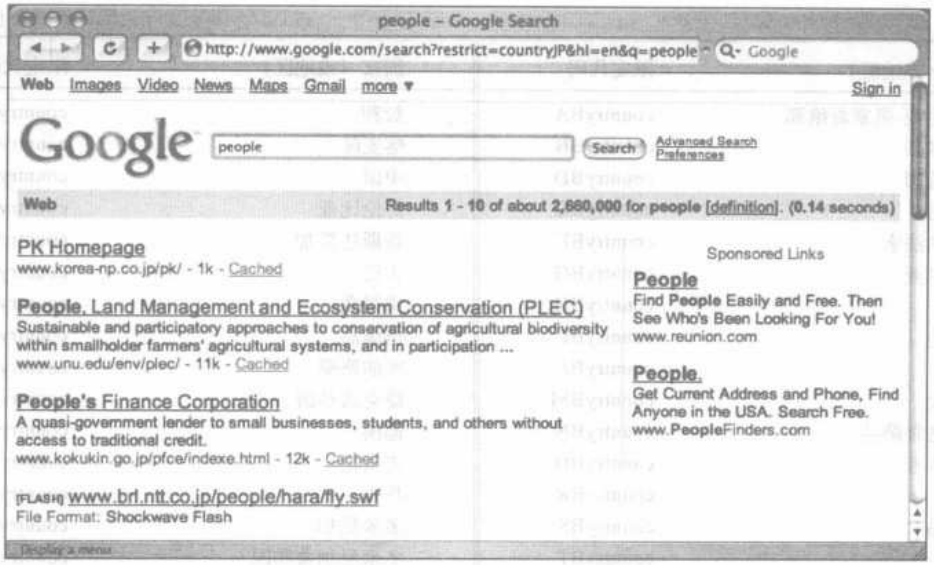


图1-19 使用restrict来让结果更为精确

Google搜索背景知识

Google是如何判断地理位置的

很容易判断主机所处的地理位置。以下将使用host和whois来判定www.unu.edu所处的地理位置：

```
wh00p:~# host www.unu.edu
www.unu.edu has address 202.253.138.42
wh00p:~# whois 202.253.138.42
role: Japan Network Information Center
address: Kokusai-Kougyou-Kanda Bldg 6F, 2-3-4 Uchi-Kanda
address: Chiyoda-ku, Tokyo 101-0047, Japan
country: JP
phone: +81-3-5297-2311
fax-no: +81-3-5297-2312
```

表1-5 restrict域值

国家（或地区）	限定代码	国家（或地区）	限定代码
安道尔	countryAD	安哥拉	countryAO
阿拉伯联合酋长国	countryAE	南极洲	countryAQ
阿富汗	countryAF	阿根廷	countryAR
安提瓜和巴布达	countryAG	美属萨摩亚群岛	countryAS
安圭拉	countryAI	奥地利	countryAT
阿尔巴尼亚	countryAL	澳大利亚	countryAU
亚美尼亚	countryAM	阿鲁巴	countryAW
荷兰安的列斯群岛	countryAN	阿塞拜疆	countryAZ

（续）

国家（或地区）	限定代码	国家（或地区）	限定代码
波斯尼亚-黑塞哥维那	countryBA	智利	countryCL
巴巴多斯	countryBB	喀麦隆	countryCM
孟加拉国	countryBD	中国	countryCN
比利时	countryBE	哥伦比亚	countryCO
布基纳法索	countryBF	哥斯达黎加	countryCR
保加利亚	countryBG	古巴	countryCU
巴林	countryBH	佛得角	countryCV
布隆迪	countryBI	圣诞岛	countryCX
贝宁	countryBJ	塞浦路斯	countryCY
百慕大	countryBM	捷克共和国	countryCZ
文莱达鲁萨兰	countryBN	德国	countryDE
玻利维亚	countryBO	吉布提	countryDJ
巴西	countryBR	丹麦	countryDK
巴哈马	countryBS	多米尼加	countryDM
不丹	countryBT	多米尼加共和国	countryDO
布维群岛	countryBV	阿尔及利亚	countryDZ
博茨瓦纳	countryBW	厄瓜多尔	countryEC
白俄罗斯	countryBY	爱沙尼亚	countryEE
伯利兹	countryBZ	埃及	countryEG
加拿大	countryCA	西撒哈拉	countryEH
科科斯（基林）群岛	countryCC	厄立特里亚	countryER
民主刚果	countryCD	西班牙	countryES
中非共和国	countryCF	埃塞俄比亚	countryET
刚果	countryCG	欧盟	countryEU
布隆迪	countryBI	芬兰	countryFI
贝宁	countryBJ	斐济	countryFJ
百慕大	countryBM	福克兰群岛（玛尔维娜）	countryFK
文莱达鲁萨兰	countryBN	密克罗尼西亚	countryFM
玻利维亚	countryBO	法罗群岛	countryFO
巴西	countryBR	法国	countryFR
巴哈马	countryBS	法属美特罗波利坦	countryFX
不丹	countryBT	加蓬	countryGA
布维群岛	countryBV	英国	countryUK
博茨瓦纳	countryBW	乔治亚	countryGD
白俄罗斯	countryBY	法属圭亚那	countryGE
伯利兹	countryBZ	加纳	countryGF
加拿大	countryCA	直布罗陀	countryGH
科科斯（基林）群岛	countryCC	格陵兰	countryGL
民主刚果	countryCD	冈比亚	countryGM
中非共和国	countryCF	几内亚	countryGN
刚果	countryCG	瓜德罗普	countryGP
瑞士	countryCH	赤道几内亚	countryGQ
科特迪瓦	countryCI	希腊	countryGR
库克群岛	countryCK	南乔治亚和南桑威治	countryGS

（续）

国家（或地区）	限定代码	国家（或地区）	限定代码
危地马拉	countryGT	摩洛哥	countryMA
关岛	countryGU	摩纳哥	countryMC
几内亚比绍	countryGW	摩尔多瓦	countryMD
圭亚那	countryGY	马达加斯加	countryMG
中国香港	countryHK	马歇尔群岛	countryMH
赫德岛和麦克康纳群岛	countryHM	马其顿，前南斯拉夫共和国	countryMK
洪都拉斯	countryHN	马里	countryML
克罗地亚（本地名：赫尔瓦次卡）	countryHR	缅甸	countryMM
海地	countryHT	蒙古	countryMN
匈牙利	countryHU	中国澳门	countryMO
印度尼西亚	countryID	北马里亚纳群岛	countryMP
爱尔兰	countryIE	马提尼克	countryMQ
以色列	countryIL	蒙特塞拉特	countryMS
印度	countryIN	马耳他	countryMT
英属印度洋领地	countryIO	毛里求斯	countryMU
伊拉克	countryIQ	马尔代夫	countryMV
伊朗伊斯兰共和国	countryIR	马拉维	countryMW
冰岛	countryIS	墨西哥	countryMX
意大利	countryIT	马来群岛	countryMY
牙买加	countryJM	莫桑比克	countryMZ
约旦	countryJO	纳米比亚	countryNA
日本	countryJP	新喀里多尼亚	countryNC
肯尼亚	countryKE	尼日尔	countryNE
吉尔吉斯斯坦	countryKG	诺福克岛	countryNF
柬埔寨	countryKH	尼日利亚	countryNG
基里巴斯	countryKI	尼加拉瓜	countryNI
科摩罗	countryKM	荷兰	countryNL
圣基茨和尼维斯	countryKN	挪威	countryNO
朝鲜	countryKP	尼泊尔	countryNP
韩国	countryKR	瑙鲁	countryNR
科威特	countryKW	纽埃	countryNU
开曼群岛	countryKY	新西兰	countryNZ
哈萨克斯坦	countryKZ	阿曼	countryOM
老挝	countryLA	巴拿马	countryPA
黎巴嫩	countryLB	秘鲁	countryPE
圣卢西亚	countryLC	法属玻利尼西亚（French Polynesia）	countryPF
列支敦士登	countryLI	巴布亚新几内亚	countryPG
斯里兰卡	countryLK	菲律宾	countryPH
利比里亚	countryLR	巴基斯坦	countyPK
莱索托	countryLS	波兰	countryPL
立陶宛	countryLT	圣皮埃尔和密克隆岛	countryPM
卢森堡	countryLU	皮特凯恩	countryPN
拉脱维亚	countryLV	波多黎各	countryPR
大阿拉伯利比亚	countryLY	巴勒斯坦	countryPS

（续）

国家（或地区）	限定代码	国家（或地区）	限定代码
葡萄牙	countryPT	塔吉克斯坦	countryTJ
帕劳群岛	countryPW	托克劳群岛	countryTK
巴拉圭	countryPY	土库曼斯坦	countryTM
卡塔尔	countryQA	突尼斯	countryTN
留尼旺	countryRE	汤加	countryTO
罗马尼亚	countryRO	东帝汶	countryTP
俄罗斯联邦	countryRU	土耳其	countryTR
卢旺达	countryRW	特立尼达和多巴哥	countryTT
沙特阿拉伯	countrySA	图瓦卢	countryTV
所罗门群岛	countrySB	中国台湾	countryTW
塞舌尔	countrySC	坦桑尼亚	countryTZ
苏丹	countrySD	乌克兰	countryUA
瑞典	countrySE	乌干达	countryUG
新加坡	countrySG	美国本土外小岛屿	countryUM
圣路易斯	countrySH	美利坚合众国	countryUS
斯洛文尼亚	countrySI	乌拉圭	countryUY
斯瓦尔巴特和扬马延岛	countrySJ	乌兹别克斯坦	countryUZ
斯洛伐克（斯洛伐克共和国）	countrySK	梵蒂冈	countryVA
塞拉利昂	countrySL	圣文森特和格林纳丁斯	countryVC
圣马力诺	countrySM	委内瑞拉	countryVE
塞内加尔	countrySN	英属维尔京群岛	countryVG
索马里	countrySO	美属维尔京群岛	countryVI
苏里南	countrySR	越南	countryVN
圣多美和普林西比	countryST	瓦努阿图	countryVU
萨尔瓦多	countrySV	瓦利斯群岛和富图纳群岛	countryWF
叙利亚	countrySY	萨摩亚	countryWS
斯威士兰	countrySZ	也门	countryYE
特克斯和凯克斯群岛	countryTC	马约特	countryYT
乍得湖	countryTD	南斯拉夫	countryYU
法属南部领地	countryTF	南非	countryZA
多哥	countryTG	赞比亚	countryZM
泰国	countryTH	扎伊尔	countryZR

1.5 总结

虽然Google的界面看起来非常简单，但是它却提供了许多强大的选项，这些选项为更为高级的搜索奠定了坚实的基础。我们可以用它搜索许多不同类型的内容，包括网页、类似于USENET的消息组、图片、视频等。建议Google搜索的初学者使用Google提供的搜索界面来搜索，同时要注意观察Google所给出的关于语法的消息和警告。OR布尔操作符可以用大写的单词OR（或者管道符|）来表示，可以借助于减号来使用NOT布尔操作符，而AND操作符通常被忽略，因为Google会自动地包含搜索中的所有关键字。可以从高级搜索页面来访问高级搜索

选项，这些选项能够让用户快速地查找到精确的结果。高级Google用户通过自定义查询、丰富的经验和常识让搜索更为精确。

1.6 快速查找解决方案

探索Google的Web界面

- Google有几个不同的搜索区域（包括网页、Groups、视频和图片搜索），每个区域都具有各自不同的搜索特征和结果页面。
- 网页搜索页面是Google的灵魂，它简洁、流畅、功能强大，即便对于大多数的高级搜索也是如此。
- Google Groups搜索帮助用户搜索到所有过去的和当前的新闻组帖子。
- 可以使用图片搜索功能来查找到接近十亿个符合搜索关键字的图片。
- Google的使用偏好和语言工具提供了自定义搜索功能、翻译服务、指定语言的搜索以及更多的功能。

建立Google查询

- Google查询的建立是一个过程，这个过程包括确定一个固定的基本搜索，扩展或缩简基本搜索以得到需要的结果。
- 熟记Google搜索的“黄金法则”。这些基础条件是成功搜索的前提。
- 合理地使用布尔操作符和特殊字符来帮助扩展或缩简搜索。它们也能够让别人更容易地读懂你的查询。

使用Google URL

- 当你提交完一个Google查询之后，可以看到相应的Google结果页面，而这个页面的URL可以用于修改一个查询或者在以后重新使用这个查询。
- 虽然可以在Google搜索URL中设置许多个不同的变量，但唯一真正需要的是变量q或query。
- 有些高级搜索选项，例如as_qdr（通过月份进行的日期限制搜索），不能很容易地在除了URL之外的其他地方设置。

1.7 网站链接

- www.google.com　Google的主页，大部分搜索的入口。
- http://groups.google.com　Google Groups页面。
- http://images.google.com/　使用Google搜索图片。
- http://video.google.com　使用Google搜索视频。
- www.google.com/language_tools　各种语言和翻译选项。
- www.google.com/advanced_search　高级搜索表单。
- www.google.com/preferences　使用偏好页面，允许用户设置如界面语言、搜索语言、安全搜索过滤以及每页显示结果数等选项。

1.8 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实际问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：有些人喜欢使用简便的工具栏。我可以在哪里获得有关Google工具栏的信息？

答：问Google。进一步讲，如果你还没有养成碰到和Google相关的问题时去问Google的习惯，那么你应当养成这个习惯。如果你能指定查询条件，那么Google几乎都能够给出答案。下面是一些流行的Google搜索工具：

平　台	工　具	位　置
Mac	Google Notifier、Google Desktop、Google Sketchup	www.google.com/mac.html
PC	Google Pack（包括IE、Firefox工具栏和Google Desktop等）	www.google.com/tools
Mozilla Browser	Googlebar	http://googlebar.mozdev.org/
Firefox、Internet Explorer	Groowe多引擎工具栏	www.groowe.com/

问：有没有什么技术我可以用来学习构建Google URL的？

答：有，有许多。首先，通过Web界面提交基本的查询，然后观察在提交之后生成的URL。在搜索结果页面，对查询稍做修改并观察提交之后URL是如何变化的。这可以概括为“如此往复”。第二种方法是使用“查询构造工具”程序。这些程序提供图形界面，你可以在其中选择搜索选项，然后在你通过其界面进行浏览时创建Google URL。请关注http://johnny.ihackstuff.com的搜索引擎Hacking论坛，尤其是“coders corner”区，许多用户都在那里讨论具有这种类型的功能的程序。

问：使用Google的界面，使用工具栏或者直接写URL，哪一种更好？

答：断言哪种技术比另一种技术更好是不妥的。这归根结底取决于个人的喜好，许多高级Google用户采用不同的方式来使用这些技术。许多冗长的Google会话都是由在www.google.com的网页界面中输入简单的查询开始的。根据其不断精确的过程，可以很容易地直接从搜索域中增加或删除查询关键字。其他一些时候，例如在使用时间范围操作符时（在第2章中进行讨论），在URL的结尾快速地添加as_qdr参数就更为容易。当你正在浏览其他网页的时候，工具栏在提供快速访问Google搜索方面表现得就更为杰出了。大部分工具栏都允许选择页面内的文本，然后在页面上右击并选择Google serach（Google搜索）来把选择的文本作为一次查询提交给Google。你所使用的技术最终取决于你的喜好以及执行搜索时的环境。

第2章 高级操作符

2.1 简介

除了在前一章中介绍的基本搜索技术之外，Google还提供了称为高级操作符的特殊关键字来帮助你进行更为高级的查询。合理地使用这些操作符，能够大量地节约查找精确信息所需要的时间，而不用浪费时间在众多的搜索结果页面中找来找去了。如果没有在查询中使用高级操作符，那么Google会在网页内的所有地方搜索你的关键字，包括标题、文本和URL等。在本章中，我们将学习如下的一些高级操作符：

- intitle, allintitle
- inurl, allinurl
- filetype
- allintext
- site
- link
- inanchor
- daterange
- cache
- info
- related
- phonebook
- rphonebook
- bphonebook
- author
- group
- msgid
- insubject
- stocks
- define

2.2 操作符语法

高级操作符无非就是构成精确搜索结果的检索项的一部分。使用高级操作符在Google中进行查询就像使用其他基本查询一样。但是，和形式相对自由的标准Google查询相比，高级操作符必须遵循相当严格的语法规范。Google高级操作符的基本语法是operator:search_term。使用高级操作符时，须牢记下面的几条规则：

- 在操作符、冒号、搜索关键字之间是没有空格的。忽略这个规范会产生不符合需求的结果，而且会妨碍Google理解高级操作符。一般情况下，Google会把不符合语法规范的高级操作符当成另一个搜索关键字。例如，如果在高级操作符intitle后面没有加上冒号和关键字，则Google只会返回包含单词intitle的网页。
- 操作符搜索的搜索关键字部分的语法和前一章中讲到的关键字语法是一样的。例如，你可以使用一个单词或者用引号引起来的词组作为关键字。如果用词组作为关键字的话，须保证在操作符、冒号和词组的第一个引号之间没有任何空格字符。
- 布尔操作符和特殊字符（例如OR和+）仍可用于高级操作符查询，但是不能把它们放在

冒号之前而把冒号和操作符分开。

- 高级操作符能够和单独的查询混合使用，但是必须遵循基本Google查询语法和高级操作符语法。某些高级操作符结合使用比其他的操作符结合使用的效果更好，而有些则完全不能结合使用。稍后我们将在本章中讨论这些限制。
- ALL操作符（以单词ALL开头的操作符）非常古怪。一般情况下，一个查询中只能使用一次ALL操作符，而且不能和其他操作符混用。

下面是一些使用了高级操作符的有效的查询实例：

- intitle:Google　这个查询将返回标题包含单词Google的页面。
- intitle:"index of"　这个查询将返回标题包含词组index of的页面。回忆前一章中学到的知识，我们可以知道这个查询也可以写成intitle:index.of，因为句号可以匹配任意字符。利用这种技术，我们可以避免键入词组中的空格以及两端的引号。
- intitle:"index of" private　这个查询将返回标题包含词组index of，且网页的任何地方（URL、标题、文本等）包含单词private的页面。要注意的是，intitle只对词组index of起作用，而不会影响单词private，这是因为引号外的第一个空格位于词组index of之后。Google把这个空格解释为高级操作符搜索关键字的结尾，然后接着处理查询中剩下的部分。
- intitle:"index of" "bakcup files"　这个查询返回标题包含词组index of，且网页的任何地方（URL、标题、文本等）包含词组backup files的页面。同样要注意到intitle仅对词组index of起作用。

语法排错

在我们深入学习高级操作符之前，先来讨论如何排除在使用这些操作符的过程中出现的语法错误。当错误发生时，Google会非常友好地告诉你错误所在，如图2-1所示。

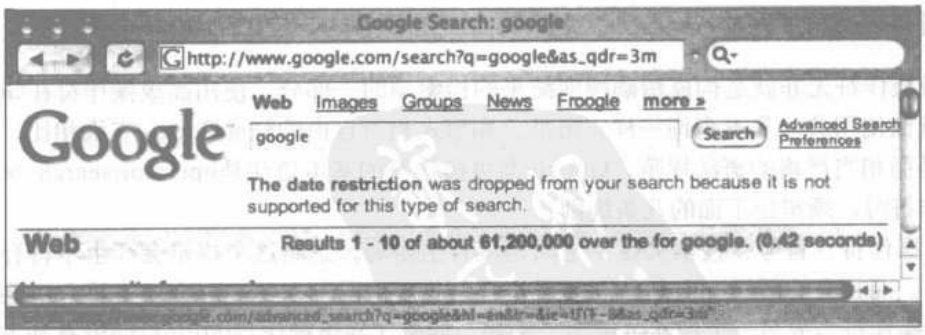


图2-1　Google给出的有用的错误信息

本例中，我们试图给URL中的as_qdr变量一个无效值。（稍后将看到，正确的语法应该是as_qdr=m3。）Google的搜索结果页面在顶部列出了搜索可能存在某种问题。这些消息通常都是解决查询字符串或者URL中的错误的关键，因此，请注意关注结果页面的顶部消息提示。我们发现人们很容易忽略结果页面中的这个位置，因为我们一般都是直接向下滚动页面来查看结果的。

但是，有些时候Google就不能起到帮助作用了，而只是返回一个空的没有错误信息的结果页面，如图2-2所示。

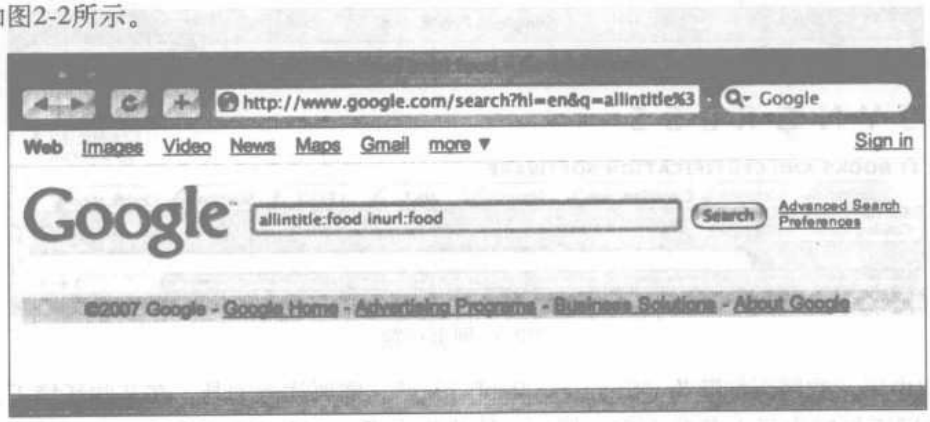


图2-2 Google的没有错误信息的空白页面

幸运的是，一旦理解了本章所讲的内容，便可以很容易解决这类问题。本例中，我们误用了allintitle操作符。大多数以all开头的操作符都不能很好地与其他操作符配合使用，例如我们提供的inurl操作。Google提供查询关键字。这个搜索让Google无法理解，所以它返回了一个空的页面。

Google搜索背景知识

但那就是我想要的！

在验证Google功力时，你需要尝试执行一个Google语法不支持的搜索。当执行这个Google语法不支持的搜索时，你必须要找到其他解决该问题的方法。不过，当下可以选择一种最简便的方式，即按Google规则办。

2.3 Google高级操作符

Google的高级操作符功能极为丰富，但是正如前例所示，并非所有的操作符都可以随处使用。有些操作符只能用于网页搜索，而另外一些则只能用于Groups搜索。2.4一节中的表2-3列出了这些差别。如果无法记清楚那些规则，你也可以关注结果页面顶部的信息提示。如果Google检测出不正确的语法，那么它会显示出错误消息，以便你知道错在何处。但有些时候则不然，此时Google因不能检测出不正确的语法而继续执行搜索。如果发生这种情况，那么需要注意结果页面，尤其是Google在结果中以粗体显示的那些单词。这些单词都是由Google所理解的搜索关键字。例如，假如你看到单词intitle以粗体字显示，那么很有可能在使用intitle操作符的过程中发生了某些错误。

2.3.1 Intitle与Allintitle：在页面标题中搜索

从技术的角度来看，页面的标题可以描述为HTML文档的TITLE标记中的文本。当浏览网页时，标题通常显示在浏览器的顶端，如图2-3所示。如果是在Google groups中进行搜索，那

么intitle会在帖子的标题中查找关键字。

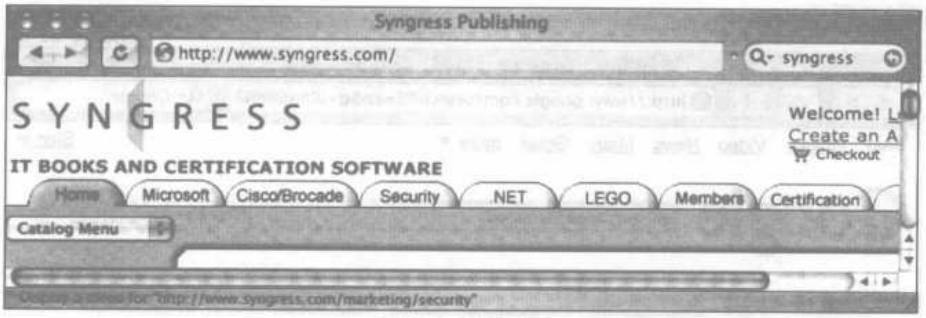


图2-3 网页标题

图2-3中所示的网页标题为“Syngress Publishing”。需要注意的是，在某些环境下，一些浏览器会在网页标题中插入某些文字。例如，让我们来看一下如图2-4所示的同一页面，这次显示的是这个页面在完成加载之前的网页标题。

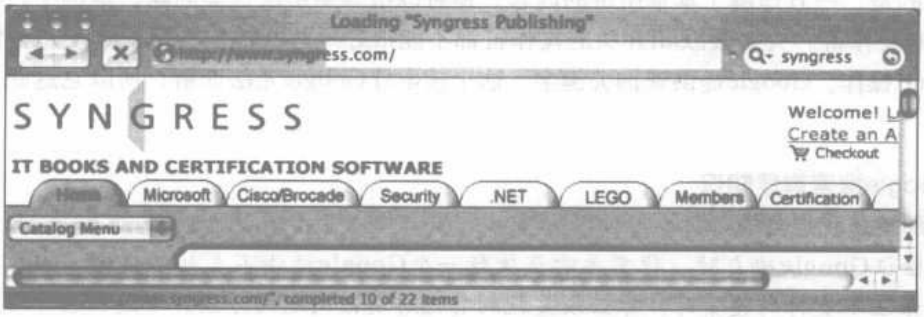


图2-4 浏览器插入的标题元素

此时，页面的标题被冠以前缀“Loading”和引号，而这是由Safari浏览器自动加上去的。在使用intitle时，必须要考虑标题中实际包含什么文本以及浏览器可能会加上的文本。

标题文本并不完全局限于TITLE HTML标记。有许多方法生成网页文档，在某些情况下，网页甚至可能根本没有标题。所以，需要记住的是所谓的标题是指在网页顶端显示的文本，可以用intitle来查找这个地方的文本。

在使用intitle的时候，尤其要注意查询字符串的语法，因为只有在单词intitle后面的单词或词组才被认为是查询关键字。而Allintitle却没有这一规则。Allintitle告诉Google，要在页面的标题中查找后面的每一个词组或者单独的单词。我们再回头看看intitle搜索的例子intitle:“index of”“backup files”。在这个查询中，第二个结果链接并不只在标题中查找关键字“backup files”，而是在文档的所有文本中都进行搜索，如图2-5所示。

如果我们把这个查询改为allintitle:“index of”“backup files”，那么将得到不同的搜索结果，如图2-6所示。

现在，每个结果链接的文档标题中都找到了“index of”和“backup files”。另外，还要注意，allintitle更为精确，仅返回intitle搜索的部分结果。

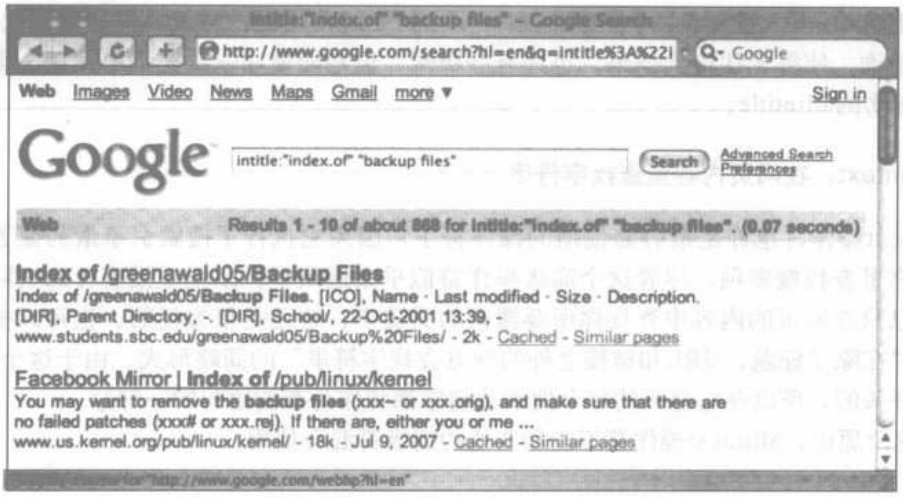


图2-5 Intitle操作符

图2-6 和Allintitle结果进行比较

Google搜索背景知识

Google关键字高亮

当你查看某个页面的Google缓存版本时，Google会使用多种颜色对关键字进行高亮着色显示，而且在搜索结果页面上使用粗体显示搜索关键字。不要因为高亮的关键字和你的搜索语法不一致而感到困惑。因为Google会对搜索结果中任何出现关键字的地方进行着色高亮显示。你也可以把Google的缓存当作一种虚拟的高亮着色工具来使用。可以通过修改Google缓存URL进行实验。在URL中找到你的搜索关键字，然后在其周围添加其他单词。如果你的操作正确并且页面中确实存在这些单词，那么Google会在页面中高亮显示那些新的单词。

慎重使用allintitle操作符。在和其他高级操作符一起使用时，它就显得非常笨拙，而且会打乱整个查询，使得无法得到结果。也许有些极端，但是即便在一次查询中使用一串intitle也好过使用拙劣的allintitle。

2.3.2 Allintext：在网页内容里查找字符串

Allintext操作符也许是最容易使用的操作符了，因为它执行了搜索引擎最为显著的功能：在网页内容里查找搜索项。尽管这个高级操作符似乎太通用了，以至于没有什么实际的用途，不过它可以只在网页的内容中查找你所要搜索的关键字，这还是很方便的。也可以把allintext作为一种“在除了标题，URL和链接之外的地方查找字符串”的简略形式。由于这个操作符是以单词all开头的，所以在它之后的每个搜索关键字都将作为查询语句的一部分。

基于这个原因，allintext操作符不能和其他的高级符混合使用。

2.3.3 Inurl与Allinurl：在URL中查找文本

在学习了intitle操作符之后，似乎看起来使用inurl操作符就应该是一件相当容易的事情了。虽然我鼓励进行随意的搜索，但是首先必须认识到URL比一个简单的网页标题要复杂得多，而且inurl操作符的工作机制也比intitle更为复杂。

首先，我们来看看空间什么是URL。URL是统一资源定位符（Uniform Resource Locator）的缩写形式，简单来说它就是网页的地址，即网址。URL的开始部分由协议及://组成，类似于常见的http://或者ftp://。紧跟在协议后面的是路径名，路径之间以正斜杠（/）分开。路径名后面是一个可选的文件名。一个常见的基本URL，如http://www.uriah.com/apple-qt/1984.html可以看作是由几个不同的组件所组成。其http协议表示希望从服务器上获取一个网页文档。服务器位于www.uriah.com，请求的文件1984.html可以在这个服务器上的/apple-qt目录中找到。正如我们在前一章中看到的那样，一个Google搜索也可以转换为一个URL，类似于www.google.com/search?q=ihackstuff。

我们已经讨论了URL中的协议、服务器、目录和文件，但是前面的示例URL中的最后一部分?q=ihackstuff需要更多的解释。简单来说，它是一个将要传递给“搜索”程序或者文件的参数列表。我们不需要涉及太多的细节，只需要简单地把这些东西理解为URL的一部分，而且可以通过inurl和allinurl操作符来让Google进行搜索。

即使到目前为止仍然看不出inurl比intitle有什么更为复杂的地方，但是确实存在一些复杂之处的。首先，Google不能有效地搜索URL的协议部分，例如http://。其次，URL中包含大量的特殊字符，而Google也不能很好地加以处理。试图在查询中以特殊的方式来包含这些特殊字符会导致不可预料的结果，而且可能会对查询造成许多不必要的限制。第三，也是最为重要的一点，其他的高级操作符（例如site和filetype）却能够在URL中的特殊的地方进行搜索，而这比使用inurl要好得多。通过比较，不难看出这些因素使得inurl比intitle更难处理。但是，不管怎样，inurl对于高级Google用户而言都是不可或缺的一个操作符，而且我们会看到本书也大量地使用了这个操作符。

和intitle操作符一样，inurl也有一个相应的操作符，叫作allinurl。参见如图2-7中所示的使用inurl操作符进行搜索的结果页面。

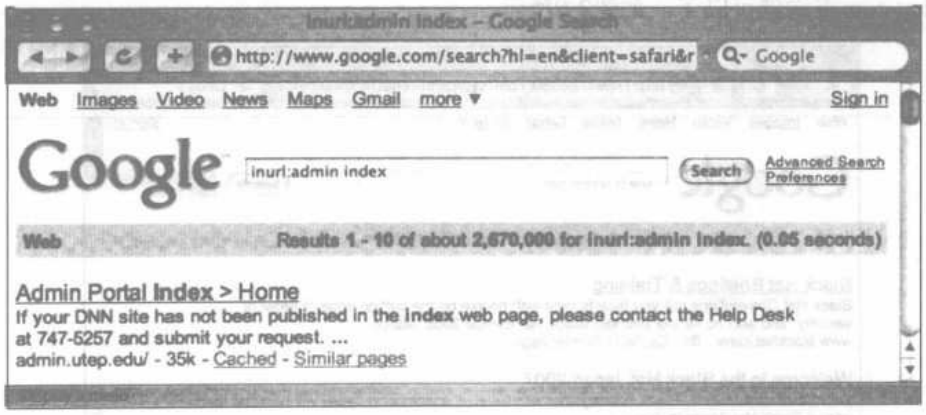


图2-7 使用Inurl进行搜索

这个查询是在文档的URL中查找单词admin，并且在文档中的任何地方查找单词index，它返回了200多万条结果。把intitle搜索替换成allintitle搜索，便可以得到如图2-8所示的结果页面。

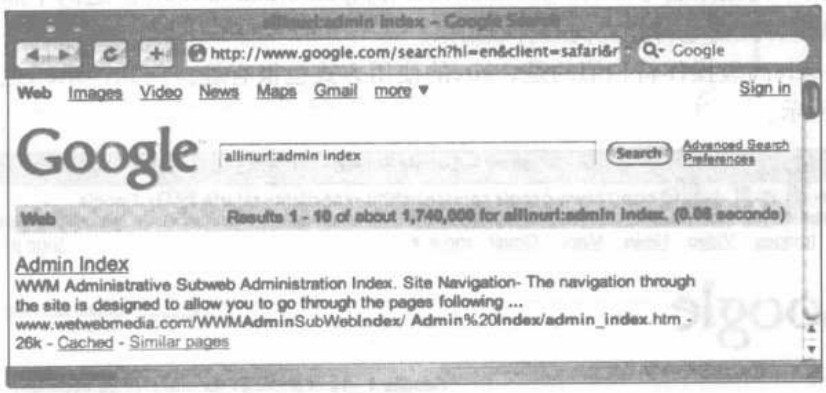


图2-8 Allinurl对比搜索结果

此时，我们让Google只在文档的URL中查找单词admin和index，它只返回了不足100百万条的结果。类似于allintitle，allinurl告诉Google只能在网页的URL中搜索它后面的单词或者词组，同时allinurl也不能很好地和其他查询共存。如果你需要在URL中查找多个单词或词组，最好在查询中使用多个inurl操作符，而不要使用不友好的allinurl。

2.3.4 Site：把搜索精确到特定的站点

虽然从技术上来讲，服务器的地址（或者域名）是URL的一部分，但是我们可以使用site操作符来对它进行更好地搜索。site允许你仅在某个特定的网站或者特定的域中搜索网页。尽管这看起来很简单，但是如何合理地使用site操作符还需要多注意一些问题，这是因为Google

是从右向左读取服务器名的，而这恰好与人类从左向右读取站点名相反。考虑一个普通的网站服务器名，www.blackhat.com。为了查找位于blackhat.com上的网页，我们只需要一个简单的site:blackhat.com查询就可以了，如图2-9所示。

图2-9 Site操作符的基本用法

注意到前两个结果来自于www.blackhat.com和japan.blackhat.com。这两个服务器都是以blackhat.com结尾并且都是符合查询的有效结果。

与很多Google高级操作符相似的是，site的使用方式也很有趣。例如，site:r这个搜索的结果，如图2-10所示。

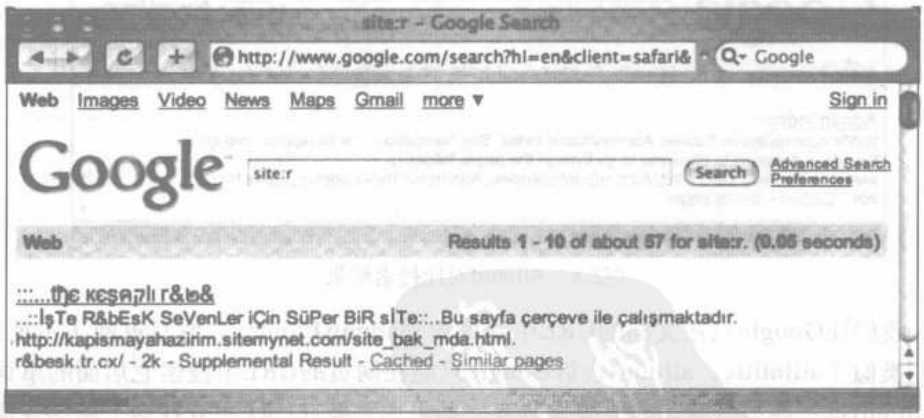


图2-10 Site的不恰当使用

仔细观察这个查询的结果，我们会发现第一个返回的结果的URL看起来有些奇怪。事实上，它确实很不合理。Google（以及整个互联网）都是从右向左读取服务器名（更确切地说是域名）的，并不是从左向右。因此，对site:r进行的Google查询不可能返回有效的结果，因为根本不存在.r域名。但是，为什么Google会返回结果呢？这很难判定，但是有一件事可以肯定的是：这些古怪的搜索以及它们相关的响应都将被那些高级的搜索引擎用户所关注，并激发更深层次的针对于此的研究。

Google搜索背景知识

Googleturd

那么，Google返回的那个链接r&besk.tr.cx是怎么来的？它是什么？作者使用术语googleturd来描述这些由Google给出的类似于印刷错误的链接。基于特定的未知环境，类似于此的奇怪的链接有时候也会保存下来。我们稍后将看到，googleturd也是有用的。

我们会在本章的后面看到，site操作符可以很方便地和其他搜索操作符混合使用。

2.3.5 Filetype:搜索指定类型的文件

Google不但能搜索网页，还能够搜索许多不同类型的文件，包括PDF（Adobe Portable Document Format，Adobe便携文档格式）和Microsoft Office文档。filetype操作符可以帮助你搜索这些文件。更为特别的是，filetype也可以查找以特定文件扩展名结尾的网页。文件扩展名是URL的一部分，它位于文件名的最后一个句号之后，参数列表的开头的问号之前。一般来讲，文件扩展名表示了能够用何种类型的程序来打开文件，因此可以通过Google的filetype操作符搜索特定的文件扩展名来寻找特定类型的文件。表2-1列出了Google所搜索的主要几种文件类型，来源于www.google.com/help/faq_filetypes.html#what。

表2-1 Google搜索的主要文件类型

文件类型	文件扩展名
Adobe Portable Document Format	Pdf
Adobe PostScript	Ps
Lotus 1-2-3	wk1、wk2、wk3、wk4、wk5、wki、wks、wku
Lotus WordPro	Lwp
MacWrite	Mw
Microsoft Excel	Xls
Microsoft PowerPoint	Ppt
Microsoft Word	Doc
Microsoft Works	wks、wps、wdb
Microsoft Write	Wri
Rich Text Format	Rtf
Shockwave Flash	Swf
Text	ans、txt

表2-1并没有列出Google能够搜索的所有文件类型。根据http://filext.org，目前已知有几千种文件扩展名。Google在它的数据库当中对所有的文件扩展名都有相应的例子！这意味着Google能够抓取任意一种扩展名的页面，但是同样需要明白的是，Google可能无法搜索那些目前还未知的文件类型。虽然表2-1列出了Google搜索的最主要的几种文件类型，但是你还可能想知道在几千多种文件扩展名之中，哪些在网络中是最为常见的。表2-2列出了可以在网络中搜索到的排名前25的文件扩展名，表中按该种文件类型的搜索结果数排名。

工具和陷阱

如何做到

表2-2中的数据有两个来源：filext.org和Google。首先，我使用lynx来抓取filext.org Web站点的一部分，以便编译一个已知文件扩展名列表。例如，这一行代码将会提取所有以字母A开头的文件扩展名，并且将它们输出到一个名为extensions的文件中。

```
lynx -source "http://filext.com/alphalist.php?extstart=%5EA" | grep "<td
width=\"120\"" | awk -F "file-extension/" '{print $2}' | awk -F "\""
'{print$1}' > extensions
```

接着，所有扩展名将通过Google的filext搜索查找，并集中显示在结果（Results）行。

```
for ext in `cat extensions`; do lynx -dump
"http://www.google.com/search?q=filetype:$ext"|grep Results|grep "ofabout";
done
```

这个过程会导致上万条查询，花上几个小时。但是，Google非常友善，并没有把我记入严重违反使用条款的黑名单。

表2-2 前25种文件扩展名，来源于Google

2004		2007	
扩展名	搜索结果数目（大约）	扩展名	搜索结果数目（大约）
HTML	18 100 000	HTML	4 960 000 000
HTM	16 700 000	HTM	1 730 000 000
PHP	16 600 000	PHP	1 050 000 000
ASP	15 700 000	ASP	831 000 000
CGI	11 600 000	CFM	481 000 000
PDF	10 900 000	ASPX	442 000 000
CFM	9 880 000	SHTML	310 000 000
SHTML	8 690 000	PDF	260 000 000
JSP	7 350 000	JSP	240 000 000
ASPX	6 020 000	CGI	83 000 000
PL	5 890 000	DO	63 400 000
PHP3	4 420 000	PL	54 500 000
DLL	3 050 000	XML	53 100 000
PHTML	2 770 000	DOC	42 000 000
FCGI	2 550 000	SWF	40 000 000
SWF	2 290 000	PHTML	38 800 000
DOC	2 100 000	PHP3	38 100 000
TXT	1 720 000	FCGI	30 300 000
PHP4	1 460 000	TXT	30 100 000
EXE	1 410 000	STM	29 900 000
MV	1 110 000	FILE	18 400 000
XLS	969 000	EXE	17 000 000
JHTML	968 000	JHTML	16 300 000
SHTM	883 000	XLS	16 100 000
BML	859 000	PPT	13 000 000

自扩展名查询伊始，三年间的改变已经相当之巨。你看Google反馈的结果数有如此之多！查询结果的进步让人瞠目结舌。如果你对某些扩展名不熟悉，请访问www.filext.com，这是一个了解文件扩展名详细信息的一个相当好的网站，在这里，你可以了解到扩展名是什么，以及它们可以与何种程序关联。

提示

ext操作符可以用来替代filetype。filetye:xls查询与ext:xls等价。

Google会把它搜索到的每个文档都转换为HTML或者文本文件以方便在线查看。你可以从图2-11中看到一个Google搜索到并转换过的文件。

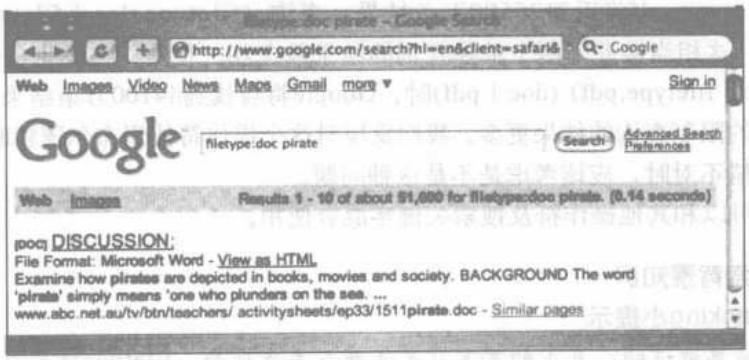


图2-11 在搜索页面中的文件转换

注意到第一个结果在文档的标题前列出了[DOC]字样，并且也指出了文件格式为Microsoft Word。这表示Google把这个文件识别为一个Microsoft Word文档。除此之外，Google也提供了一个查看HTML版的链接，点击这个链接之后，你会看到该文件的HTML形式，如图2-12所示。

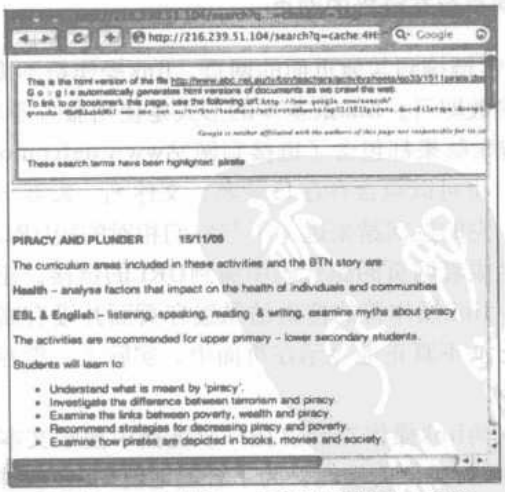


图2-12 一个由Google转换过的Word文档

当你点击Google已经转换过的文档的链接时，会在页面顶部显示一个标题，提示你正在浏

览该页面的HTML版。Google当然也提供了原文件的链接。如果你认为这看起来像是页面的缓存视图，那么你说对了。这是原始页面的缓存版本，同时也是转换后的HTML版。

虽然这些功能都很好，但是Google并不是完美的。请记住下面的几句话：

- Google并不总是提供页面转换版本的链接。
- Google并不总能正确地识别出文件类型，即便对最常见的文件类型而言也是如此。
- 当Google抓取到一个以特定文件扩展名结尾的内容为空的文件时，Google仍会给出一个有效的文件以及把这个文件转换后的链接。即便一个空白的Word文档的HTML版本也是空白的，Google也会这么做。

当OR操作符和filetype操作符混用时，就会产生问题了。例如查询filetype:doc返回3900万条结果。查询filetype:pdf将返回25500万条结果。查询（filetype:doc | filetype:pdf）能返回33500万条结果，这相当接近于两个单独查询的结果数之和。但是，当你把前一个查询改为类似于(filetype:doc | filetype:pdf) (doc | pdf)时，Google将会找到44100万条结果：得到的结果甚至比原来的更少的限制查询的结果更多。我们发现对这个操作符使用布尔逻辑时，经常有问题，所以当你发现事情不对时，应该考虑是不是这种问题。

这个操作符可以和其他操作符及搜索关键字混合使用。

Google搜索背景知识

Google Hacking小提示

我们无法简单地证明：真正的黑客总是活跃在灰色地带。filetype操作符给那些真正的Google黑客开辟了另一片有趣的天地。考虑查询filetype:xls -xls。这个查询不应该返回结果，因为所有的XLS文件的URL中都包含XLS，对吗？错。在本书写作之时，这个查询确实给出了7000多条结果，别的不说，至少这些结果都是相当有趣的。

2.3.6 Link：搜索与当前网页存在链接的网页

link操作符可以用来搜索链接到其他页面的网页。link操作符不需要搜索关键字，只需要一个URL或者服务器名作为参数即可。link最基本的形式是使用服务器名，如图2-13所示。

图2-10中所示的每个搜索结果都包含了链接到网站www.defcon.org的HTML链接。link操作符不仅能包含基本的URL，也可以包含存在目录名、文件名、参数等元素的完整URL。但是要记住的是长URL非常特殊，它的返回结果远少于与它们相对的短URL。

可以在浏览器的状态栏或者网页的源代码中看到URL的链接。正是由于这种原因，所以和其他缓存页面不一样的是，link操作符的搜索结果缓存页面并没有高亮搜索关键字，这是因为搜索关键字（链接的网站）从不真正地显示在页面中。实际上，缓存标题根本就没有提到你的搜索关键字，如图2-14所示。

一个最常见的误解是认为link操作符能够在一个链接中搜索文本。稍后我们将会看到实际上是由于inanchor操作符能执行类似的功能。为了合理地使用link操作符，你必须提供一个完整的URL（包括协议、服务器、目录以及文件），或者一个部分URL（只包括协议和主机），或者一个简单的服务器名；否则，Google会返回不可预料的结果。例如，考虑查询link:linux，它返

回15100条结果。这个查询并不符合link搜索的语法，因为这个域名是无效的。这个查询的正确的语法应该类似于link:linux.org（返回317条结果）或者link:linux.org（没有结果）。这些数字看似没有什么意义，当然，它们也无法说明原始查询返回151000条结果的原因。那么对link:linux而言，Google返回的结果是什么？图2-15和图2-16给出了这个问题的答案。

图2-13 Link操作符

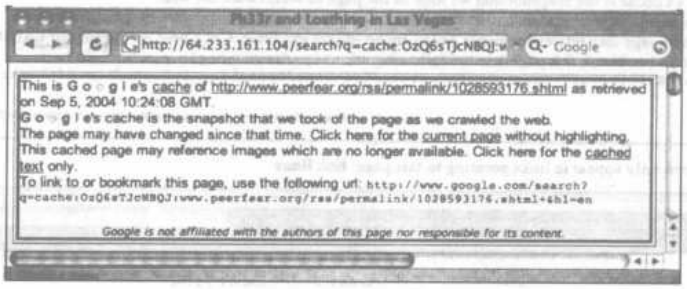


图2-14 一个常见的Link搜索缓存标题

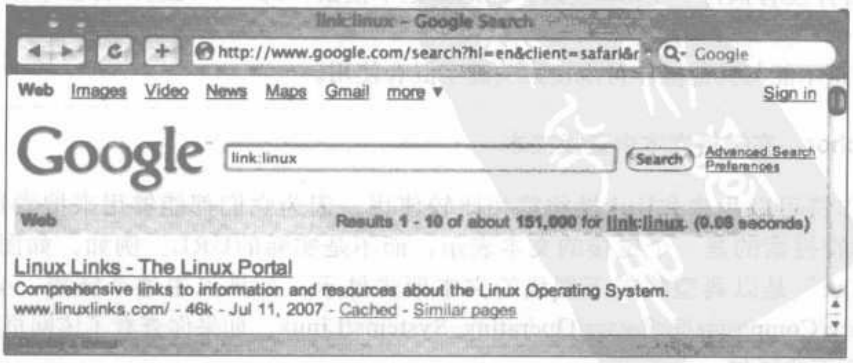


图2-15 link:linux返回151000条结果

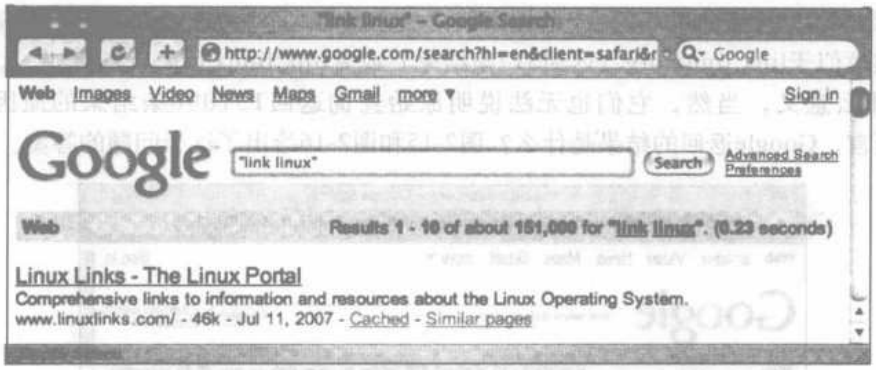


图2-16 “link linux”同样返回151000条结果

当提交一个无效的link:语法给Google时，它会把这个查询看作为一个词组搜索。Google通过缓存页面给出另外一个提示它是怎样处理无效link搜索的线索。如图2-17所示，由link:linux搜索得到的网站的缓存标题并不像典型的link搜索缓存标题，而更像是一个标准的带有高亮关键字的标准搜索缓存标题。

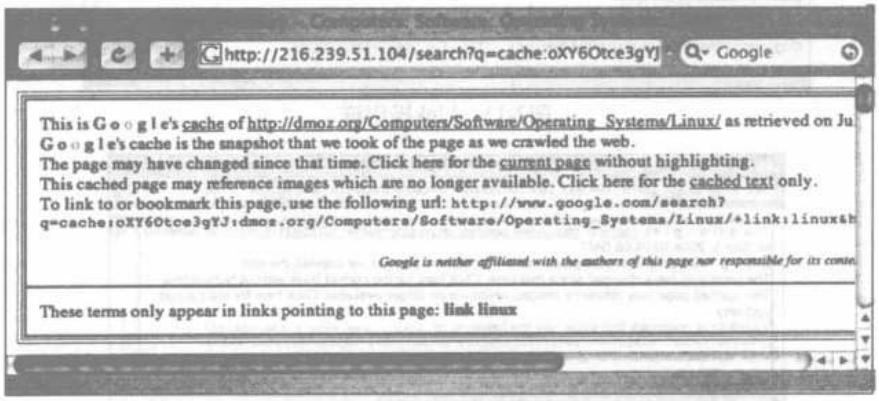


图2-17 一个无效的Link搜索页面

这意味着并没有执行一个link搜索，而是把这个搜索作为一个词组，将冒号作为分隔单词的标记。

link操作符不能与其他操作符或搜索关键字混合使用。

2.3.7 Inanchor：在链接文本中查找文本

这个操作符可以用来和link操作符相比较使用，因为它们都能够用来搜索链接。但是inachor操作符搜索的是一个链接的文本表示，而不是实际的URL。例如，如图2-17所示，“current page”是以典型的带下划线的文本形式显示。如果点击这个链接，会转到URL http://dmoz.org/Computers/Software/Operating_Systems/Linux。如果你查看了该网页的源码，会看到类似于下面的语句

```
<A HREF="http://dmoz.org/Computers/Software/Operating_Systems/Linux/">current
page</A>
```

inanchor操作符搜索的是锚点，或者说是链接上显示的文本，在本例中，即是“current page”。这与使用inurl利用“inurl:Computers inurl:Operation Systems”查询查找该页面不同。

Inanchor的参数可以是一个单词或者词组，如inanchor:click或inanchor:James.Foster。我们会在后面看到这个操作符的方便之处，尤其是在搜索网站之间的关系时。inanchor操作符可以和其他操作符及搜索关键字混合使用。

2.3.8 Cache：显示网页的缓存版本

正如我们前面讨论过的，Google保存了它所抓取到的网页的快照，我们可以通过搜索结果页面的缓存链接来访问。如果你想直接打开某个页面的缓存版本，而不用先执行一次Google搜索来得到结果页面上的缓存链接的话，只需要简单地使用cache高级操作符即可，如cache:blackhat.org或cache:http://www.netsec.net/content/index.jsp。如果你没有正确地提交一个完整的URL或者主机名，Google会返回不可预料的结果。和link操作符一样，如果给cache传递一个无效的主机名或者URL，那么Google也会把这个查询当成是一个词组搜索。查询cache:linux会返回与查询“cache linux”同样的结果，这说明Google确实是把该缓存搜索当作一个标准的词组搜索来处理的。

尽管会得到不可预期的搜索结果，cache操作符还是可以与其他的操作符以及搜索关键字混合使用的。

2.3.9 Numrange：搜索数字

numrang操作符需要两个参数，一个是最小数，一个是最大数，以破折号分隔。这个操作符非常强大，但是一旦被恶意的Google黑客所使用，也是非常危险的。正如名字中所说的那样，numrang可以用来查找某一范围内的数字。例如，为了查找数字12345，只要使用查询numrang:12344-12346即可。在搜索数字时，Google会忽略像货币标记和逗号之类的符号，这使得很容易在网页内搜索数字。这个操作符也有一种缩写的版本。你可以简单地在查询中提供两个数字，并且它们之间用两个句点分开，而不用使用numrang操作符。例如，前面一个例子可以表示为12344..12346。注意numrang操作符完全独立于查询之外。

这个操作符可以和其他的操作符以及搜索关键字混合使用。

Google搜索背景知识

危险的Google黑客！

如果由灰袍巫师甘道夫[⊖]来写这个部分的话，他可能会忍不住地说类似于这样的话“在Google缓存的黑暗地带存在比字符更为邪恶的东西”。最能说明Google力量强大的例子就是numrang操作符。如果把那些强大的查询都告诉你们，那这对我们来讲是十分不负责任

⊖ 电影指环王中的人物。——译者注

的。幸运的是，这个操作符的危害已经在http://johnny.ihackstuff.com上的“搜索引擎Hacking”论坛中那些勤奋的成员的努力下受到了控制。这个社区的成员们一次又一次用他们的语言来告诉人们Google黑客的危险，但没有泄露Goolge黑客是如何进行攻击的，也没有因此制造出更多黑客。这一部分是献给他们的！

2.3.10 Daterange：查找在某个特定日期范围内发布的网页

虽然daterang操作符有些笨拙，但它还是很有用的，而且也值得去了解。你可以使用这个操作符来查找由Google索引的在某一特定日期范围的网页。Google每次抓取一个页面，都会相应地改变其日期。如果Google查找到了某些非常生僻的网页，它可能只抓取一次，而不会重新进行索引。如果你发现你的搜索充斥着这种生僻的网页，就可以通过有效地使用daterang操作符来把这些网页剔除掉（而得到更新的结果）。

这个操作符的参数通常也必须表示为一个范围，即用破折号分开的两个日期。如果你只想查找某个特定日期索引的网页，必须提供同样的日期两次，并以破折号分开。如果你觉得这听起来太简单了以至于无法相信它是否真得这么简单，那么你说对了。它确实非常容易。传递给这个操作符的两个日期必须使用儒略历（Julian date）。儒略历是指距公元前4713年1月1日所经过的天数。例如，日期2001年9月11日以儒略历表示为2452164。因此，为了查找Google在2001年9月11日索引的，包含词组“osama bin laden”的网页的查询可以写成`daterang:2452164-2452164 "osama bin laden"`。

Google没有公开宣布支持daterang操作符，如果公开支持的话，那就更好了。Google更希望你使用高级搜索表单中的日期限制，你可以在http://www.google.com//advanced_search上找到。正如我们在上一章中讨论的那样，这个表单会在URL字符串中创建特殊的域来执行特殊的功能。Google设计了as_qdr域来帮助你查找在某一时间范围内更新的网页。例如，查找在过去的3个月里更新的并且包含单词Google的页面，可以使用查询http://www.google.com/search?q=google&as_qdr=m3。

这也许是一个更好的代替笨拙的daterange操作符的限定日期的方法。但是要知道它们都是各不相同的功能。Daterange并不是as_qdr的等价高级操作符，而且更为遗憾的是，根本没有操作符与之等价。如果你想查找在过去一年或少于一年中的更新的页面，那么就必须使用Google的高级搜索界面或者在URL的结尾加上&as_qdr=m3（或与之类似的值）。

daterange操作符必须和其他搜索关键字或高级操作符同时使用。单独使用时，它不会返回任何结果。

2.3.11 Info：显示Google的摘要信息

info操作符显示出某个站点的摘要信息以及其他Google搜索到的可能和该站点相关的链接，如图2-18所示。该操作符的参数必须是一个有效的URL或者网站名。你也可以通过把一个网站名或者URL作为一次查询提交来实现同样的功能。

如果你没有提供一个完整的URL或者主机名，Google可能会返回不可预料的结果。正如link和cache操作符一样，Google会把无效的主机名或者URL参数作为词组搜索来处理。

info:linux查询会返回与“info linux”查询同样多的结果，这表明Google确实把该info搜索当成一个标准的词组搜索来处理。

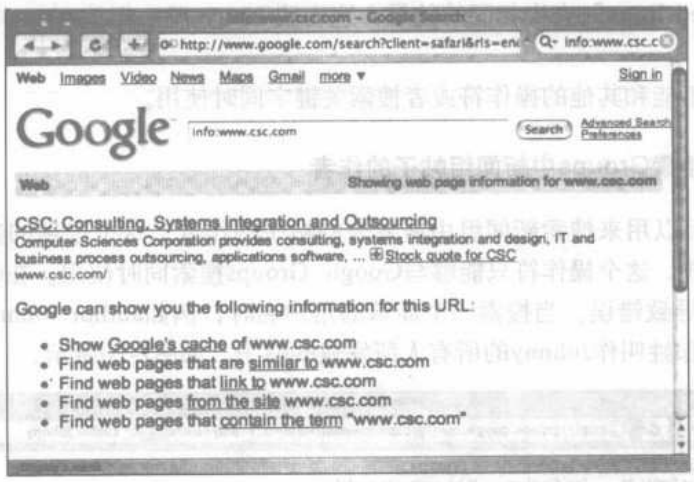


图2-18 一个Google Info查询的输出

info操作符不能和其他操作符或者搜索关键字混合使用。

2.3.12 Related：显示相关站点

related操作符会列出Google断定的和某个网站相关的那些站点，如图2-19所示。该操作符的参数为一个有效的网站名或者URL。你也可以通过点击任一搜索结果页面上的类似网页链接，或者使用高级搜索表单中的“Find pages similar to the page”（搜索类似以下网页的网页）来实现同样的功能（如图2-19所示）。

图2-19 动作相关

如果你没有提供一个完整的URL或者主机名，Google会返回不可预料的结果。将无效的主机名或者URL作为参数传递给related无异于将该查询作为一个词组搜索来提交。related:linux查询会返回与“related linux”查询相同的结果，这表明Google确实把该related搜索看作为一个标准的词组搜索。

related操作符不能和其他的操作符或者搜索关键字同时使用。

2.3.13 Author：搜索Groups中新闻组帖子的作者

author操作符可以用来搜索新闻组中所发表的帖子的作者。这个选项的参数由一个用户名或者E-mail地址组成。这个操作符只能够与Google Groups搜索同时使用。在Groups搜索之外单独使用这一搜索会导致错误。当搜索一个简单的用户名时，例如author:Johnny，其搜索结果会包含由名字，中名和姓叫作Johnny的所有人所发表的帖子，如图2-20所示。

图2-20 Author:Johnny查询

我们可以看到，结果中有Johnny Lurker、Johnny Walker、Johnny和Johnny Anderson。你可能想知道这些是否都是真名呢？一般情况下，这都不是真名。这是新闻组的特征。任何人都能够通过Google在新闻组中发布帖子，而只需要一个免费的E-mail账号来进行验证，这时用随机匿名性来维护就相当容易。

author操作符用起来有些不方便，因为它并不像其他操作符一样解释其参数。author:Johnny或者author:Johnny@ihackstuff.com这样简单的查询没有什么问题，但是当搜索词组形式的用户名时就会出现问题。让我们来尝试一下类似的搜索author:”Johnny Long”，搜索姓名为Johnny Long的作者。这个搜索确实出现了问题，如图2-21所示。

然而，通过author:Johnny Long查询，我们得到了我们预期的结果，如图2-22所示。

author操作符可以和其他有效的Groups操作符及搜索关键字混合使用。

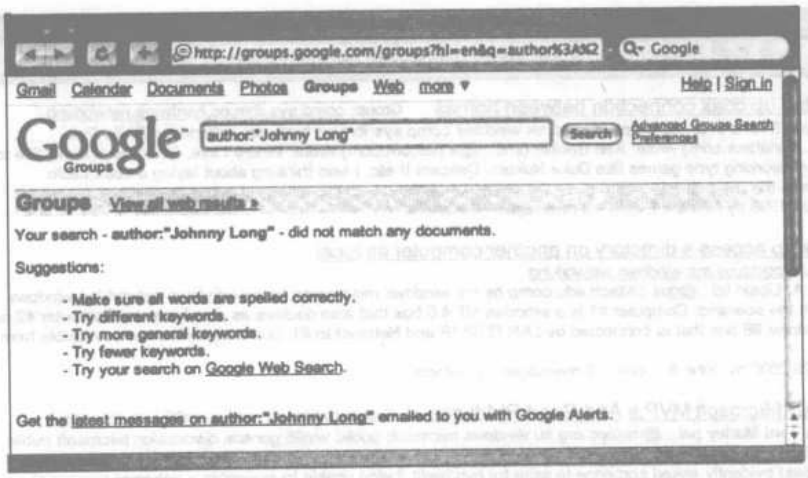


图2-21 词组搜索与Author不能混用

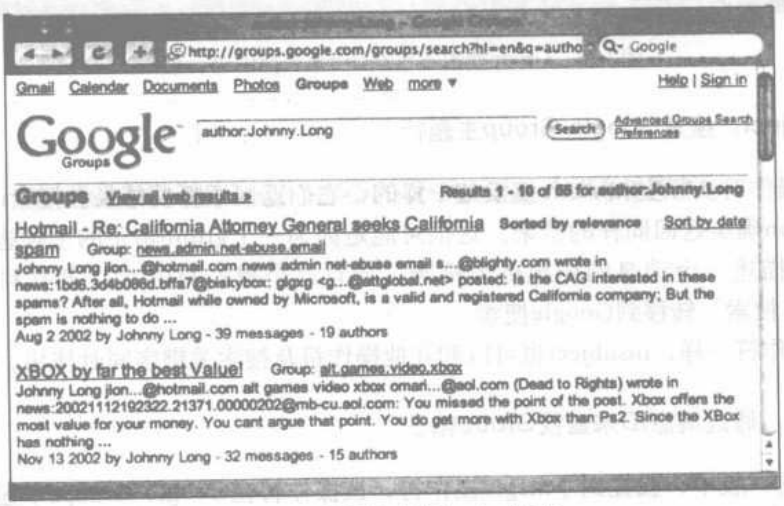


图2-22 使用句点的Author搜索

2.3.14 Group：搜索Group标题

这个操作符可以用来在Google Groups帖子的标题中搜索含有关键字的帖子。这个操作符仅用于Google Groups。它是能够很好地和通配符匹配使用的操作符之一。例如搜索以forsale结尾的Groups，使用group:*.forsale即可。在某些情况下，Google并不是在group的实际名称中查找关键字，而是在描述该group的关键词中进行搜索。让我们来一起看一下如图2-23所示的查询group:windows。虽然并不是所有的结果都包含单词windows，但是所有返回的group都是讨论Windows软件的。

以我们的经验来看，group操作符并不能很好地和其他操作符混合使用。如果当你混用group操作符而得到奇怪的结果时，不妨使用其他的如intitle之类的操作符来进行查找。

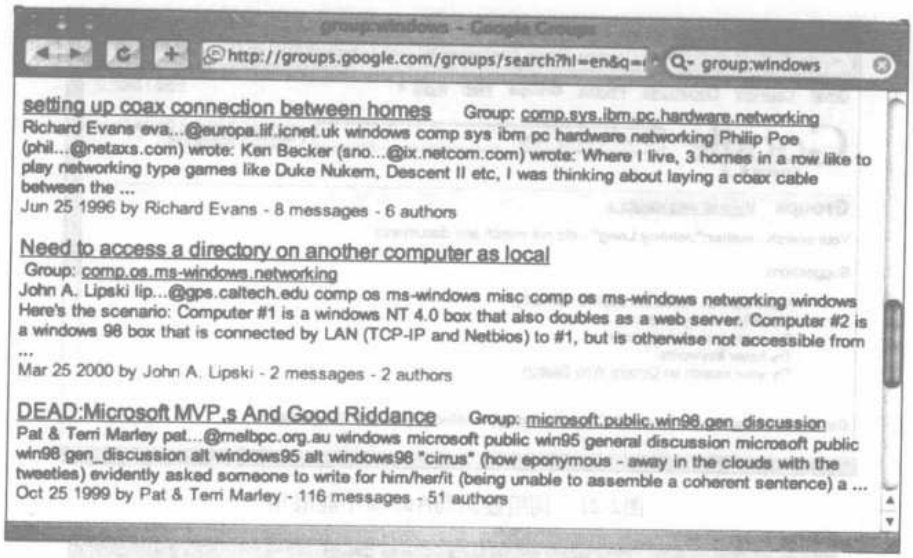


图2-23 Group搜索不仅在Group名称中进行搜索

2.3.15 Insubject：搜索Google Group主题行

insubject操作符实际上和intitle搜索是一样的，它们返回同样的结果。搜索intitle:dragon和insubject:dragon确实返回同样的结果。这很可能是因为一个group帖子的主题也是该帖子的标题。用主题来描述一个消息的标题更为精确，而且这个操作符的存在更有利于在实质上从“deja/USENET 搜索”转移到Google搜索。

和intitle操作符一样，insubject也可以和其他操作符及搜索关键字同时使用。

2.3.16 Msgid：通过消息ID来查找Group帖子

在本书的前一版中，我提到了msgid操作符，该操作符在Google Groups中显示一条特殊的消息。该操作符只有一个参数，即group消息标识。消息标识（或消息ID）是一个新闻组帖子的唯一标识字符串。其格式类似于xxx@yyy.com。自那一版图书发行后，事情有了一些改变，现在msgid几乎停用了，取而代之的是as_msgid搜索URL参数，目前可以通过高级组页面http://groups.google.com/advanced_search来使用。然而，我们仍将在此讨论消息ID以便向你清晰地讲解该功能的工作原理，以防msgid参数又恢复使用。

为了认清楚消息ID，必须先了解原始的group帖子格式。在浏览帖子时，只需要点击原始格式链接即可，如图2-24所示。它将转到一个列出了整个帖子内容的文本页面，如图2-25所示。

该消息的消息ID（IUpug.102004$wl.92198@text.news.blueyonder.co.uk）可以与as_msgid URL参数一起（或者当msgid操作符恢复使用时与msgid操作符一起）用于高级搜索形式。

msgid操作符不能和其他的操作符或者搜索关键字混合使用。

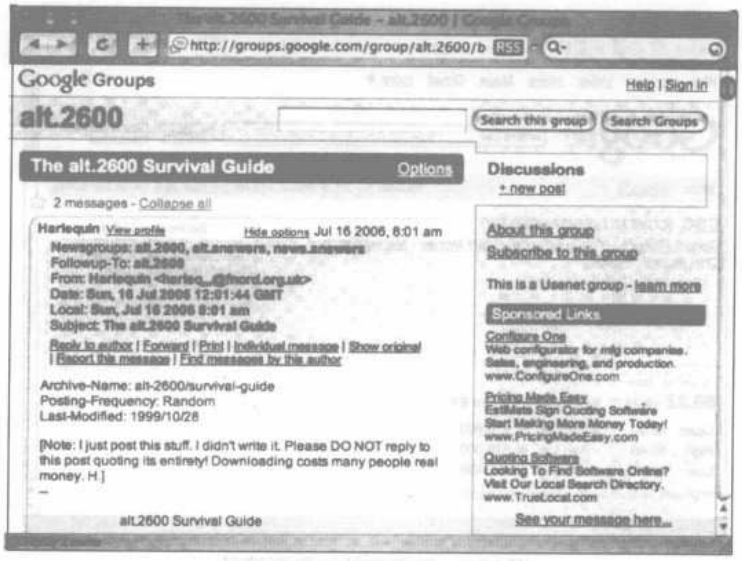


图2-24 典型的Group消息

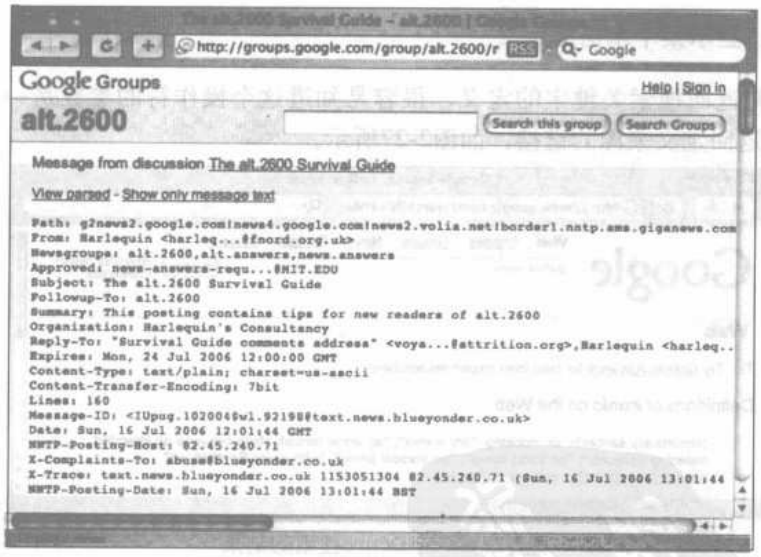


图2-25 帖子的消息ID只能在帖子的原始格式下才可以看到

2.3.17 Stocks：搜索股票信息

stocks操作符可以用来搜索关于某个特定公司的股市信息。这个操作符的参数必须是一个有效的股票简称（Stock Abbreviation）。如果你没有提供一个有效的证券报价机符号，Google会转到一个能够进一步搜索正确的证券报价机符号的页面，如图2-26所示。

stocks操作符不能和其他的操作符及搜索关键字同时使用。

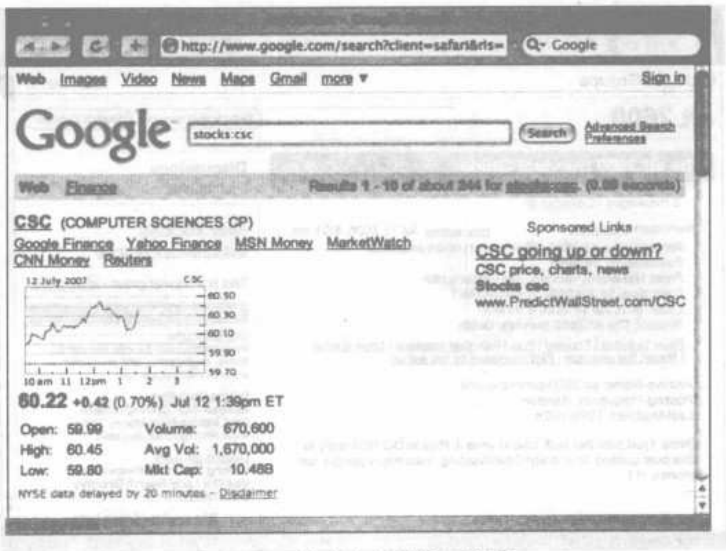


图2-26 搜索有效的股票符号

2.3.18 Define：显示某个术语的定义

define操作符返回搜索关键字的定义。很容易知道这个操作符的参数是一个单词或词组。搜索结果还可以返回定义来源的链接，如图2-27所示。

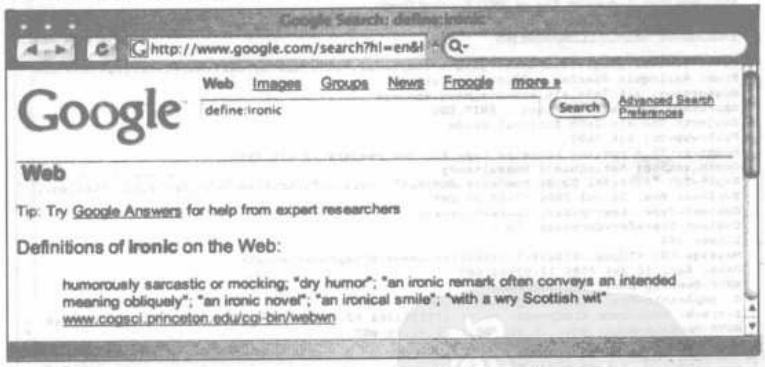


图2-27 一个Define搜索的结果

define操作符不能和其他操作符或搜索关键字混合使用。

2.3.19 Phonebook：搜索电话列表

phonebook操作符可以用来搜索商业和住宅电话列表。有三个操作符可以用来操作电话号码簿：rphonebook、bphonebook和phonebook，分别搜索住宅电话列表，商业电话列表，商业和住宅电话列表。这三个操作符的参数是一样的，通常由一系列描述列表和地理位置的单词组成。大多数情况下，这个操作符的功能和allintitle搜索有些相似，因为在操作符后面的每个单词都包

含在操作符搜索之中。例如phonebook:john darling ny会列出John Darling在New York的商业与住宅电话列表。如图2-28所示，它还提供了流行的地图网站的链接以查看地址或者位置。

图2-28 Phonebook查询的输出

要获取商业电话列表，可以试着使用bphonebook操作符。该操作符并不总能达到预期的工作效果，不过某些特定的查询（例如，如图2-29所示的bphonebook:korean food washington）的效果相当之好，它会找到与此描述相匹配的Google本地商业电话列表并显示出来。

图2-29 Google的商业电话列表操作：bphonebook

还有一些不需要使用phonebook操作符来获取信息的方式。如果你诸如地址（包括州名）或者姓名以及州名作为一个标准查询的话，Google将会在姓名和街道匹配的情况下列出地址列

表或者电话列表，返回一个允许你在地图上查看该处所的链接。

Google搜索背景知识

嘿，快放我出去！

你不用担心Google数据库中的你的信息会被其他人看到。Google允许你删除这些信息以防止别人通过Google访问这些信息。通常，你只需在www.google.com/help/pbremoval.html页面填写一些信息，你的信息就会在48小时内删除。但是这并不能把你的信息从Internet中删除（如果你发现了这样的链接，请告诉我们），而且这个页面只是给你提供了一些能够显示相似信息的地方。这时，Google会相信你不是利用这个表单来删除其他人的信息。

phonebook操作符不能提供非常丰富的错误提示消息，而且很难指出你是否使用了不正确的语法。让我们来看一个查询phonebook:john smith。这个查询不返回任何结果，而且结果页面看起来非常像是标准的“没有结果的”页面，如图2-30所示。

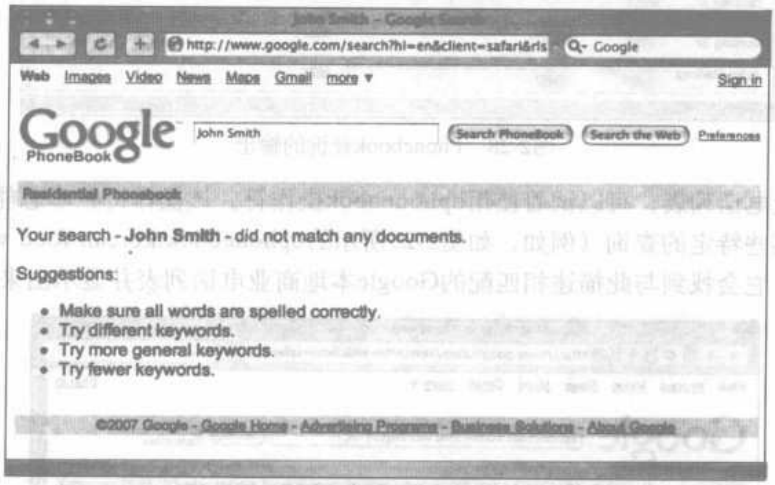


图2-30 容易引起误解的Phonebook错误消息

更糟糕的是，Google给出的纠正这一查询的建议全都是错误的。对这个例子，Googel建议，你需要在查询提供更多的信息才能得到结果，而不是如此少的关键词。例如phonebook:john smith ny，会返回大约600条结果。

2.4 操作符冲突与糟糕的Search-Fu

在你开始使用高级操作符之后，你会意识到它们的一些组合比其他组合能更好地查找到你需要的东西。同样，很快你也将意识到有些操作符根本就不能和其他操作符混合使用。表2-3列出了可以混合使用的操作符。列为“否”的操作符不能和其他操作符在同一个查询中使用。再者，这些操作符有时会产生奇怪的结果，尤其是在没有正确地使用它们相应的语法的时候，所以当这种情况发生时，不要过于惊讶。

表中也列出了只能用于特定Google搜索区域的操作符，以及不能单独使用的操作符。表中

的值需要做些解释。如果标记为“是”则意味着该操作符可以在相应的环境中应用；如果标记为“否”则表示该操作符不能用于该环境并且Google会给出警告信息；如果标记为“不确定(歧义)”则表示在用于相应的环境时，Google会试图把你的查询翻译成其他的意思。真正的Google黑客喜欢探索那些灰色地带，就像这些标记标记为“不确定”的环境。

表2-3 操作符的混合使用

操作符	能否和其他操作符混合	可否单独使用	能否用于搜索网页	能否用于搜索图片	能否用于搜索Groups	能否用于搜索新闻
intitle	是	是	是	是	是	是
allintitle	否	是	是	是	是	是
inurl	是	是	是	是	不确定	同intitle
allinurl	否	是	是	是	是	同intitle
filetype	是	否	是	是	否	不确定
allintext	不确定	是	是	是	是	是
site	是	是	是	是	否	不确定
link	否	是	是	否	否	不确定
inanchor	是	是	是	是	不确定	是
numrange	是	是	是	否	否	不确定
daterange	是	否	是	不确定	不确定	不确定
cache	否	是	是	否	不确定	不确定
info	否	是	是	不确定	不确定	不确定
related	否	是	是	否	否	不确定
phonebook、rphonebook、bphonebook	否	是	是	否	否	不确定
author	是	是	否	否	是	不确定
group	不确定	是	否	否	是	不确定
insubject	是	是	同intitle	同intitle	是	同intitle
msgid	否	是	不确定	不确定	是	不确定
stocks	否	是	否	否	否	同intitle
define	否	是	是	不确定	不确定	不确定

当Allintext和其他操作混用时会产生各种混乱的结果。例如，allintex:moo goo gai filetype:pdf可以很好地查找中国菜的食谱，而allintext:Sum Dum Goy intitle:Dragon没有任何结果——它显然丢掉了1985年的经典影片The Last Dragon（如图2-31所示）。

图2-31 Allinext会产生非常糟糕的结果

尽管一些操作符确实能够和其他操作符组合使用，但是当你把它们混合使用时，仍然有可能无法得到比其他操作符混用更好的结果。在这一节中我们把重点放在指出那些会让你感到头疼的潜在的操作符冲突。我们将从最明显的几个看起。

首先，考虑类似于something -something这样的查询。这种查询不会返回任何结果，而且Google会给出更多的建议。这是一个很显然的例子，不过，可以先来看一下这个查询intitle:something -intitle:something。这个查询和第一个一样，也不返回任何结果，因为我们使用NOT操作符忽略了搜索结果。从字面意义上来讲，我们是说“在标题中查找某些东西并且把标题中包含某些东西的结果隐藏”。这两个例子都很清楚地说明了你不能既查找某样东西又忽略它，因为你得到的结果为空。

但是当同时使用几个高级操作符时，情况就变得不那么明显了。先来看一下site和inurl。URL包含网站的名称。因此，延伸“不要自相矛盾”的规则，不要在site中包含一个关键字同时又在inurl中排除这个关键字，这样才能得到预期的结果，反之亦然。像site:microsoft.com -inurl:microsoft这样的搜索没有任何意义，照理说也无法得到结果，但是出乎意料的是，它能搜索到如图2-32所示的结果。

图2-32 黑客都是违背常规的

当你真要试图跟踪某个主题时，就要牢记这些“规则”，并且要加快步伐朝着你的目标前进。但是，如果为了测试你的Google Hacking技术，则需要打破这些规则！

下面是一些糟糕的搜索以及它们为什么不合理的快速分析。

- site:com site:edu 结果中不可能同时既有edu又包含com。可能你最想搜索的是(site:edu | site:com)，它能够搜索两种域名中的任意一种。
- inanchor:click -click 这是矛盾的。记住，除非你使用高级操作符，你的搜索关键字才能在网页的任何地方出现，包括标题、URL、文本，甚至是锚点。
- allinurl:pdf allintitle:pdf 混合使用以all开头的操作符是最为糟糕的。在你养成使用它们的习惯之前，一定要摈弃这种混用它们的习惯！你需要把allinurl替换为inurl，allintitle替

换为intitle，而且不要使用allintext，它很是令人讨厌。

- site:syngress.com allinanchor:syngress publishing 这个查询没有返回结果，如果从前一个例子以及大多数以all开头的搜索都是不好的事实来看，得到这种结果似乎很合理。然而，这个查询的问题却是出在顺序上，这是一个相当普遍的问题，它会在精确搜索的过程中出现。通过把allinanchor放到查询的开头，即改为allinanchor:syngress publishing site:syngress.com，我们能够得到更多的结果。这看起来似乎不太合乎常理，因为allintitle操作符会把所有跟在它后面的关键字都看作是它的参数，但往往事实就是如此。
- link:www.microsoft.com linux 对于初学者来讲，这是个非常糟糕的搜索，它查找链接到Microsoft并且页面中提到了linux的网站，这看起来似乎是没有问题的。遗憾地是，link是不能和其他操作符混合使用的，但是Google没有给出一个错误的消息，却在帮你把这个查询“修正”之后提供了和“link.www.microsoft.com” linux查询同样的结果。

2.5 总结

Google提供了大量的选项用于执行高级搜索。在第1章中讨论的URL修改能够提供许多选项来修改上一次提交的查询，但是最好在搜索中使用高级操作符。高级操作符比URL修饰符更容易记忆，它也正是任何一个Google黑客的武器库。正因为如此，从考虑保护基于Web的信息方面来讲，它们只应该是好人所使用的工具。

大多数操作符都可以用以混合使用，但最值得注意的例外是allintitle、allinurl、allinanchor和allintext操作符。高级Google搜索人员会试图避免使用这些操作符，而分别使用intitle、inurl和link操作符在标题、URL或者网页的链接中搜索字符串。用于在文档的文本中查找所有提供的关键字的操作符Allintext，是最少使用的，而且也是高级操作符当中最为多余的操作符。Filetype和site是非常强大的用于搜索特定类型的文件或者特定网页的操作符。daterange操作符可以用来搜索在某一特定时间范围内索引的文件。在Google抓取网页时，它会同时生成某些特定的信息，如一个页面的缓存副本，关于该页面的信息摘要以及可能与之相关的网站列表。这些信息可以分别通过cache、info以及related操作符来获得。如果要搜索Google Groups文档的作者，需要使用author操作符。phonebook系列操作符可以返回商业或者住宅电话列表以及特定地址的地图。stocks操作符返回关于某个特定的证券报价机符号的股票信息，而define操作符可以返回一个单词或者一个简单的词组的定义。

2.6 快速查找解决方案

Intitle

- 在页面标题中查找字符串。
- 能够很好地和其他操作符混合使用。
- 擅长搜索网页、Group、图片和新闻。

Allintitle

- 在页面标题中查找所有的关键字。

- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页、Group、图片和新闻。

Inurl

- 在页面的URL中查找字符串。
- 能够很好地和其他操作符混合使用。
- 擅长搜索网页和图片。

Allinurl

- 在页面的URL中查找所有的关键字。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页、Group和图片。

Filetype

- 根据文件扩展名查找特定类型的文件。
- 等同于ext。
- 需要附加搜索关键字。
- 能够很好地和其他操作符混合使用。
- 擅长搜索网页和Group。

Allintext

- 在页面的文本中查找所有提交的关键字。
- 相当讨厌——不要用它。
- 就当你从没听说过allintext。

Site

- 限定在某个特定的网站或者域搜索。
- 能够很好地和其他操作符混合使用。
- 可以单独使用。
- 擅长搜索网页、Groups以及图片。

Link

- 搜索链接到一个网站或者URL的链接。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Inanchor

- 在链接的描述文本中查找文本。
- 能够很好地和其他操作符和搜索关键字混合使用。
- 擅长搜索网页、图片和新闻。

Daterange

- 查找在某一特定日期范围内索引的页面。
- 需要搜索关键字。
- 能够很好地和其他操作符或搜索关键字混合使用。

- 擅长搜索网页。
- 可能会逐步淘汰并为as_qdr所取代。

Numrange

- 在某一特定范围内查找数字。
- 能够很好地和其他操作符或搜索关键字混合使用。
- 擅长搜索网页。
- 与ext同效。

Cache

- 显示页面的缓存版本。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Info

- 显示页面的摘要信息。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Related

- 显示与“提交的网站或URL相关的”站点。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Phonebook，Rphonebook，Bphonebook

- 显示住宅或者商业电话列表。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Author

1. 搜索Group帖子的作者。
2. 能够很好地和其他操作符或搜索关键字混合使用。
3. 擅长搜索Group。

Group

- 搜索Group名称，选择特定的Group。
- 能够很好地和其他操作符或搜索关键字混合使用。
- 擅长搜索Group。

Insubject

- 在Group帖子的主题中查找字符串。
- 能够很好地和其他操作符或搜索关键字混合使用。
- 擅长搜索Group。

Msgid

- 通过消息ID来查找Group消息。

- 不能很好地和其他操作符或搜索关键字混合使用。
- 擅长搜索Group。
- 极其古怪。请在groups.google.com/advanced_search中使用高级搜索形式来替代之。

Stocks

- 针对某个证券报价机显示出Yahoo Finance（Yahoo财经）的股票列表。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

Define

- 显示对提供的单词或者词组的各种定义。
- 不能很好地和其他操作符或者关键字混合使用。
- 擅长搜索网页。

2.7 网站链接

- Google文件类型FAQ，www.google.com/help/faq_filetypes.html。
- 文件扩展名信息资源，www.filext.com。这个网站能够帮助你找到某个特定的扩展名与何种程序相关联。
- http://searchenginewatch.com/searchday/article.php/2160061。这篇文章讨论了和Google日期限定搜索选项有关的话题。
- 非常棒的在线Julian日期转换工具，www.24hourtranslations.co.uk/dates.htm和www.tesre.bo.cnr.it/~mauro/JD/

2.8 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：其他的搜索引擎也提供高级操作符吗？它们的高级操作符和Google的比起来如何？

答：是的，大部分搜索引擎都提供类似的操作符。其中Yahoo和Google最像。这可能是因为Yahoo曾经主要依靠Google作为其搜索提供商。Yahoo的操作符包括site（域名搜索），hostname（完整的服务器名），link，url（只显示一个文档），inurl和intitle。Yahoo高级搜索页面提供了其他的选项和URL修饰符。你可以借助于http://search.yahoo.com/search/options中的HTML表单来仔细研究那些有趣的选项。这个搜索页面非常相似于Google的高级搜索页面。

AltaVista提供domain，host，link，title和url操作符。可以在www.altavista.com/web/adv找到这个高级搜索页面。最为有趣的是imeframe搜索，它比Google的as_qdr URL修饰符更为细致，不仅能够按时间范围搜索，还可以指定时间段搜索，例如过去的一周，两周或更长。

问：哪里可以获得所有高级操作符的概要信息？

答：浏览www.google.com/help/operators.html。这个页面描述了各种操作符，也是对本章

的一个很好的总结。新的操作符会在这个页面上公布，但是有的操作符在正式公开之前会进行beta测试。有时候这些操作符是在由于Google用户使用了太多的冒号而在偶然情况下发现的。谁知道呢，说不定下一个发现最新的隐藏操作符的人就是你！

问：当新的操作符出来时我怎样才能及时地了解它？其他的Google相关的新闻和提示呢？

答：我们可以在许多网站上浏览关于Google的新闻和信息。首先是http://googleblog.blogspot.com，它是Google的官方Weblog。尽管它本质上不是讨论技术的，但是还是可能通过它来了解Google内部正在发生的事情。另外一个是Aaron Swartz的非官方Google Blog，http://google.blogspace.com/。虽然这个网站没有t得到Google的赞同或支持，但是人们会经常提到它，而且有些东西是很有深度的。第三个必须收藏的网站是位于http://labs.google.com的Google实验室。这是了解Google所提供的最新特性和功能的最好地方。同时，还可以浏览Goolge Alerts（Goolge通知）主页www.google.com/alerts来获得最新的Google查询更新，即使有些和Google关系不大。当有某项搜索关键字更新时，Google通知会发送一封E-mail给你。你可以使用这个工具，通过某项搜索关键字的通知，例如google高级操作符site:google.com来发现新的操作符。最后至少要在www.google.com/trends and Google Zeitgeist（www.google.com/press/zeitgeist.html）查看Google Trends（Google趋势）以留意其他人在搜索什么。你可能恰好看到了少许漂流在外的Google黑客。

问：查询中的单词顺序重要吗？

答：有些时候很重要，有时则不然。如果你对某个网站的排名比较感兴趣，尤其是那些显示在前几页的网站，顺序就很重要了。Google会取出查询中相邻的两个单词，然后查找网站中是否含有按照你所指定顺序的单词。不管你的查询中的关键字以何种顺序排列，Google都会返回同样的站点（除非你用引号把单词引起来，强制Google按照该顺序去搜索）。可以做一些简单的查询实验来了解这是如何工作的，如food clothes与clothes food。

问：你是否能给我提供一些更酷的操作符？

答：这个列表永无止境。要想跟上Google的步伐很难。好的。这个怎么样：view。在Web查询的结尾放上view:map或者view:timeline，以便在地图视图或者一个酷酷的时间轴视图中查看结果。要想执行一个有教育意义的查询，不妨来尝试一下“Abraham Lincoln” view:timeline。要找出世界上所有的黑客所在，可以尝试一下这个搜索hackers view:map。如果想看一下喇叭裤是否真的又再次流行了，可以尝试一下这个搜索“bell bottoms” view:timeline。这有点捣乱。

第3章 Google Hacking基础

3.1 简介

本书的大部分内容描述的都是那些“坏家伙”所使用的技术，这些技术可以用来查找敏感的信息。我们提到这个信息主要是为了帮助你了解他们的动机，以便让你更好地保护自己和客户。我们已经学习了一些基本的搜索技巧，这是任何一个想克服基础知识的限制而突破到下一个级别——Google黑客之路的Google用户所必须掌握的。现在我们先来学习最为基本的技术，然后再进行更为深入的钻研。

首先，我们将讨论Google的缓存。如果你还没有用缓存做过实验，那你就已经落后一步了。我们建议你在继续读下去之前，至少曾在Google的搜索结果页面上点击过各种缓存链接。任何乐于助人的Google黑客都会告诉你，在浏览页面的缓存版本时，存在某种匿名隐藏的特征。只有在这种环境下才具有匿名特征，而且这种匿名是有其局限性的。但是，Google却能够非常好地掩饰你的浏览行为，即当你浏览网站时，目标网站甚至无法得到任何一个来自于你的数据包。我们将告诉你Google是怎样做到这一点的。

接下来，我们将讨论目录列表。这些界面不太美观的网页全都是有用的信息，而且其中有些信息是我们在后面章节中将用到的高级攻击搜索的基础。

最后，我们将了解一种称为遍历（traversing）的技术，即试图收集更多信息的搜索扩展技术。我们将学习目录遍历、数字范围扩展以及扩展名搜索，这些都是任何正义的黑客以及好人应该了解并加以防范的技术。

3.2 使用缓存进行匿名浏览

Google的缓存功能是件让人感到非常惊奇的事情。一个最简单的事实是如果Google曾经抓取了某个网页或者文档，即使源文件现在已经不存在或者更新了，那么你仍然很可能获得它的一个副本。当然它同样也有消极的一面，比如即使你已经从网站服务器上把敏感数据删除了，黑客仍然可以通过Google来得到这些数据的拷贝。缓存的另外一个不好的地方是那些坏家伙甚至不需要给服务器发送任何一个单独的数据包就能够抓取你的整个网站（包括网站中你已经“忘记”的区域）。如果你的网站服务器没有得到这些数据包，那么就无法在日志文件中记录。（你经常记录你的网络连接，对么？）如果在日志文件中没有记录，那么你也许就不会知道你的敏感数据已经被窃取了。每天都想着有成千上万兆字节的敏感数据从网站服务器中泄露出去是一件很让人发愁的事。了解黑客是如何通过Google的缓存对敏感数据实施匿名攻击是十分重要的。

Google会保存大部分它所抓取到的网页数据。虽然并不都是如此而且这种情况也是可以防

止的，但是大多数Google抓取的数据都被复制了一份，而且可以通过搜索结果页面的缓存链接来访问。我们需要仔细研究Google的缓存文档标题中的微妙之处。图3-1所示的标题收集于www.phrack.org。

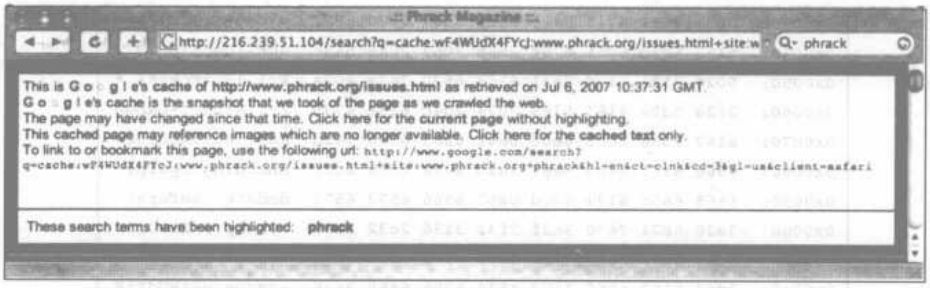


图3-1 缓存的标题包含一个关于图片的细心的警告

如果你已经对缓存标题相当熟悉并且打算一略而过，那么请你稍微放慢一点并认真地看一看这个标题。图3-1中所示的缓存标题说“本缓存网页可能引用了已经不存在的图片”。通常很容易忽略这个消息，但它却提供了一条关于Google在幕后都做了什么的重要线索。

为了便于理解，我们来看一看当我们浏览这个缓存网页时，tcpdump所收集到的部分数据。可以使用tcpdump -n命令来捕捉这些数据。由于tcpdump的安装或者实现的不同，可能还需要使用-i选项来建立一个侦听接口。tcpdump命令的输出如图3-2所示。

```
10.0.1.6.49847 > 200.199.20.162.80:
10.0.1.6.49848 > 200.199.20.162.80:
200.199.20.162.80 > 10.0.1.6.49847:
10.0.1.6.49847 > 200.199.20.162.80:
200.199.20.162.80 > 10.0.1.6.49848:
10.0.1.6.49848 > 200.199.20.162.80:
10.0.1.6.49847 > 200.199.20.162.80:
10.0.1.6.49848 > 200.199.20.162.80:
66.249.83.83.80 > 10.0.1.3.58785:
66.249.83.83.80 > 10.0.1.3.58790:
66.249.83.83.80 > 10.0.1.3.58790:
66.249.83.83.80 > 10.0.1.3.58790:
66.249.83.83.80 > 10.0.1.3.58790:
66.249.83.83.80 > 10.0.1.3.58790:
```

图3-2 在浏览一个缓存网页时tcpdump所收集到的输出

我们来逐步分析这个输出，先从底部开始。这是网页浏览器（10.0.1.6）和Google服务器（66.249.83.83）间的80端口的对话。这正是我们所期望的与Google互联时的流量类型，但是这次捕捉到的信息的开头显示了200.199.20.162相连的是另一个80端口（Web）连接。这并不是一个Google的服务器地址，该IP的nslookup显示它是www.phrack.org Web服务器。与该服务器的连接可以通过重新运行tcpdump来解释——这些tcpdump带有更多明确设计来显示这些数据包的几百个数据字节以及头部里的选项。如图3-3所示的部分捕捉数据是在运行以下语句并且在按住Shift的同时重新加载该缓存页面之时捕捉而得。

```
tcpdump -Xx -s 500 -Cn
```

按住Shift的同时重新加载会迫使大多数的浏览器与Web主机再次连接，而不是依赖浏览器可能用到的任何缓存。

```
0x0030:  085c 0661 4745 5420 2f69 6d67 2f70 6872  .\.aGET./img/phr
0x0040:  6163 6b2d 6c6f 676f 2e6a 7067 2048 5454  ack-logo.jpg.HTT
0x0050:  502f 312e 310d 0a41 6363 6570 743a 202a  P/1.1..Accept:.*
0x0060:  2f2a 0d0a 4163 6365 7074 2d4c 616e 6775  /*..Accept-Langu
0x0070:  6167 653a 2065 6e0d 0a41 6363 6570 742d  age:.en..Accept-
0x0080:  456e 636f 6469 6e67 3a20 677a 6970 2c20  Encoding:.gzip,.
0x0090:  6465 666c 6174 650d 0a52 6566 6572 6572  deflate..Referer
0x00a0:  3a20 6874 7470 3a2f 2f32 3136 2e32 3339  :.http://216.239
0x00b0:  2e35 312e 3130 342f 7365 6172 6368 3f71  .51.104/search?q
0x00c0:  3d63 6163 6865 3a77 4634 5755 6458 3446  =cache:wF4WUdX4F
0x00d0:  5963 4a3a 7777 772e 7068 7261 636b 2e6f  YcJ:www.phrack.o
0x00e0:  7267 2f69 7373 7565 732e 6874 6d6c 2b73  rg/issues.html+s

[...]
0x01b0:  6565 702d 616c 6976 650d 0a48 6f73 743a  eep-alive..Host:
0x01c0:  2077 7777 2e70 6872 6163 6b2e 6f72 670d  .www.phrack.org.
```

图3-3 Host头部域所显示的局部HTTP请求

0x30和0x40行表明我们正在从该服务器下载（通过一个GET请求）一个图片文件——确切地说是一个JPG图片。接下来的那一行显示了Host域，这指出我们正在和www.phrack.org网站服务器进行会话。根据这个Host头部以及这个数据包是发送到IP地址200.199.20.162的事实来看，我们可以大胆地假设Phrack网站服务器的虚拟主机服务器位于该地址。这意味着当我们在浏览Phrack网页的缓存版本时，是直接从Phrack服务器本身下载图片的。当我们正在借助于浏览Google的缓存网页来努力隐藏真实信息时，我们恰恰是在暴露自己！而且，0x90表明REFERER域也被传送给了Phrack服务器，而且这个域包含的URL正是Phrack的网页的Google缓存副本。这意味着不仅我们自己不是匿名的，而且我们的浏览器也告诉了Phrack网站服务器我们正在试图浏览该网页的缓存版本！关于匿名，我们就讲到这里。

大多数真正的黑客在浏览目标网页时都使用代理服务器，即便代理服务器在第一时间就能够知晓他们的Google行为，这也没有什么意义。如果我们使用的是一个匿名的代理服务器来进行测试的话，那么Phrack网站服务器只能得到代理服务器的IP地址，而不是我们的真实IP地址。

Google搜索背景知识

Google黑客的提示

如果你比较重视在线匿名，那么使用代理服务器是个不错的主意。渗透测试人员会使用代理服务器来模拟一个真实的黑客在一次入侵试图中做了哪些事情。查找有效、高质的代理服务器是一项很费力的任务，当然我们可以用一点Google hacking技巧来做这件事！使用Google来查找代理服务器，可以试试下面的查询：

```
inurl:"nph-proxy.cgi" "Start browsing"
```

或者

"cacheserverreport for" "This analysis was produced by calamaris"

这些查询能够查找到可以用于测试目的的在线公开代理服务器。没有什么比使用Google搜索来查找代理服务器更为方便的了！但是，还要记住，还可以通过许多其他方式来获取代理服务器，如网站atomintersoft或者samair.ru代理网站。试着用Google搜索来查找！

不过，缓存标题给出的是一个仅浏览Google所捕捉的数据的选项，而没有任何外部引用。正如图3-1所示的，在标题的头部有一个标题为“Click here for the cached text only”（单击此处，只查看缓存文本）的链接。点击这个链接会产生如图3-4所示的tcpdump输出，此时使用的是tcpdump -n命令。

```
216.239.51.104.80 > 10.0.1.6.49917:
216.239.51.104.80 > 10.0.1.6.49917:
216.239.51.104.80 > 10.0.1.6.49917:
10.0.1.6.49917 > 216.239.51.104.80:
10.0.1.6.49917 > 216.239.51.104.80:
216.239.51.104.80 > 10.0.1.6.49917:
216.239.51.104.80 > 10.0.1.6.49917:
216.239.51.104.80 > 10.0.1.6.49917:
10.0.1.6.49917 > 216.239.51.104.80
```

图3-4 由tcpdump捕捉到的缓存文本

尽管事实上我们加载的仍是同一个页面，但是现在我们只与Google服务器(216.239.51.104）进行连接，而不与其他的外部服务器连接。如果观察缓存网页中的通过点击“cached text only”（只缓存文本）链接产生的URL，便可以发现Google添加了一个有用的参数，&strip=1。这个参数强制Google缓存URL只用来显示缓存文本，而避免任何外部引用。这个URL参数仅能用于引用Google缓存网页的URL。

总结上面的内容来看，我们可以不使用代理服务器就能够达到相当的匿名性，只需要使用一次快速的剪切和粘贴操作以及一个URL修改即可。例如一个Google查询site:phrack.org[⊖]。与点击缓存链接不同的是，我们在缓存链接上右击，然后把该URL复制到剪贴板，如图3-5所示。不同的浏览器对这个操作的处理均不同，所以你需要根据自己的情况去捕捉这个链接的URL。

在把URL复制到剪贴板中之后，把它粘贴到浏览器的地址栏中，然后在URL的后面追加参数&strip=1。此时的URL应该类似于http://216.239.51.104/search?q=cache:LBQZIrSkMgUJ:www.phrack.org/+site:phrack.org&hl=en&ct=clnk&cd=1&gl=us&client=safari&strip=1。[⊖]在修改完URL之后按回车键以加载该页面，浏览器应该转到缓存页面的文本版本，它的标题与缓存页面的标题有些不同，如图3-6所示。

注意到该缓存标题读起来与标准的缓存标题不同。与“This cached page may reference images which are no longer available”（本缓存网页可能引用了已经不存在的图片）不同的是，这一行变成了“Click here for the full cached version with images included”（单击此处，查看完整的缓存网

⊖ 由于phrack.org网站的更新，该查询有可能失败，读者可作适当的调整，如site:www.phrack.org inurl:index。——译者注

⊖ 由于phrack.org网站的更新，该链接有可能失效，读者可自行调整。——译者注

页（包含图片））。这表明当前缓存页面已经剔除了外部引用。遗憾的是，这个页面并不包含图片，所以看起来和原始页面不一样，而且在某些时候根本就看不清楚。如果碰到这种情形，你仍可以使用代理服务器来打开这个页面，但是真正的Google黑客“根本就不需要代理服务器”！

图3-5 利用剪切和粘贴来实现匿名缓存浏览

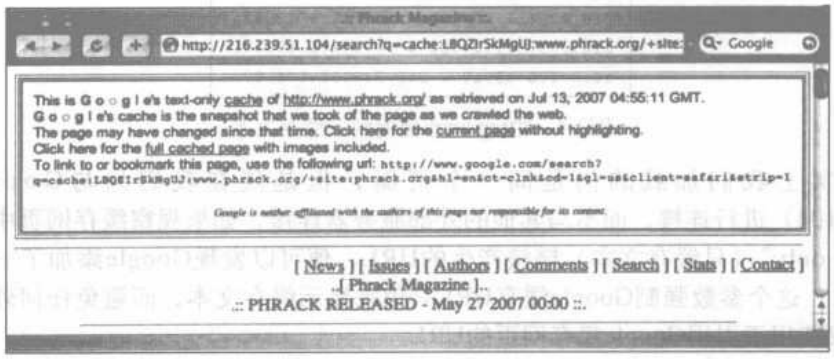


图3-6 清理后的缓存页面标题

Google搜索背景知识

玩转关键字高亮

如果你曾在文档中一页页地翻着来查找某个特定的单词或者词组，那么你也许已经知道Google的缓存版本页面会对搜索关键字进行高亮显示。但是你可能还不知道可以使用Google的高亮工具在缓存网页中高亮显示某些并不包含在原始搜索中的关键字。虽然这样会让URL更为冗长，但其操作是非常显而易见。例如，如果你搜索peeps marshmallows并且浏览第二个缓存网页，那么该缓存网页的URL的一部分类似于www.peepresearch.org/peeps+marshmallows&hl=en。请注意基础URL后面列出的我们使用的搜索关键字。如果想高亮其他的关键字，只需注意基础URL的后部的区域，在本例中是+peeps+marshmallows。此时，只要在这里添加或者删除一些单词，然后按回车，Google就会在浏览器中高亮这些关键字！例如，要在高亮单词表中添加fear和risk，只需在URL中添加这两个单词即可，添加后的URL如下：www.peepresearch.org/+fear+risk+peeps+ marshmallows&hl=en。你知道Marshmallow Peeps是否真的害怕？不相信我？那就问Google好了。

3.3 目录列表

目录列表是一种能够列出网站服务器上存在的文件和目录的网页类型。目录列表可以通过直接点击目录链接来浏览，它一般包含有一个描述当前目录的标题，文件和目录列表都可以直接点击。而且，通常脚注会用来标注目录列表的底部。这些元素均可以参见如图3-7所示的目录列表样例。

图3-7 一个包含了常见元素的目录列表

目录列表非常类似于FTP服务器，针对“按照分类文件夹存储的文件”的授权访问，它提供了一种简单易用的解决方案。遗憾的是，目录列表本身具有许多缺点，尤其是：

- 它们自身并不安全。它们不能阻止某个用户下载某个特定的文件或者访问某个特定的目录。这种安全任务通常是留给网站服务器软件内置的保护措施、第三方脚本、模块或者专门针对这个目的所设计的程序来解决的。
- 它们显示出来的信息能够帮助攻击者了解网站服务器的某些特定技术细节。
- 它们无法区分公开的文件和那些私有的文件。
- 它们通常都是偶然显示出来的，这是因为许多网站服务器都是在顶层索引文件（index.htm、index.html、default.asp等）丢失或者无效的时候才会显示目录列表。

所有这些缺点组合起来就是致命的弱点。

在这一节中，我们将讨论Google黑客如何利用目录列表的。

3.3.1 查找目录列表

攻击者利用目录列表的最容易的方法首先是找到它！由于目录列表提供了“parent directory”（父目录）的链接并允许通览所有的文件和文件夹，所以即便是最初级的攻击者也能很快地发现，只要简单地查找并浏览这些列表就可以找到一些敏感的数据。

利用Google来查找目录列表相当容易。图3-7表明大多数目录列表都是以词组“Index of”开始的，而且其标题中也包含这个词组。一个很明显得查找这种类型网页的查询为intitle:index.of，它能够查找到标题中含有关键字index of的文档。其中的句点（“.”）在Google中是一个单字符通配符。遗憾的是，这个查询会返回大量我们不想要的结果，例如那些含有如

下几种标题的网页：

```
Index of Native American Resources on the Internet
LibDex - Worldwide index of library catalogues
Iowa State Entomology Index of Internet Resources
```

从这些文档的标题来判断，显然它们不仅不是我们要查找的网页，而且也不是我们想要查找的目录列表类型。Ben Kenobi可能会说："这并不是你所查找的目录列表"。有几种完善后的查询能够提供更为精确的结果，例如intitle:index.of "parent directory"（如图3-8所示）或者intitle:index.of name size。这些查询之所以确实能提供目录列表，不仅仅是通过关注标题中的index.of，而且还通过关注常出现在目录列表内的关键字，例如parent directory、name和size。即便是从搜索结果页面的总结来判断，你也能看出这些结果就是我们所要查找的目录列表类型。

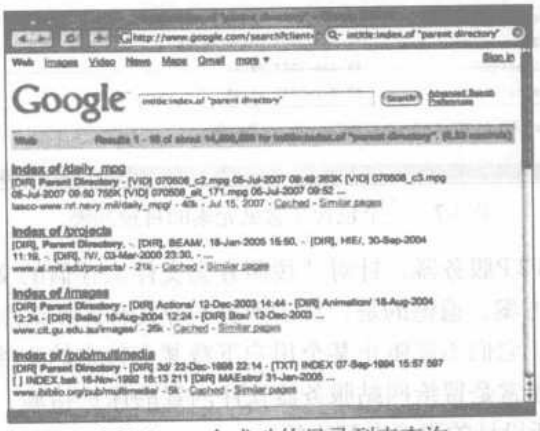


图3-8 一个成功的目录列表查询

3.3.2 查找特定的目录

在某些情况下，不但需要查找目录列表，而且需要查找能够访问某个特定目录的目录列表，这可能会更有用处。这可以通过在查询中添加目录的名称来完成。为了查找能够通过目录列表访问的"admin"目录，可以使用查询intitle:index.of.admin或者intitle:index.of inurl:admin，如图3-9所示。

图3-9 在目录列表中查找特定的目录

3.3.3 查找特定的文件

基于这些类型的页面已经列出了文件的名称和目录，所以我们可以在目录列表中查找某些特定的文件。例如，如果想查找WS_FTP的日志文件，不妨试一试查询intitle:index.of ws_ftp.log，如图3-10所示。这种技术可以进一步扩展为查找任意文件，只需要在标题中查找index.of，并在网页的文本中查找文件名即可。

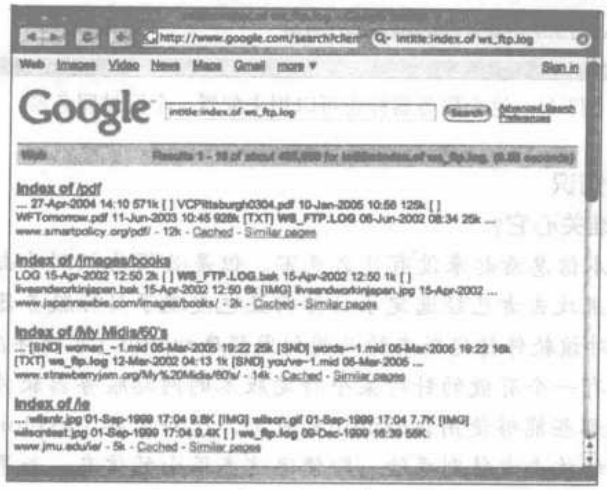


图3-10 在目录列表中查找文件

你也可以使用filetype和inurl来查找特定的文件。再次查询ws_ftp.log文件时，不妨试一试查询filetype:log inurl:ws_ftp.log。这种技术通常能够比限定在index.of的搜索查找到更多的结果。在本书中，我们还将对特定文件搜索做更深入的研究。

3.3.4 服务器的版本

网站服务器的确切的软件版本信息可以帮助攻击者决定使用何种方法来攻击服务器。攻击者可以通过直接连接到服务器的Web端口，并且发送一个HTTP（Web）头部请求来得到这一信息。但是，还有一种利用Google来得到类似信息的方法，使用这个方法不需要直接连接到目标服务器。一种方法是使用目录列表中所提供的信息。

图3-11给出了一个典型的目录列表的底部。我们可以注意到有些目录列表会提供服务器软件的名称及其版本号。精于此道的Web管理员可能会伪造这些服务器标志（server tag），但是大多数情况下这些信息都是正常的，而且确实是攻击者用来优化对服务器的攻击所需要的信息类型。

这种用于搜索服务器的Google查询方法是一种intitle:index.of查询的简单拓展。图3-11所示的列表是使用intitle:index.of “server at”查询搜索到的。这个查询会查找网络上所有标题中包含index of并且页面内的文本包含server at的目录列表。这也许看起来并不是一个非常特殊的搜索，但是其结果非常简洁，而且不需要额外的优化。

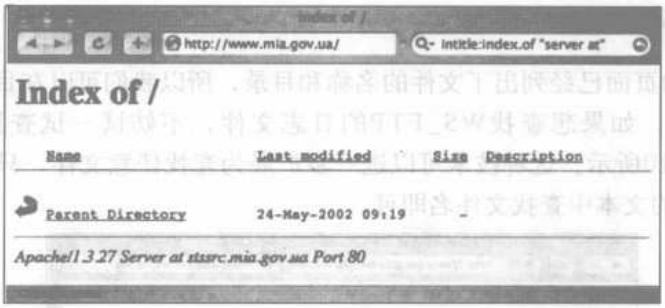


图3-11 这个服务器标志可以用来配置一个网站服务器

Google搜索背景知识

服务器的版本？谁关心它？

尽管服务器的版本信息看起来没有什么危害，但是必须意识到攻击者可以有两种方式来利用这类信息。如果攻击者已经选定了目标而且也发现了目标服务器的版本信息，那么他可能就会去搜索针对该软件特定版本的漏洞利用程序exploit（可能存在，也可能不存在）。反之，若攻击者手上有一个有效的针对某个特定版本的网站服务器软件的exploit，他可能就会用Google来搜索那些能够使用该exploit的目标。这种装备了exploit，并且四处寻找具有潜在风险的服务器的攻击者特别危险。即使像这类很小的信息，如果一旦泄露，都会让那些聪明的攻击者获得很重要的信息。

为了查找某个特定版本的服务器，可以把intitle:index.of查询进一步拓展为intitle:index.of "Apache/1.3.27 Server at"。这个查询能够查到类似于图3-11中列出的那样的页面。见表3-1，许多不同的服务器都可以通过目录列表来区分。

表3-1 可以通过目录列表查找到的某些特定的服务器

网站服务器的目录列表
"AnWeb/1.42h" intitle:index.of
"Apache Tomcat/" intitle:index.of
"Apache-AdvancedExtranetServer/" intitle:index.of
"Apache/df-exts" intitle:index.of
"Apache/" intitle:index.of
"Apache/AmEuro" intitle:index.of
"Apache/Blast" intitle:index.of
"Apache/WWW" intitle:index.of
"Apache/df-exts" intitle:index.of
"CERN httpd 3.0B (VAX VMS)" intitle:index.of
"CompySings/2.0.40" intitle:index.of
"Davepache/2.02.003 (Unix)" intitle:index.of
"DinaHTTPd Server/1.15" intitle:index.of
"HP Apache-based Web "Server/1.3.26" intitle:index.of
"HP Apache-based Web "Server/1.3.27 (Unix) mod_ssl/2.8.11 OpenSSL/0.9.6g"

（续）

网站服务器的目录列表
intitle:index.of
“HP-UX_Apache-based_Web_Server/2.0.43” intitle:index.of
“httpd+ssl/kttd” * server at intitle:index.of
“IBM_HTTP_Server” intitle:index.of
“IBM_HTTP_Server/2.0.42” intitle:index.of
“JRun Web Server” intitle:index.of
“LiteSpeed Web” intitle:index.of
“MCWeb” intitle:index.of
“MaXX/3.1” intitle:index.of
“Microsoft-IIS/* server at” intitle:index.of
“Microsoft-IIS/4.0” intitle:index.of
“Microsoft-IIS/5.0 server at” intitle:index.of
“Microsoft-IIS/6.0” intitle:index.of
“OmniHTTPd/2.10” intitle:index.of
“OpenSA/1.0.4” intitle:index.of
“OpenSSL/0.9.7d” intitle:index.of
“Oracle HTTP Server/1.3.22” intitle:index.of
“Oracle-HTTP-Server/1.3.28” intitle:index.of
“Oracle-HTTP-Server” intitle:index.of
“Oracle HTTP Server Powered by Apache” intitle:index.of
“Patchy/1.3.31” intitle:index.of
“Red Hat Secure/2.0” intitle:index.of
“Red Hat Secure/3.0 server at” intitle:index.of
“Savant/3.1” intitle:index.of
“SEDWebserver *” “server at” intitle:index.of
“SEDWebserver/1.3.26” intitle:index.of
“TcNet httpsrv 1.0.10” intitle:index.of
“WebServer/1.3.26” intitle:index.of
“WebTopia/2.1.1a “ intitle:index.of
“Yaws 1.65” intitle:index.of
“Zeus/4.3” intitle:index.of

表3-2 Apache版本的目录列表

能够通过目录列表来查找到Apache版本的查询
“Apache/1.0” intitle:index.of
“Apache/1.1” intitle:index.of
“Apache/1.2” intitle:index.of
“Apache/1.2.0 server at” intitle:index.of
“Apache/1.2.4 server at” intitle:index.of
“Apache/1.2.6 server at” intitle:index.of
“Apache/1.3.0 server at” intitle:index.of
“Apache/1.3.2 server at” intitle:index.of

（续）

能够通过目录列表来查找到Apache版本的查询
“Apache/1.3.1 server at” intitle:index.of
“Apache/1.3.1.1 server at” intitle:index.of
“Apache/1.3.3 server at” intitle:index.of
“Apache/1.3.4 server at” intitle:index.of
“Apache/1.3.6 server at” intitle:index.of
“Apache/1.3.9 server at” intitle:index.of
“Apache/1.3.11 server at” intitle:index.of
“Apache/1.3.12 server at” intitle:index.of
“Apache/1.3.14 server at” intitle:index.of
“Apache/1.3.17 server at” intitle:index.of
“Apache/1.3.19 server at” intitle:index.of
“Apache/1.3.20 server at” intitle:index.of
“Apache/1.3.22 server at” intitle:index.of
“Apache/1.3.23 server at” intitle:index.of
“Apache/1.3.24 server at” intitle:index.of
“Apache/1.3.26 server at” intitle:index.of
“Apache/1.3.27 server at” intitle:index.of
“Apache/1.3.27-fil” intitle:index.of
“Apache/1.3.28 server at” intitle:index.of
“Apache/1.3.29 server at” intitle:index.of
“Apache/1.3.31 server at” intitle:index.of
“Apache/1.3.33 server at” intitle:index.of
“Apache/1.3.34 server at” intitle:index.of
“Apache/1.3.35 server at” intitle:index.of
“Apache/2.0 server at” intitle:index.of
“Apache/2.0.32 server at” intitle:index.of
“Apache/2.0.35 server at” intitle:index.of
“Apache/2.0.36 server at” intitle:index.of
“Apache/2.0.39 server at” intitle:index.of
“Apache/2.0.40 server at” intitle:index.of
“Apache/2.0.42 server at” intitle:index.of
“Apache/2.0.43 server at” intitle:index.of
“Apache/2.0.44 server at” intitle:index.of
“Apache/2.0.45 server at” intitle:index.of
“Apache/2.0.46 server at” intitle:index.of
“Apache/2.0.47 server at” intitle:index.of
“Apache/2.0.48 server at” intitle:index.of
“Apache/2.0.49 server at” intitle:index.of
“Apache/2.0.49a server at” intitle:index.of
“Apache/2.0.50 server at” intitle:index.of
“Apache/2.0.51 server at” intitle:index.of
“Apache/2.0.52 server at” intitle:index.of
“Apache/2.0.55 server at” intitle:index.of
“Apache/2.0.59 server at” intitle:index.of

目录列表除了能够确定Web服务器的版本，而且还能够判定服务器的操作系统（同时还有模块以及其他安装的软件）。我们将在后面讲解能够完成这一目的的更为特殊的技术，但是刚刚看到的查找服务器版本的技术可以通过在查询中加入更多的细节来做进一步的拓展。表3-3列出了能够查找到极为隐蔽的服务器软件组合信息的查询，这些信息都是由服务器标志透露出来的。这些标志能够给出相应服务器的大量信息，而且它们也能很好地证明即使一个看起来很小的信息泄露有时就能够让整个事态失去控制，而透露出比想像中更多的信息。

表3-3 搜索特定且隐蔽的服务器版本

搜索特定且隐蔽的服务器版本的查询
"Apache/1.3.12 (Unix) mod_fastcgi/2.2.12 mod_dyntag/1.0 mod_advert/1.12 mod_czech/3.1.1b2" intitle:index.of
"Apache/1.3.12 (Unix) mod_fastcgi/2.2.4 secured_by_Raven/1.5.0" intitle:index.of
"Apache/1.3.12 (Unix) mod_ssl/2.6.6 OpenSSL/0.9.5a" intitle:index.of
"Apache/1.3.12 Cobalt (Unix) Resin/2.0.5 StoreSense-Bridge/1.3 ApacheJServ/1.1.1 mod_ssl/2.6.4 OpenSSL/0.9.5a mod_auth_pam/1.0a FrontPage/4.0.4.3 mod_perl/1.24" intitle:index.of
"Apache/1.3.14 - PHP4.02 - Iprotect 1.6 CWIE (Unix) mod_fastcgi/2.2.12 PHP/4.0.3pl1" intitle:index.of
"Apache/1.3.14 Ben-SSL/1.41 (Unix) mod_throttle/2.11 mod_perl/1.24_01 PHP/4.0.3pl1 FrontPage/4.0.4.3 rus/PL30.0" intitle:index.of
"Apache/1.3.20 (Win32)" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) PHP/4.0.3pl1 mod_auth_pam_external/0.1 FrontPage/4.0.4.3 mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) PHP/4.0.4 mod_auth_pam_external/0.1 FrontPage/4.0.4.3 mod_ssl/2.8.4 OpenSSL/0.9.6b mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) PHP/4.0.6 mod_ssl/2.8.4 OpenSSL/0.9.6 FrontPage/5.0.2.2510 mod_perl/1.26" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.0.3pl1 mod_auth_pam_external/0.1 FrontPage/4.0.4.3 mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.0.3pl1 mod_fastcgi/2.2.8 mod_auth_pam_external/0.1 mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.0.4 mod_auth_pam_external/0.1 mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.0.6 mod_auth_pam_external/0.1 FrontPage/4.0.4.3 mod_perl/1.25" intitle:index.of
"Apache/1.3.20 Sun Cobalt (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b mod_auth_pam_external/0.1 mod_perl/1.25" intitle:index.of
"Apache/1.3.26 (Unix) Debian GNU/Linux PHP/4.1.2 mod_dtcl" intitle:index.of
"Apache/1.3.26 (Unix) PHP/4.2.2" intitle:index.of
"Apache/1.3.26 (Unix) mod_ssl/2.8.9 OpenSSL/0.9.6b" intitle:index.of
"Apache/1.3.26 (Unix) mod_ssl/2.8.9 OpenSSL/0.9.7" intitle:index.of
"Apache/1.3.26+PH" intitle:index.of
"Apache/1.3.27 (Darwin)" intitle:index.of
"Apache/1.3.27 (Unix) mod_log_bytes/1.2 mod_bwlimited/1.0 PHP/4.3.1

（续）

搜索特定且隐蔽的服务器版本的查询
FrontPage/5.0.2.2510 mod_ssl/2.8.12 OpenSSL/0.9.6b" intitle:index.of
"Apache/1.3.27 (Unix) mod_ssl/2.8.11 OpenSSL/0.9.6g FrontPage/5.0.2.2510 mod_gzip/1.3.26 PHP/4.1.2 mod_throttle/3.1.2" intitle:index.of

这些无序的标志使用的共同之处就是使用括号来标注服务器的操作系统。例如，Apache/1.3.26 (Unix)表示它是一个基于UNIX的操作系统。还有其他一些更特殊的标志，部分列举如下：

- CentOS
- Debian
- Debian GNU/Linux
- Fedora
- FreeBSD
- Linux/SUSE
- Linux/SuSE
- NETWARE
- Red Hat
- Ubuntu
- UNIX
- Win32

利用这些操作系统标志以及网站服务器版本标志上的信息，攻击者便可以量身定制一个特定的攻击。即便该信息没有暴露出某一特定的缺陷，攻击者仍然可以将该信息来进行数据挖掘以及信息搜集工作——我们将在后续章节讨论。

3.4 险境：遍历技术

我们将要研究的下一个技术称作遍历（traversal）。这里的遍历有跨越的意思。利用遍历技术，攻击者可以通过一个很细微的“蛛丝马迹”来制造巨大的破坏。

3.4.1 目录遍历

为了阐明遍历有什么作用，我们不妨来看一下使用intitle:index.of inurl:“/admin/*”查找到的一个目录列表，如图3-12所示。

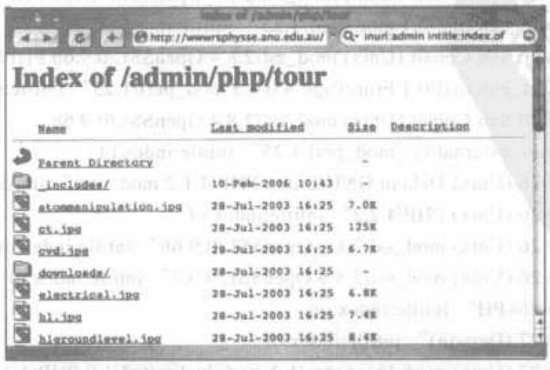


图3-12 使用index.of找到的遍历

在这个例子中，我们得到了一个相对URL：/admin/php/tour。如果仔细观察这个URL，不难发现当前位置的上两级目录是一个名为“admin”的目录。如果我们点击“parent directory”链接，将回到上一级目录，即“php”目录。点击“php”目录中的“parent directory”链接会返回到“admin”目录，这是一个很有用的目录。这是最基本的目录遍历。我们可以挖掘每个父目录以及每个子目录，以寻找有用的东西。或者，我们也可以使用一个和inurl组合在一起的富有创意的site搜索在某个特定的子目录中查找特定的文件或者关键字，例如site:anu.edu inurl:admin ws_ftp.log。也可以通过在地址栏中修改URL来挖掘这个目录结构。

不管在目录树中怎样“行走”，我们都已经超出了Google搜索的范围，而在目标Web服务器中徘徊。这是最基本的遍历，我们把这种遍历单独称为目录遍历（directory traversal）。另外一个简单的例子是用单词student或者public来代替admin。还有一种更为危险的遍历技术，它能够允许攻击者利用软件的缺陷来遍历网站服务器目录树之外的目录。例如，如果一个网站服务器安装在/var/www目录下，并且可以公开访问的Web文档位于/var/www/htdocs目录，默认情况下任何访问该网站服务器顶层目录的用户实际上都是在浏览/var/www/htdocs目录中的文件。正常情况下，该网站服务器不会允许网络用户访问/var/www/htdocs目录之上的文件。现在，我们假设在这个服务器上安装了一个编码很糟糕的第三方软件产品，它的参数是一个目录名。这个软件所使用的一个正常的URL为www.somesadsite.org/badcode.pl?page=/index.html。这个URL命令badcode.pl程序去“抓取”位于/var/www/htdocs/index.html之上的文件，并且把它显示给用户，这一动作可能还会加上一个漂亮的页面头部和脚注。攻击可能会试图利用这种程序来发送一个URL给服务器，例如www.somesadsite.org/badcode.pl?page=../../../etc/passwd。如果这个badcode.pl程序容易受到目录遍历攻击，那么它就会突破/var/www/htdocs目录，到达服务器的真实根目录，然后再钻到/etc目录并“获取”系统的口令文件，再用一个带有漂亮的头部和脚注的网页来显示它，并呈现在用户面前！

有些自动化的工具能够更好地完成这种查找这类文件和缺陷的工作，当然前提是你不介意它们制造的噪音。如果你是一名程序员，可能会对Libwhisker Perl Library比较感兴趣，它由Rain Forest Puppy(RFP)编写并维护，可以在www.wiretrip.net/rfp上找到。Security Focus写了一遍介绍怎样使用Libwhisker的文章。这篇文章很好，可以从www.securityfocus.com/infocus/1798获得。如果你不是一名程序员，那么同样可以从Wiretrip网站获得的RFP的Whisker工具以及其他基于Libwhisker的工具（例如由sullo@cirt.net编写的nikto）都非常好，据说它们甚至比Whisker程序本身更新。另一种（在其他事件中）执行文件和目录挖掘的工具来自SensePost的Wikto，可以访问www.sensepost.com/research/wikto来下载它。Wikto的优点是它不会遇到网站反馈友好的404信息页的误报。

3.4.2 递增置换

另一种和遍历相似的技术是递增置换（incremental substitution）。这种技术通过替换URL中的数字来查找隐藏的，或者没有被其他页面链接的目录或文件。通常Google只会查找被其他页面链接的文件，所以如果没有链接，那么Google就无法找到这些文件。（是的，每种规则都会有例外。参见本章结尾的“常见问题”。）举一个简单的例子，考虑一个由Google搜索到的叫

做exhc-1.xls文档。你可以很容易地修改这个文档的URL，把1换成2之后文件名就变为exhc-2.xls。如果能找到这个文档，那么你就成功地使用了递增置换技术！有些时候能够很容易地使用Google在网站上查找相似的网页，但是要记住的是并不是网站上所有的文件都位于Google的数据库中。仅当你首先确定了简单的修改查询不能找到文件时，再使用这种技术。

这种技术不仅能够用于文件名，任何在URL中包含数字的都可以使用，甚至能够用于脚本的参数。使用这种技术来玩转脚本的参数已经超出了本书内容的范围，但是如果你对亲自尝试某些简单的文件或者目录置换比较感兴趣的话，可以通过类似于filetype:xls inurl:1.xls或者intitle:index.of inurl:0001，甚至1.jpg这样的图片搜索来定位一些测试站点。然后使用置换技术来修改URL中的数字以查找网站上的其他文件或目录。下面是一些例子：

- /docs/bulletin/2.xls可以修改为/docs/bulletin/2.xls。
- /DigLib_thumbnail/spmg/hel/0001/H/可以修改为/DigLib_thumbnail/spmg/hel/0002/H/。
- /gallery/wel008-1.jpg可以修改为/gallery/wel008-2.jpg。

3.4.3 拓展遍历

我们已经讨论了文件扩展名以及如何使用filetype操作来查找具有特定文件扩展名的文件。例如，我们可以很容易使用类似于filetype:HTM ⊖ 这样的查询来查找HTM文件。（注意filetype查询需要一个查询参数。以HTM结尾的文件通常在URL中都含有HTM！）一旦你找到了HTM文件，就可以应用置换技术来查找具有相同的文件名，但扩展名不同的文件。例如，如果你找到了/docs/index.htm，那么就可以把URL修改为/docs/index.asp来查找docs目录下的index.asp文件。如果你觉得这样做似乎没有什么意义，那么放心，它确实是非常没有意义的。但是，我们可以创造更为灵巧的置换。考虑如图3-13所示的目录列表。这个列表给出了一个十分常见的做法的证据，即网页备份文件的创建。

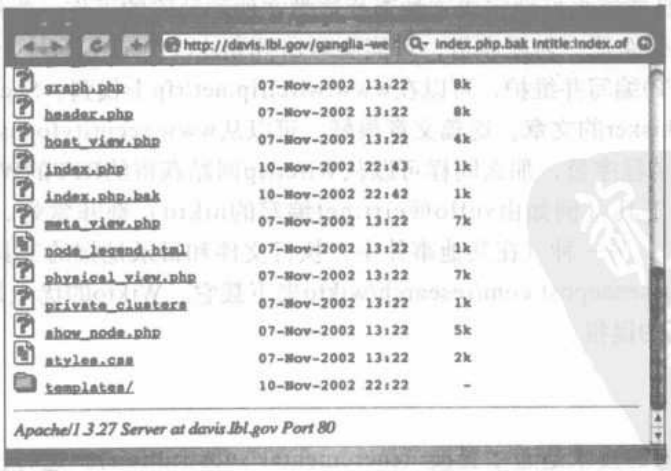


图3-13 网页文件的备份很常见

⊖ 要记住的是，过去，filetype搜索常常需要一个搜索参数。现在，它们不再需要了。以前所有的文件类型查找都要求多写出一个扩展名。filetype:htm无法正常工作，但是filetype:htm htm可以！

从安全性的观点来看，备份文件是一个非常有用的发现。一般情况下，备份文件都是原始文件的老版本。这一点可以由图3-13证明。网站上的备份文件有一个很有趣的副作用，即能够看到它们的源代码。对于安全行业从业者来说，一个网页的源代码十分有用，因为它可能包含关于作者、代码创建和更新的过程以及认证信息等幕后信息。

为了深入了解这一概念，我们再来看一看图3-13中所示的目录列表。点击index.php链接能够在浏览器中显示出带有相应的图片和文本的网页，这正是该网页的作者想要的结果。如果这是一个HTM或者HTML文件，那么查看该页面的源代码就非常简单——只需右击页面并选择view source（查看源文件）即可。相反的是，PHP文件最先在服务器上执行。接着，执行程序的效果被以HTML代码格式发送到浏览器，之后浏览器再显示之。对那些由PHP脚本生成的HTML代码执行view source（查看源文件）命令并不会显示PHP源代码，而只会显示HTML。除非服务器的配置出了某些问题，否则是不可能查看到实际的PHP源代码的。此类的错误配置会将PHP代码拷贝到一个文件名以非PHP（例如BAK）结尾的文件中。大多数Web服务器都无法理解BAK文件是什么。然后，这些服务器会以文本的方式显示PHP.BAK文件。当发生这种情况时，实际的PHP源代码就会以文本的方式显示在浏览器中。如图3-14所示，PHP源代码将能够泄露大量的信息，例如泄露SQL（Structured Query Language，结构化查询语言）查询语句，而这些查询语句又泄露了存储Web服务器数据的SQL数据库的结构信息。

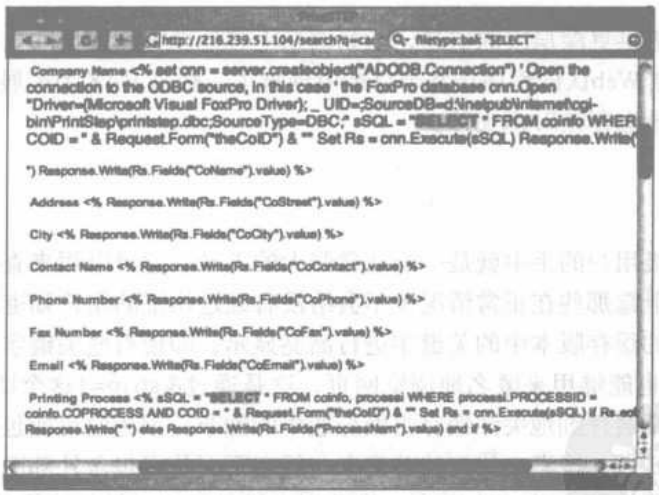


图3-14 暴露SQL数据的备份文件

确定服务器上备份文件的名称的最容易的方法是使用intitle:index.of来查找目录列表，或者使用类似于intitle:index.of index.php.bak或inurl:index.php.bak之类的查询来搜索特定的文件。但是目录列表一般很少见，尤其是在企业级别的Web服务器更为少见。尽管如此，需要知道Google的缓存总是会及时地捕捉到一个页面的快照。就因为Web服务器当前并不存放目录列表并不能意味着该网站从不显示目录列表。图3-15中的页面是在Google缓存中发现的，这个服务器曾因为缺失index.php文件（或者类似的文件）而显示出了目录列表。在这个例子中，如果你现在去访问该服务器网站，会发现它是一个很正常的页面，这是因为缺失的index文件已经重新

创建了。但是，如果点击显示这个目录列表的缓存链接，就会显示出该服务器上的文件列表。这个文件列表可以用来非常方便地查找那些可能仍然存在于服务器上的文件（通过修改URL），而不用再去猜测文件的扩展名。

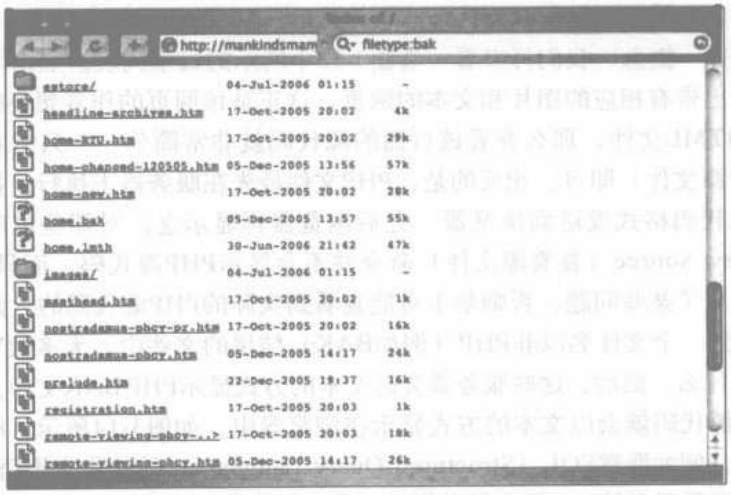


图3-15 缓存页面能够泄露目录列表

目录列表也能够提供更深层次的信息，比如网站上其他地方所使用的文件扩展名。即，如果一个系统管理员或者Web认证程序在某个目录中创建了一些以.BAK为扩展名的备份文件，那么很有可能在其他目录也存在BAK文件。

3.5 总结

Google缓存在高级用户的手中就是一个非常强大的工具。它可以用来查找某些页面的老版本，这些页面可能会泄露那些在正常情况下不会给没有经过认证的用户知道的信息。Google缓存也可以用来对网页的缓存版本中的关键字进行高亮显示，即使有些关键字并不是用于搜索该页面的。Google缓存也能够用来匿名地浏览网页，这是通过&strip=1这个URL参数来实现的。高级的Google用户常常会仔细地关注缓存页面标题中的细节，因为它可能包含了十分重要的信息，比如网页的抓取日期，搜索中找到的关键字，缓存页面是否包含外部图片，原始页面的链接以及用于访问页面缓存版本的URL文本。目录列表提供了Web服务器的独特的隐匿的信息，目录遍历技术则允许攻击者四处寻找不能向大众公开的文件。

3.6 快速查找解决方案

使用缓存进行匿名浏览

- 点击缓存链接并不是只从Google的数据库中加载网页，它还会连接到真实的服务器以访问图片以及其他非HTML内容。
- 在缓存URL的尾部添加&strip=1将只显示缓存网页的文本。以这种方式访问缓存网页不

会连接到网站的真实服务器，而且如果你使用的是本章中提到的剪切和粘贴的方法，还能够保护你的匿名性。

查找目录列表

- 目录列表包含了大量很有价值的信息。
- 最好的查找包含目录列表页面的方法是使用诸如intitle:index.of “parent directory”或者intitle:index.of name size之类的查询。

在列表中查找特定的目录

- 你可以很容易地在目录列表中查找特定的目录，而只需给index.of查询加上一个目录名即可。例如，intitle:index.of inurl:backup可以用来查找URL中含有单词backup的目录列表。如果URL中含有单词backup，那么很有可能这是一个目录名。

在目录列表中查找特定的文件

- 你可以很容易地在目录列表中查找特定的文件：只需简单地在index.of查询中加入文件名即可，例如intitle:index.of ws_ftp.log

目录列表中的服务器版本信息

- 某些服务器，尤其是Apache和Apache的派生服务器常常会在目录列表的底部加上一个服务器标志。这些服务器标志可以通过拓展index.of搜索来查找到，如intitle:index.of server.at。
- 你也可以搜索特定版本的Web服务器，这是通过添加一个正确格式的服务器标志来实现的。例如，intitle:index.of server.at “Apache Tomcat/”能够找到运行各种版本的Apache Tomcat服务器的网站。

目录遍历

- 一旦你在目标Web服务器上找到了某个特定的目录，就可以使用这个技术来查找其他的目录或者子目录。
- 完成这种任务的一个简单的办法是利用目录列表。只要点击parent directory链接，就可以进入当前目录的上一级目录。如果该父目录仍包含其他的目录列表，那么你可以通过点击其链接来挖掘其他的目录。如果父目录没有显示目录列表，那么你可能就需要诉诸于一种更为复杂的方法了，即猜测目录名并把它们添加到父目录URL的尾部。或者，考虑在Google搜索中使用site和inurl关键词。

递增置换

- 递增置换是“取一个数并用下一个更大的或者更小的数来替代这个数”的形象的说法。
- 这个技术可以用于挖掘在目录名或者文件名中使用数字的站点。这种技术只需要用更大的或者更小的数来替代原来的数字，但是要注意保持文件名或者目录名中剩下的部分不变（注意那些零）。或者，在Google搜索中使用site，再加上inurl或者filetype关键字。

拓展遍历

- 这种技术可以用于查找文件名相同，而扩展名不同的文件（例如备份文件）。
- 实现拓展遍历技术最容易的方法是在URL中，用一种扩展名来替代另一种扩展名，例如，用bak代替html。

- 目录列表，尤其是缓存目录列表，是最容易判定备份文件是否存在的方法，也能够判定可能用于网站其他部分的扩展名。

3.7 网站链接

www.all-nettools.com/pr.htm 一个简便的代理检查工具，它能够帮助你测试正在使用的代理服务器。

http://www.sensepost.com/research/wikto Sensepost的Wikto工具，一个很强大的Web扫描工具，它还结合了使用Ggoogle Hacking数据库的Google查询测试。

3.8 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：查询备份文件看起来很麻烦，是否有更好的方式？

答：如果说更好是更快的意思的话，那么这种办法是有的。很多自动的Web工具（例如来自www.spidynamics.com的WebInspect）都提供了这种功能，它们能够针对现有的文件名查询各种相应的扩展名文件是否存在，例如，将一个现有的index.html文件转换为index.html.bak或者index.bak。这些搜索通常都非常全面，不过有些过于大张旗鼓了，几乎都会让站点警觉到你在进行地毯式的搜索。相比于Google Hacking，WebInspect更适合于查询备份文件，但是很多情况下，一个简单的Google搜索可以用来在不对站点管理员或侵入检测系统（Intrusion Detection Systyem，IDS）“打草惊蛇”的情况下侦测站点的安全性。除此之外，还有一个好处，那就是任何通过Google搜集到的信息都可以在以后的评估中重复使用。

问：备份文件似乎能够引发安全问题，但是这些文件能够帮助网站的开发，而且还提供了在进行回滚操作时所需要的信息。有没有办法既可以保留备份文件而又不至于导致安全风险？

答：有这种办法。备份文件的主要问题是，在大多数情况下，Web服务器会用不同于HTML的方式显示它们，这是由它们的扩展名与HTML等类型的文件的扩展名不同所导致的。有几种选择可以解决这种问题。首先，在创建备份文件时，保持扩展名不变。不要把index.php复制为index.bak，而是复制为类似于index.bak.php这样的文件。这时，服务器仍然能够识别出它是一个PHP文件。其次，你可以把备份文件保存在非Web目录中。即把它们保存在一个你可以访问，而Web浏览用户无法访问的地方。第三个（也是最好的一个）选择是使用一个真正的配置管理系统。可以考虑使用CVS风格的系统，这种系统允许你注册并能够检查源代码。这时，你可以随时回滚到一个老版本，而不需要担心备份文件的问题。

第4章　文档加工与数据库挖掘

4.1　简介

互联网上的文档都是有价值的。“好人”和“坏人”都能够利用在文档中发现的信息来达到他们各自不同的目的。在这一章中，我们不仅学习如何利用Google来搜索这些文档，还会了解怎样在这些文档中搜索信息。文档的类型很多，我们不指望把所有的类型都讲述一遍，我们将看一看一些根据功能分类的文档。具体来说，我们会学习一些例如配置文件，日志文件和Office文档这样的类别。在了解了各种文件类型之后，我们将探究数据库挖掘领域。我们不会详细地介绍结构化查询语言（Structured Query Language，SQL）或者数据库结构和应用，而是介绍只拥有搜索引擎的Google黑客是怎样搜索和入侵数据库系统的方法。

有一点很重要，对于文档挖掘，Google只会搜索编译过的或者可视的文档的视图。例如，让我们来看一个Microsoft Word文档。如图4-1所示，这种文档包含有元数据（metadata）。元数据包括主题、作者、管理器和公司等许多东西。Goolge并不会搜索元数据。如果你对文件的元数据比较感兴趣，那么你只有把实际的文件下载下来，然后自己去检查元数据。

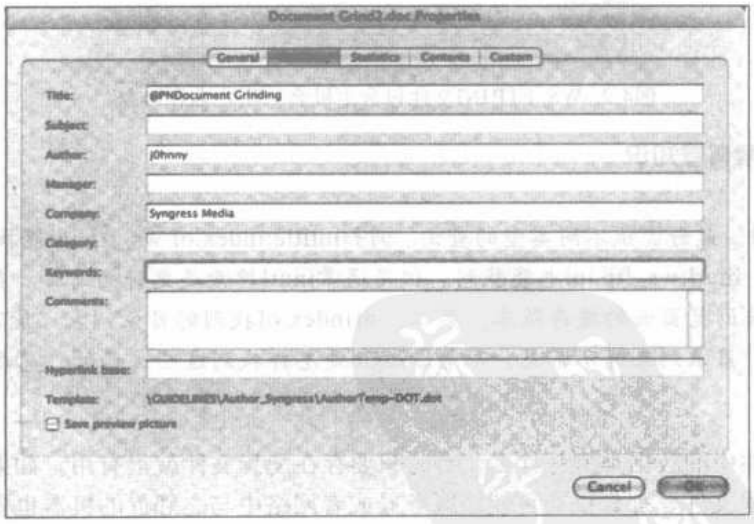


图4-1　Microsoft Word元数据

4.2　配置文件

配置文件存储了程序的设置。攻击者（亦称为“安全专家”）能够利用这些文件深入了解

程序的用法，甚至是了解文件所在的系统或网络是如何使用或配置的。正如我们在前面几章中看到的那样，即使是最少量的信息也会引起经验丰富的攻击者的兴趣。

让我们来看一下图4-2中所示的文件。这个文件是用查询filetype:ini inurl:ws_ftp搜索到的，它是WS_FTP客户端程序的配置文件。在下载并安装WS_FTP程序之后，这个配置文件只包含了一个流行的、公开的互联网FTP服务器列表。但是，过一段时间之后，这个配置文件会自动更新而包含了用户所连接的FTP服务器的名称、目录、用户名和口令。虽然口令是加密存储的，但是一些免费的程序能够比较容易地破译这些口令。

图4-2 WS_FTP.INI文件包含主机名、用户名和口令

Google搜索背景知识

搜索文件

要搜索文件，最好尝试不同类型的查询。例如intitle:index.of ws_ftp.ini能找到一些结果，同样filetype:ini inurl:ws_ftp.ini也能找到。但是通常inurl搜索是更好的选择。第一，filetype搜索能让你直接浏览页面的缓存版本。第二，由index.of找到的目录列表可能不会让你访问该文件。第三，目录列表并不常见。不管Google是怎样找到这些文件的，总之filetype搜索都能找到它们。

不管配置文件中的数据类型是什么，有时只要存在配置文件就很有用。如果在一个服务器上找到了一个配置文件，那么很有可能该服务器或者网络中与之邻近的机器也安装了相应的程序。虽然这个FTP客户端软件的例子只是一个小小的示例，但是不妨来搜索一下filetype:conf inurl:firewall，它能够搜索出通用的防火墙配置文件。这个例子向我们展示了一种最通用的针对配置文件的命名转换方法，即使用conf文件扩展名。也可以组合使用其他的通用命名转换来搜索其他相同的通用命名转换。一种最常见的查找配置文件的基本搜索是(inurl:conf OR inurl:config OR inurl:cfg)，它把三个最常见的配置文件前缀组合在一起进行搜索。你还可以选

择使用filetype操作符。

如果攻击者知道软件作者或者制造商设计的配置文件的名称，那么他就可以简单地使用filetype和inurl操作符来创建一个针对该文件名的搜索。但是，大多数程序都允许引用任何名称的配置文件，这让Google搜索变得稍为复杂。在这些情况下，可以想到利用配置文件的内容来搜索，即提取出特有的字符串来创建有效的基本搜索。有时，把一个通用的基本搜索和软件产品的名称（或其缩略形式）组合在一起也能产生令人满意的结果，例如，图4-3所示的对MRTG的搜索(inurl:conf OR inurl:config OR inurl:cfg)。

图4-3 普通配置文件搜索

虽然这里的第一个搜索结果与标记差不多，但是即使是最好的配置文件搜索，返回的结果也往往都是一页页的样例或者示例文件，如图4-4所示的MRTG样例配置文件。

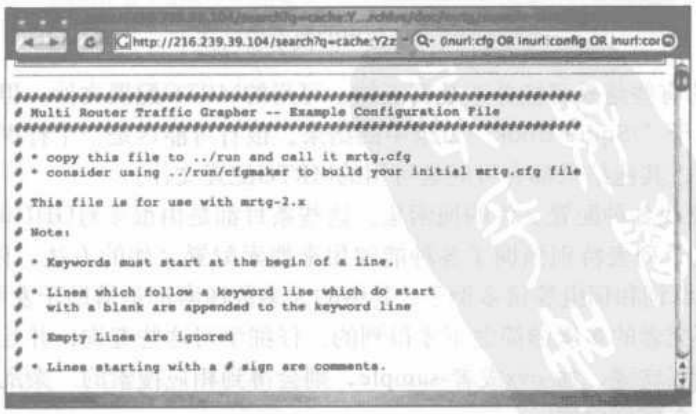


图4-4 需要过滤的样例配置文件

再次把我们带回到可能是Google黑客的武器库中最有价值的武器：有效的搜索缩简技术。下面是Google黑客在搜索配置文件时常考虑的几点。

- 使用可用的配置文件中特有的单词或词组来创建一个强大的基本搜索。
- 过滤掉sample、example、test、howto和tutorial单词以剔除明显的示例文件。
- 用-cvs过滤掉CVS存储器，它们通常存放了默认的配置文件。
- 如果是搜索UNIX程序的配置文件，那么就要过滤掉manpage或者Manual。
- 搜索示例配置文件中最常改变的域，并且对该域执行缩简搜索，以剔除可能的“垃圾”或者样例文件。

为了演示这些要点，让我们先来看一下如何搜索filetype:cfg mrtg “target[*]” -sample -cvs -example，它能够找到可用的MRTG文件。如图4-5所示，这个查询使用了一个特有的字符串“target[*]”（虽然在Google中它不太常见，但是它确实是一个不错的起点），并且删除了可能的示例和CVS文件，所以返回了恰当的结果。

图4-5 一种常用的搜索缩简技术

图4-5中所示的有些结果可能并不是真正的、可用的MRTG配置文件，但是它们也都有可能是，除了第一条位于“/Squid-Book”目录中的结果。很有可能这是一个样例文件，但是由于我们所用的缩简技术，其他结果都有可能是可用的MRTG配置文件。

表4-1列出了查找各种配置文件的搜索集。这些条目都是由很多对GHDB做出了贡献的人的手中收集而来。这个列表特别强调了各种能够用来搜索配置文件的方法。你会看到CVS缩简，样例缩简，特有的单词和词组等诸多例子。其中的大多数搜索都是借助于发明者丰富的想像力，以及在一些搜索研究者的多次缩简之下才得到的。仔细学习这些查询，并且自行尝试实践。如果删除其中的一些限定词，如-cvs或者-sample，则会得到相应搜索的“杂乱的”版本，这可能会更有助于对这些查询的理解。

表4-1 配置文件搜索示例

说 明	查 询
PHP配置文件	intitle:index.of config.php
PHP配置文件	inurl:config.php dbuname dbpass
CGIIRC配置文件	intitle:index.of cgiirc.config
CGIIRC配置文件	inurl:cgiirc.config
IPSEC配置文件	inurl:ipsec.conf -intitle:manpage
ws_ftp配置文件	intitle:index.of ws_ftp.ini
eggdrop配置文件	eggdrop filetype:user user
samba配置文件	inurl:"smb.conf" intext:"workgroup" filetype:conf
防火墙配置文件	filetype:conf inurl:firewall -intitle:cvs
vtunnelD配置文件	inurl:vtund.conf intext:pass -cvs
OpenLDAP配置文件	filetype:conf slapd.conf
PHP配置文件	inurl:php.ini filetype:ini
FTP配置文件	filetype:conf inurl:proftpd.conf -sample
WV Dial配置文件	inurl:"wvdial.conf" intext:"password"
OpenLDAP配置文件	inurl:"slapd.conf" intext:"credentials" -manpage -"Manual Page" -man: -sample
OpenLDAP配置文件	inurl:"slapd.conf" intext:"rootpw" -manpage -"Manual Page" -man: -sample
WS_FTP配置文件	filetype:ini ws_ftp pwd
MRTG配置文件	filetype:cfg mrtg "target[*]" -sample -cvs -example
WRQ Reflection配置文件	filetype:r2w r2w
Prestige路由器配置文件	"Welcome to the Prestige Web-Based Configurator"
GNU Zebra配置文件	inurl:zebra.conf intext:password sample -test -tutorial -download
GNU Zebra配置文件	inurl:ospfd.conf intext:password sample -test -tutorial -download
YTST配置文件	filetype:cfg ks intext:rootpw -sample -test -howto
Netscape服务器配置文件	allinurl:".nsconfig" -sample -howto -tutorial
UnrealIRCd配置文件	filetype:conf inurl:unrealircd.conf -cvs -gentoo
psyBNC配置文件	filetype:conf inurl:psybnc.conf "USER.PASS="
SSL配置文件	inurl:ssl.conf filetype:conf
LILO配置文件	inurl:lilo.conf filetype:conf password -tatercounter2000 -bootpwd -man
MySOL配置文件	filetype:cnf my.cnf -cvs -example
Oracle客户端配置文件	filetype:ora ora
Mandrake配置文件	filetype:cfg auto_inst.cfg
Oekakibs配置文件	filetype:conf oekakibbs
LeapFTP客户端配置文件	LeapFTP intitle:"index.of./" sites.ini modified
.Net Web应用程序配置文件	filetype:config config intext:appSettings "User ID"
WS_FTP配置文件	"index of/" "ws_ftp.ini" "parent directory"
ODBC客户端配置文件	inurl:odbc.ini ext:ini -cvs
FlashFXP配置文件	filetype:ini inurl:flashFXP.ini
通用配置文件	ext:ini intext:env.ini
Certificate Services配置文件	filetype:inf inurl:capolicy.inf
NoCatAuth配置文件	ext:conf NoCatAuth -cvs
Putty已保存的会话数据	inurl:"putty.reg"
Icecast配置文件	"liveice configuration file" ext:cfg -site:sourceforge.net
SoftCart配置文件	intitle:Configuration.File inurl:softcart.exe
Cisco配置文件	intext:"enable secret 5 $"

（续）

说　明	查　询
IIS Web配置文件	filetype:config web.config -CVS
VMWare配置文件	ext:vmx vmx
Radiator Radius配置文件	ext:cfg radius.cfg
Rsync配置文件	xt:conf inurl:rsyncd.conf -cvs -man
Eudora配置文件	ext:ini eudora.ini
emule配置文件	inurl:preferences.ini "[emule]"
abyss web服务器配置文件	intitle:index.of abyss.conf
UNIX Frontpage Extensions配置文件	filetype:cnf inurl:_vti_pvt access.cnf
Shoutcast配置文件	intitle:'Index of" sc_serv.conf sc_serv content
HP以太网交换配置文件	intitle:'DEFAULT_CONFIG - HP"
Oracle配置文件	filetype:ora tnsnames
Counterstrike配置文件	inurl:server.cfg rcon password
Steam配置文件	intext:"SteamUserPassphrase=" intext:"SteamAppUser="-"username"-"user"
CGI日历配置文件	inurl:cgi-bin inurl:calendar.cfg
Cisco配置文件	intext:"enable password 7"
YABB论坛管理文件	inurl:/yabb/Members/Admin.dat
FlashFXP站点数据文件	inurl:"Sites.dat"+"PASS="
Ruby on Rails数据库连接器文件	ext:yml database inurl:config
Cisco配置文件	enable password | secret "current configuration" -intext:the
通用配置文件	intitle:index.of.config

4.3 日志文件

日志文件记录了一些信息。根据应用程序不同，日志文件中记录的信息可以包括从时间戳和IP地址到用户名与口令，甚至是令人难以相信的敏感数据，例如信用卡账号！

与配置文件类似的是，日志文件也有一个可以用做基本搜索一部分的默认的名称。日志文件最常见的文件扩展名就是log，同样，最简单的基本搜索就是filetype:log inurl:log或者是更为简单的ext:log log。回忆前面所学的，我们知道ext(filetype)操作符至少需要一个搜索参数。似乎日志文件搜索比配置文件搜索返回的样例和示例文件少得多，但是在某些情况下，还需要用到搜索缩简。参考前面列出的配置文件缩简规则。

表4-2列出了收集自GHDB的日志文件搜索集。这些搜索给出了Google黑客所用的各种技术，而且也是一个极好的学习工具，可以帮助你在渗透测试中创建自己的搜索。

表4-2　日志文件搜索示例

查　询	说　明
"ZoneAlarm Logging Client"	ZoneAlarm日志文件
"admin account info"　filetype:log	管理员日志
"apricot - admin"　00h	Apricot日志
"by Reimar Hoven. All Rights Reserved. Disclaimer" | inurl:　"log/logdb.dta"	PHP Web统计表日志
"generated by wwwstat"	www统计表

（续）

查　询	说　明
“Index of” / “chat/logs”	聊天日志
“MacHTTP” filetype:log inurl:machttp.log	MacHTTP
“Most Submitted Forms and Scripts” “this section”	www统计表
“sets mode: +k”	IRC日志，频道密钥设置
“sets mode: +p”	IRC聊天日志
“sets mode: +s”	IRC日志，加密频道设置
“The statistics were last updated” “Daily” -microsoft.com	网络活动日志
“This report was generated by WebLog”	weblog生成的统计表
“your password is” filetype:log	口令日志
QueryProgram “ZoneAlarm Logging Client”	ZoneAlarm日志文件
+htpasswd WS_FTP.LOG filetype:log	WS_FTP客户端日志文件
+intext: “webalizer” +intext: “Total Usernames” +intext: “Usage Statistics for”	Webalizer统计表
ext:log “Software: Microsoft Internet Information Services *.*”	IIS服务器日志文件
ext:log password END_FILE	Java口令文件
filetype:cfg login “LoginServer=”	Ultima Online日志文件
filetype:log “PHP Parse error” | “PHP Warning” |	PHP错误文件
filetype:log “See `ipsec —copyright”	BARF日志文件
filetype:log access.log -CVS	HTTPD服务器读取日志
filetype:log cron.log	UNIX计时程序日志
filetype:log hijackthis “scan saved”	Hijackthis扫描日志
filetype:log inurl: “password.log”	口令日志
filetype:log inurl:access.log TCP_HIT	Squid存取日志
filetype:log inurl:cache.log	Squid缓存日志
filetype:log inurl:store.log RELEASE	Squid磁盘存储日志
filetype:log inurl:useragent.log	Squid用户代理日志
filetype:log iserror.log	MS Install Shield日志
filetype:log iserror.log	MS Install Shield日志
filetype:log iserror.log	MS Install Shield日志
filetype:log username putty	Putty SSH客户端日志
filetype:log username putty	Putty SSH客户端日志
intext: “Session Start * * * *:*:* *” filetype:log	IRC/AIM日志文件
intitle: “HostMonitor log” | intitle: ” HostMonitor report”	HostMonitor
intitle: “Index Of” -inurl:maillog maillog size	Mail日志文件
intitle: “LOGREP - Log file reporting system” -site:itefix.no	Logrep
intitle:index.of .bash_history	UNIX bash shell历史日志
intitle:index.of .sh_history	UNIX shell历史日志
intitle:index.of cleanup.log	Outlook Express清理日志
inurl:access.log filetype:log -cvs	Apache存取日志（Windows）
inurl:error.log filetype:log -cvs	Apache错误日志
inurl:log.nsf -gov	Lotus Domino
log inurl:linklint filetype:txt - “checking”	Linklint日志
Squid cache server reports	squid服务器缓存报告

日志文件会泄露各种类型的信息，例如图4-6所示的搜索filetype:log username putty的结果。

这个日志文件列出了机器名以及相应的用户名，这些信息可以在攻击该机器时再次利用。

图4-6 Putty日志文件泄露了敏感的数据

Office文档

术语“Office文档”通常是指由字处理软件、表格软件以及轻量级数据库程序创建的文档。常见的字处理软件有Microsoft Word、Corel WordPerfect、MacWrite和Adobe Acrobat。常见的表格程序有Microsoft Excel、Lotus 1-2-3和Linux的Gnumeric。其他勉强归到Office文档一类的文档还包括Microsoft PowerPoint、Microsoft Works和Microsoft Access文档。表4-3列出了一些更为常见的Office文档文件类型，大致按照互联网上的流行程度（根据Google查询的结果）排列。

表4-3 流行的Office文档文件类型

文件类型	扩展名
Adobe 可移植文档格式	Pdf
Adobe PostScript	Ps
Lotus 1-2-3	wk1、wk2、wk3、wk4、wk5、wki、wks、wku
Lotus WordPro	Lwp
MacWrite	Mw
Microsoft Excel	Xls
Microsoft PowerPoint	Ppt
Microsoft Word	Doc
Microsoft Works	wks、wps、wdb
Microsoft Write	Wri
Rich Text Format	Rtf
Shockwave Flash	Swf
Text	ans、txt

在许多情况下，简单地用filetype而不用额外的特殊搜索来查找这些文件是没有用的。Google黑客已经成功地发现了许多有用的文件，他们只是简单地在filetype搜索的结尾放上一些搜索项，如private或者password或者admin。但是，像(inurl:xls OR inurl:doc OR inurl:mdb)这样的简单基本搜索就能够用做许多文件类型的通用搜索。

表4-4列出了GHDB中一些专门针对Office文档的搜索。这个列表中有许多技术都是值得我们去研究学习的。有些搜索，例如filetype:xls inurl:password.xls关注的是具有特定文件名的文件。password.xls文件并不一定专属于某个特定的软件包，但它听起来很有趣，这主要是因为它的名字。其他一些搜索，例如filetype:xls username password email所关注的从文件名转为文件内容。理由是如果一个Excel表格包含单词username password和e-mail，那么很有可能它也包含一些敏感数据，例如口令。好的Google搜索的精髓是把一个通用搜索调整为能发现最合适的结果的搜索。在一名高级Google用户手中，Google在不同类型文档内部搜索的能力就是一种极为强大的工具。

表4-4 搜索可能敏感的Office文档的查询样例

查 询	可能暴露的信息
filetype:xls username password email	口令
filetype:xls inurl: "password.xls"	口令
filetype:xls private	私人数据（用做基本搜索）
Inurl:admin filetype:xls	管理数据
filetype:xls inurl:contact	联系信息、E-mail地址
filetype:xls inurl: "email.xls"	E-mail地址、姓名
allinurl: admin mdb	管理数据库
filetype:mdb inurl:users.mdb	用户列表、E-mail地址
Inurl:email filetype:mdb	用户列表、E-mail地址
Data filetype:mdb	各种数据（用做基本搜索）
Inurl:backup filetype:mdb	备份数据库
Inurl:profiles filetype:mdb	用户配置文件
Inurl:*db filetype:mdb	各种数据（用做基本搜索）

4.4 数据库挖掘

最近，基于Web的数据库应用程序的安全性越来越受到关注，尤其是和数据库进行交互的前端软件。在安全社区里，对SQL注入的讨论已经取代了之前的CGI漏洞的讨论，这说明比起底层的操作系统或者Web服务器软件，数据库已经成了最大的目标。

攻击者通常并不是利用Google来入侵某个数据库或者破坏某个数据库前端应用程序，相反的是Google黑客在互联网上搜索从可能存在漏洞的服务器中泄露的数据库信息。这些信息首先可以用于目标的选择，然后可以用于对该目标发起一次有组织的攻击（相对于一次地面零点盲攻击而言）。基于此，在这里我们并不讨论实际的攻击机制本身，而是讨论一个老练的Google黑客在攻击目标之前实施的令人惊讶的信息收集阶段。

4.4.1 登录入口

登录入口就是基于Web的应用程序的“前门”。显示用户名和口令对话框的登录入口是最值得Web攻击者仔细研究的，这仅仅是因为它是应用程序中安全保护最为仔细的地方。当然这个规则也有例外，但是不妨来打个比方，如果你准备对家里加强安全保护，难道你不打算先确保前门的安全性吗？

图4-7是一个典型的数据库登录入口。这个登录页面不仅表明存在一个SQL服务器，还表明存在Microsoft Web Data Administrator软件包。

图4-7 一个典型的数据库登录入口

不管登录入口的安全强弱程度如何，它的存在都暴露了目标可能使用的软件和硬件的类型。简单来说，登录入口非常适合于顺藤摸瓜。极端地说，一个不安全的登录入口就像是一个欢迎攻击者的页面。在本节结束时，我们来看一些攻击者可能用来搜索网络中数据库前端的查询。表4-5列出了搜索数据库前端或界面的查询。大多数条目都取自GHDB。

表4-5 搜索数据库界面的查询

查 询	可能暴露的信息
allinurl: admin mdb	管理数据库
Inurl:backup filetype:mdb	备份数据库
“ClearQuest Web Logon”	ClearQuest（CQWEB）
inurl:/admin/login.asp	通用登录页面
inurl:login.asp	通用登录页面
filetype:fp5 fp5 - “cvs log”	FileMaker Pro
filetype:fp3 fp3	FileMaker Pro
filetype:fp7 fp7	FileMaker Pro
“Select a database to view” intitle: “filemaker pro”	FileMaker Pro
“Welcome to YourCo Financial”	IBM Websphere
“(C) Copyright IBM” “Welcome to Websphere”	IBM Websphere

（续）

查　询	可能暴露的信息
inurl:names.nsf?opendatabase	Lotus Domino
inurl: "/catalog.nsf" intitle:catalog	Lotus Domino
intitle: "messaging login" "© Copyright IBM"	Lotus Messaging
intitle: "Web Data Administrator - Login"	MS SQL登录
intitle: "Gateway Configuration Menu"	Oracle
inurl:/pls/sample/admin_/help/	Oracle默认手册
inurl:1810 "Oracle Enterprise Manager"	Oracle企业管理程序
inurl:admin_/globalsettings.htm	Oracle HTTP监听程序
intitle: "oracle http server index" "Copyright * Oracle Corporation."	Oracle HTTP服务器
inurl:pls/admin_/gateway.htm	Oracle登录入口
inurl:orasso.wwsso_app_admin.ls_login	Oracle Single Sign-On
"phpMyAdmin" "running on" inurl: "main.php"	phpMyAdmin
"Welcome to phpMyAdmin" " Create new database"	phpMyAdmin
intitle: "phpPgAdmin-Login" Language	phpPgAdmin (PostgreSQL)管理工具
intext: SQLiteManager inurl:main.php Data filetype:mdb	各种数据（用做基本搜索）

Google搜索背景知识

登录入口

一种搜索登录入口的方法是关注单词login（登录）。另一种方法是关注页面底部的copyright（版权）字样。大多数名气很大的入口都会在页面底部放置一个版权注意事项。你可以把它和产品名以及welcome（欢迎）组合在一起进行搜索。如果你试遍了你所知道的数据库，那么可以到http://labs.google.com/sets上输入oracle和mysql，然后点击Large Set，这样就可以得到一个数据库列表。

4.4.2 帮助文件

另外一种攻击者用来搜索或者收集和数据库有关的信息的方法是通过查询与数据库软件一同安装的或者是由其生成的帮助文件。这些帮助文件包括配置文件、调试脚本以及样例数据库文件。表4-6列出了一些用来搜索流行的数据库客户端和服务器包含的或者生成的帮助文件的查询。

表4-6　搜索数据库帮助文件的查询

查　询	说　明
inurl:default_content.asp ClearQuest	ClearQuest Web帮助文件
intitle: "index of" intext:globals.inc	MySQL globals.inc文件，列出链接和证书信息
filetype:inc intext:mysql_connect	PHP MySQL Connect文件，列出链接和证书信息
filetype:inc dbconn	数据库链接文件，列出链接和证书信息
intitle: "index of" intext:connect.inc	MySQL链接文件，列出链接和证书信息
filetype:properties inurl:db intext:password	db.properties文件，列出链接信息
intitle: "index of" mysql.conf OR mysql_config	MySQL配置文件，列出端口号，版本号和MySQL服务器的路径信息

（续）

查　询	说　明
inurl:php.ini filetype:ini	PHP.INI文件，列出链和证书信息
filetype:ldb admin	Microsoft Access加锁（lock）文件，列出数据库和用户名
inurl:config.php dbuname dbpass	旧的config.php脚本，列出用户和口令信息
intitle:index.of config.php	config.php脚本，列出用户和口令信息
“phpinfo.php” -manual	phpinfo.php的输出，列出大量的信息
intitle: “index of” +myd size	MySQL数据目录
filetype:cnf my.cnf -cvs -example	MySQL my.cnf文件，能列出从路径和数据库名到口令与用户名的信息
filetype:ora ora	ORA配置文件，列出Oracle数据库信息
filetype:pass pass intext:userid	dbman文件，列出加密了的口令
filetype:pdb pdb backup (Pilot | Pluckerdb)	Palm数据库文件，能列出各种个人信息

如图4-8的帮助文件例子所示，使用mysql_connect函数的PHP脚本泄露了机器名、用户名和明文口令。严格来讲，这个文件包含了PHP代码，但INC扩展名却让其成为一个包含文件（include file）。确切地说是这个文件的内容吸引了Google黑客。

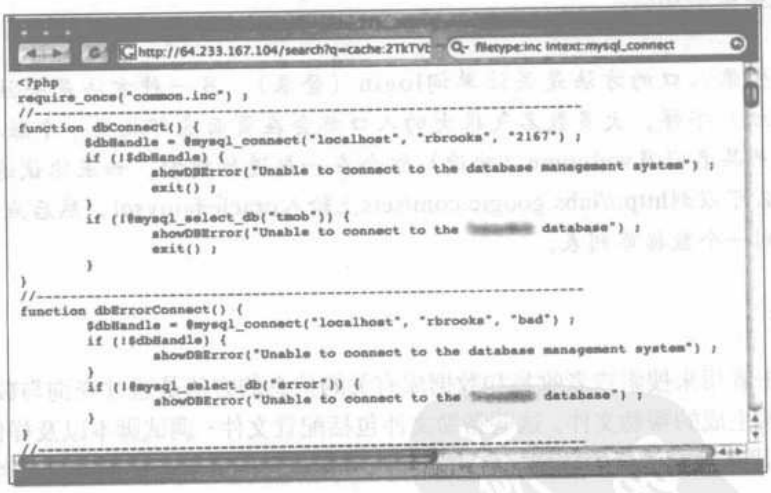


图4-8　PHP文件能泄露机器名、用户名和口令

4.4.3　错误消息

在本书的通篇讨论中，错误消息可以用于各种剖析和信息收集目的。在检测和剖析数据库系统的过程中，错误消息也能够扮演关键的角色。和大多数错误消息一样，数据库错误消息也能用来剖析操作系统和Web服务器的版本。反之，操作系统和Web服务器错误消息也能用于剖析和检测数据库服务器。表4-7列出了用于搜索数据库错误消息的查询。

表4-7 搜索数据库错误消息的查询

说 明	查 询
.NET错误消息可以泄露数据源，甚至是身份验证证书	“ASP.NET_SessionId” “data source=”
500 “Internal Server Error”（Internal服务器错误）可以泄露服务器管理员的E-mail地址以及Apache服务器标题	“Internal Server Error” “server at”
500 “Internal Server Error”（Internal服务器错误）可以泄露站点上运行的Web服务器的类型，并且具有依据信息内部格式化的方式显示其他信息的能力	intitle: “500 Internal Server Error” “server at”
ASP错误消息可以泄露使用的编译器、使用的语言、行号、程序名和部分源代码	filetype:asp“Custom Error Message”Category Source
Access错误消息可以泄露路径名、函数名、文件名和部分代码	“Syntax error in query expression “ -the
Apache Tomcat错误消息可以泄露各种各样的与错误类型相对应的信息	intitle: “Apache Tomcat” “Error Report”
CGI错误消息可能会满足部分代码列表，PERL版本，详细的服务器信息，用户名，启动文件名、格式和查询信息，以及其他更多的信息	intext: “Error Message : Error loading required libraries.”
Chatologica MetaSearch错误可以泄露Apache版本、CGI环境变量、路径名称、栈转储、进程ID、PERL版本、以及更多的信息	“Chatologica MetaSearch” “stack tracking:”
Cocoon XML可能泄露了库函数、cocoon版本叫以及完整的路径名和/或相对的路径名	“error found handling the request” cocoon filetype:xml
ColdFusion错误消息触发了可以帮助搜索SQL注入点的SQL SELECT或INSERT语句	intitle: “Error Occurred While Processing Request” +WHERE (SELECT|INSERT) filetype:cfm
ColdFusion错误消息可以泄露部分源代码、完整的路径名、SQL查询信息、数据库名称、SQL状态信息和本地时间信息	intitle: “Error Occurred” “The error occurred in” filetype:cfm
ColdFusion错误消息可以泄露SQL语句和服务器信息	intitle: “Error Occurred While Processing Request”
ColdFusion错误消息可以泄露源代码、完整路径名、SQL查询信息以及本地时间信息	intitle: “Error Occurred” “The error occurred in” filetype:cfm
ColdFusion错误消息可以泄露很多不同类型的信息	“Error Diagnostic Information” intitle: “Error Occurred While”
DB2错误消息可以泄露文件名、函数名、部分代码和程序状态	“detected an internal error [IBM][CLI Driver] [DB2/6000]”
DB2错误消息可以泄露路径名、函数名、文件名、部分代码和程序状态	An unexpected token “END-OF-STATE MENT” was found
DB2错误消息可以泄露文件名、函数名、文件名、部分代码和程序状态	“detected an internal error [IBM] [CLI Driver] [DB2/6000]”
DB2错误消息可以泄露文件名、函数名、文件名、部分代码和程序状态	An unexpected token “END-OF-STATE MENT” was found
Discuz! Board错误可能会泄露路径信息或者部分SQL代码列表	filetype:php inurl: “logging.php” “Discuz” error

（续）

说　明	查　询
常见SQL消息可以泄露路径名和部分SQL代码	“You have an error in your SQL syntax near”
常见SQL错误可以泄露路径信息	“Warning: Supplied argument is not a valid File-Handle resource in”
常见SQL错误消息可以被用来确定操作系统和Web服务器版本	intitle: “Under construction” “does not currently have”
常见错误消息可以泄露使用的编译器、使用的语言、行号、程序名称以及部分源代码	“Fatal error: Call to undefined function” -reply -the -next
常见错误消息可以泄露完整的路径信息	“Warning:” “SAFE MODE Restriction in effect.” “The script whose uid is” “is not allowed to access owned by uid 0 in” “online”
常见错误消息可以泄露各种信息	“Error Diagnostic Information” intitle: “Error Occurred While”
常见错误消息可以泄露路径名、PHP文件名、行号和包含路径（include path）	intext:” Warning: Failed opening” “on line” “include_path”
常见错误可以泄露完整的路径信息	“Warning: Division by zero in” “on line” -forum
HyperNews错误可以泄露服务器软件、服务器操作系统、服务器账户用户/组（UNIX）、服务器管理员E-mail地址，甚至是栈轨迹	intitle: “Error using Hypernews” “Server Software”
IIS 4.0错误消息可以泄露过时很久的IIS版本的存在	intitle: “the page cannot be found” inetmgr
IIS错误消息可以泄露某些未修改的（以及可能未修补的）IIS服务器	intitle: “the page cannot be found” “internet information services”
Informix错误消息可以泄露路径名、函数名、文件名和部分代码	“A syntax error has occurred” filetype:ihtml
Informix错误消息可以泄露路径名、函数名、文件名和部分代码	“An illegal character has been found in the statement” - “previous message”
MySQL错误消息可以泄露路径名	“supplied argument is not a valid MySQL result resource”
MySQL错误消息可以泄露多种信息	“mySQL error with query”
MySQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“Can't connect to local” intitle:warning
MySQL错误消息可以泄露路径名和部分SQL代码	“You have an error in your SQL syntax near”
MySQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“ORA-00921: unexpected end of SQL command”
MySQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“Supplied argument is not a valid MySQL result resource”
MySQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“Incorrect syntax near”
MySQL错误消息可以泄露路径名、函数名、文件名和部分代码	“Incorrect syntax near” -the

（续）

说　明	查　询
MySQL错误消息可以泄露路径名、函数名、文件名和部分代码	“Unclosed quotation mark before the character string”
MySQL错误消息可以泄露用户名、数据库、路径名和部分SQL代码	“access denied for user” “using password”
MySQL错误消息可以泄露真实的路径名以及服务器上其他PHP脚本列表	“supplied argument is not a valid MySQL result resource”
MySQL错误消息可以泄露各种信息	“MySQL error with query”
MySQL错误可以泄露数据库模式和用户名	“Warning: mysql_query()” “invalid query”
Netscape应用程序服务器或者iPlanet应用程序服务器错误可以泄露过时很久的软件	intitle:” 404 SC_NOT_FOUND”
ODBC SQL错误可能会泄露表查询或者行查询、完整的数据库名以及更多的信息	filetype:asp + “[ODBC SQL”
Oracle SQL错误消息可以泄露完整的Web路径名和/或PHP文件名	“ORA-00921: unexpected end of SQL command”
Oracle SQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“ORA-00933: SQL command not properly ended”
Oracle SQL错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“ORA-00936: missing expression”
Oracle错误消息可以泄露路径名、函数名、文件名和部分SQL代码	“ORA-00933: SQL command not properly ended”
Oracle错误消息可以泄露路径名、函数名、文件名和部分数据库代码	“ORA-00936: missing expression”
Oracle错误消息可以泄露部分SQL代码、路径名、文件名和数据源	“ORA-12541: TNS:no listener” intitle: “error occurred”
Oracle错误消息可以泄露SQL代码、路径名、文件名和数据源	“ORA-12541: TNS:no listener” intitle: “error occurred”
PHP错误日志可以泄露不同类型的信息	filetype:log “PHP Parse error” | “PHP Warning” | “PHP Error”
PHP错误消息可以泄露路径名、函数名、文件名和部分代码	“Warning: Cannot modify header information - headers already sent”
PHP错误消息可以泄露Web根目录和用户ID	“The script whose uid is” “is not allowed to access”
PHP错误消息可以泄露路径名、PHP文件名、行号和包含路径	PHP application warnings failing “include_path”
PHP错误可以泄露Web根路径	“Parse error: parse error, unexpected T_VARIABLE” “on line” filetype:php
PostgreSQL错误消息可以泄露路径信息和数据库名称	“Warning: pg_connect(): Unable to connect to PostgreSQL server: FATAL”
PostgreSQL错误消息可以泄露路径名、函数名、文件名和部分代码	“PostgreSQL query failed: ERROR: parser: parse error”

（续）

说　明	查　询
PostgreSQL错误消息可以泄露路径名、函数名、文件名和部分代码	"Supplied argument is not a valid PostgreSQL result"
PostgreSQL错误消息可以泄露路径名、函数名、文件名和部分代码	"PostgreSQL query failed: ERROR: parser: parse error"
PostgreSQL错误消息可以泄露路径名、函数名、文件名和部分代码	"Supplied argument is not a valid PostgreSQL result"
PostgreSQL错误消息可以泄露路径信息和数据库名称	"Warning: pg_connect(): Unable to connect to PostgreSQL server: FATAL"
SQL错误消息可能会泄露潜在的SQL注入点	"[SQL Server Driver][SQL Server]Line 1: Incorrect syntax near" -forum -thread -showthread
SQL错误消息可以泄露完整的路径信息	"Invision Power Board Database Error"
SQL错误消息可以泄露完整的路径名和/或PHP文件名	"ORA-00921: unexpected end of SQL command"
SQL错误消息可以泄露路径名、函数名、文件名和部分代码（可变）	"Can't connect to local" intitle:warning
SQL错误消息可以泄露路径名、函数名、文件名和部分代码（可变）	"Incorrect syntax near" -the
SQL错误消息可以泄露路径名、函数名、文件名和部分代码（可变）	"access denied for user" "using password"
SQL错误消息可以泄露路径名、函数名、文件名和部分代码	"Incorrect syntax near"
SQL错误消息可以泄露路径名、函数名、文件名和部分代码	"Unclosed quotation mark before the character string"
Sablotron XML错误可以泄露部分源代码、路径和文件名信息以及更多	warning "error on line" php sablotron
Snitz Microsoft Access数据库错误可能会泄露数据库的位置和名称，可能会让论坛很容易遭受有害的下载	databasetype. Code : 80004005. Error Description :
Softcart错误信息可以泄露配置文件位置以及服务器文件路径	intitle:Configuration.File inurl:softcart.exe
这个错误可以泄露因某些原因被拒绝的数据库登录	"Warning: mysql_connect(): Access denied for user: '*@*' "on line" -help -forum
Windows 2000错误可以泄露Windows相当早的版本的存在	intitle:" the page cannot be found" "2004 microsoft corporation"
cgiwrap错误可以泄露管理名和E-mail，端口号，路径名称，以及也可能包含支持人员的诸如电话号码之类的任意信息	intitle: "Execution of this script not permitted"
ht://Dig错误可以泄露管理E-mail、cgi-bin可执行目录、目录结构、搜索数据库文件的位置以及可能的命名规则	intitle: "htsearch error" ht://Dig error
vbulletin错误可以泄露SQL摘要	"There seems to have been a problem with the" " Please try again by clicking the Refresh button in your web browser."

错误消息除了会泄露与数据库服务器有关的信息外，还会泄露更多关于服务器上可能存在的漏洞的危险信息。例如，让我们先来看一下如图4-9所示的一条错误消息“SQL command not properly ended”。这个错误消息是说在SQL语句的结尾没有找到结束字符。例如，如果一个命令接受用户输入，那么攻击者就能够利用这条错误消息中的信息来执行一次SQL注入攻击。

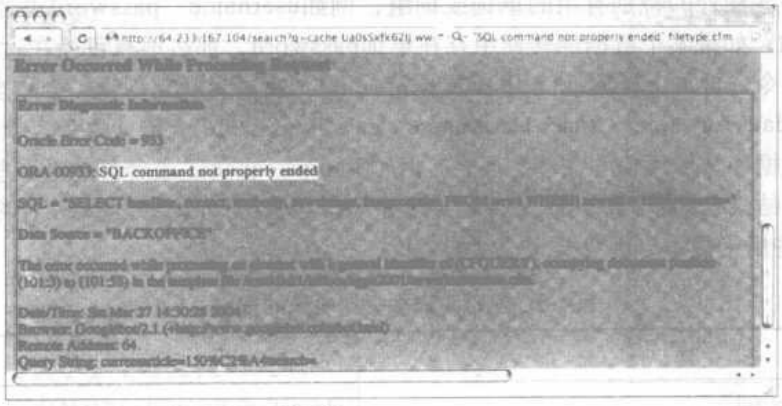


图4-9 一条危险的错误消息

4.4.4 数据库转储

数据库的任意格式的输出都可看做是数据库的转储(dump)。但是，针对Google hacking的目的，我们将使用术语数据库转储（database dump）来描述数据库的基于文本的转换。我们会在下一章中看到，攻击者完全有可能搜索到任何二进制数据库文件，但是在互联网上标准的格式（例如，图4-10所示的基于文本的SQL转储）还是很常见的。

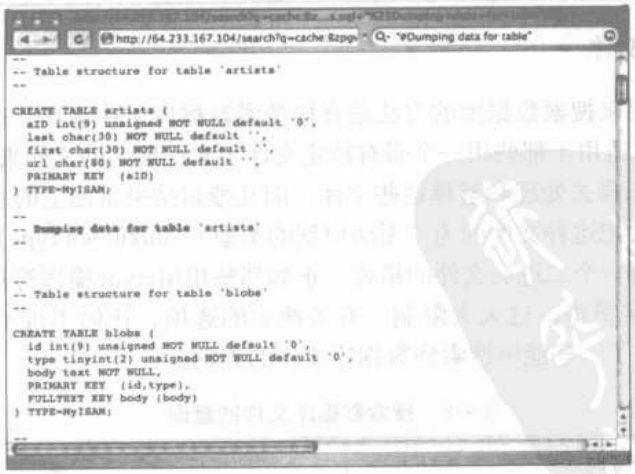


图4-10 一个典型的SQL转储

利用一个完整的数据库转储，数据库管理员就完全能够重建数据库。这意味着一个完整的转储不仅描述了数据库表的结构，还描述了每张表中的记录。根据数据库所包含的数据的敏感

程度，数据库转储能够泄露许多信息，而这也给攻击者提供了一个极好的工具。攻击者有几种方法来搜索数据库转储。最明显的一种方法是关注转储的头部信息，如图4-10所示的查询“#Dumping data for data”。这个技术可以通过仅关注存在于每个转储中的头部来扩展到任意类型的数据库转储头部，这些头部短语是那些不太可能引起误报的特殊短语。

如果指定一些额外的特别有用的单词或词组，例如username、password或者user能让搜索更为精确。例如，如果某个数据库转储中存在单词password，那么很有可能在这个数据库转储中列出了某种口令。再加上合理地使用OR操作符（|），攻击者就能创建一个相当有效的搜索，如“#Dumping data for table” (user | username | pass | password)。除此之外，攻击者还可以关注一些工具给数据库转储添加的文件扩展名，比如filetype:sql sql，或者再加上一些特定的单词、词组或者站点让结果更为精确。SQL文件扩展名也可以用做为一种批处理SQL命令的一般描述。表4-8列出了一些用于搜索SQL数据库转储的查询。

表4-8 搜索SQL数据库转储的查询

查 询	说 明
inurl:nuke filetype:sql	php-nuke或者postnuke CMS转储
filetype:sql password	SQL数据库转储或者批处理SQL命令
filetype:sql “IDENTIFIED BY” -Ccvs	SQL数据库转储或者批处理SQL命令，关注的是“IDENTIFIED BY”，这样能搜索到口令
“# Dumping data for table (username|user|users|password)”	SQL数据库转储或者批处理SQL命令，关注的是有用的项目
“#mysql dump” filetype:sql	SQL数据库转储
“# Dumping data for table”	SQL数据库转储
“# phpMyAdmin MySQL-Dump” filetype:txt	由phpMyAdmin创建的SQL数据库转储
“# phpMyAdmin MySQL-Dump” “INSERT INTO” -“the”	由phpMyAdmin创建的SQL数据库转储（可变）

4.4.5 实际的数据库文件

另外一种攻击者用来搜索数据库的方法是直接搜索数据库文件本身。这种技术并不适用于所有的数据库系统，只适用于那些用一个带有特定文件名或扩展名的文件来表示数据库的系统。Google可能不太理解怎样去处理和解释这些文件，而且搜索结果页面上的总结（或者“摘要”）是空白的，Google还会把这种文件列为“无法识别的类型”（unknown type），如图4-11所示。

如果Google不理解一个二进制文件的格式，正如那些用filetype操作符找到的文件，就无法在该文件的内部搜索字符串。这大大限制了有效搜索的选项，让你不得不转而依靠inurl或者site操作符。表4-9列出了一些能够搜索到数据库文件的查询。

表4-9 搜索数据库文件的查询

查 询	说 明
filetype:cfm “cfapplication name” password	ColdFusion源代码
filetype:mdb inurl:users.mdb	Microsoft Access用户数据库
inurl:email filetype:mdb	Microsoft Access E-mail数据库

（续）

查　询	说　明
inurl:backup filetype:mdb	Microsoft Access备份数据库
inurl:forum filetype:mdb	Microsoft Access论坛数据库
inurl:/db/main.mdb	ASP-Nuke数据库
inurl:profiles filetype:mdb	Microsoft Access用户资料数据库
filetype:asp DBQ=" * Server. MapPath("*.mdb")	Microsoft Access数据库连接字符串搜索
allinurl: admin mdb	Microsoft Access管理数据库

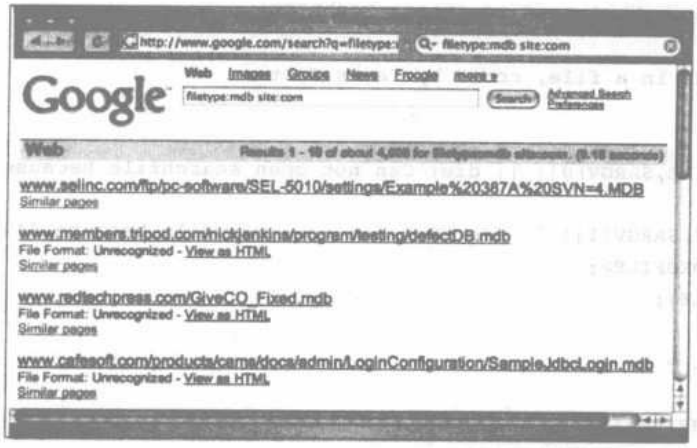


图4-11　Google通常无法识别数据库文件自身

4.5　自动加工

搜索文件是相当简单的，尤其是在你知道了所要查找的文件类型时。我们已经了解了怎样简单地搜索包含敏感数据的文件，但是某些时候可能需要离线搜索文件。例如，假设我们要查找yahoo.com的E-mail地址。例如查询“@yahoo.com” email根本不能作为一个有效的Web搜索，即使作为一个Group搜索，它也是有问题的，如图4-12所示。

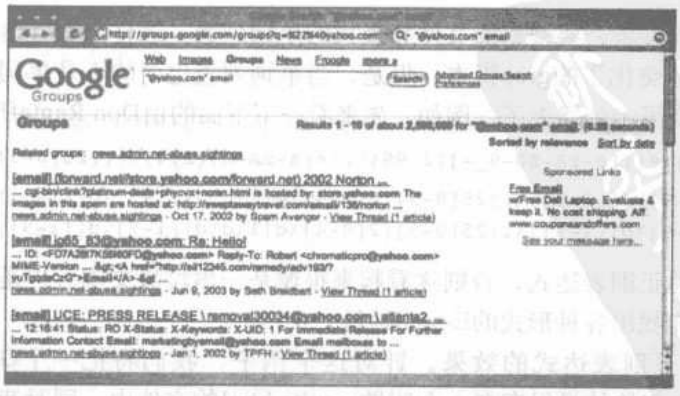


图4-12　一个普通的还有许多待改进的E-mail搜索

这个搜索找到了一个E-mail地址，jg65_83@yahoo.com，但是也搜索到了store.yahoo.com，而这并不是一个有效的E-mail地址。类似于这种情况，最好的用于搜索特定字符串的选择是使用正则表达式（regular expression）。这需要下载你要搜索的文档（可以用Google查询搜索到），并且分析这些文件以获得你要查找的信息。你可以选择让下载这些文件的过程自动执行，我们将在第12章介绍它。但是在你下载完这些文件之后，需要一种简便的方法在文件中搜索有用的信息。我们先来看一下如下的Perl脚本：

```
#!/usr/bin/perl
#
# Usage: ./ssearch.pl FILE_TO_SEARCH WORDLIST
#
# Locate words in a file, coded by James Foster
#
use strict;
open(SEARCHFILE,$ARGV[0]) || die("Can not open searchfile because $!");

open(WORDFILE,$ARGV[1]) || die("Can not open wordfile because $!");
my @WORDS=<WORDFILE>;
close(WORDFILE);

my $LineCount = 0;

while(<SEARCHFILE>) {
   foreach my $word (@WORDS) {
      chomp($word);
      ++$LineCount;
      if(m/$word/) {
         print "$&\n";
         last;
      }
   }
}
close(SEARCHFILE);
```

这个脚本有两个参数：要搜索的文件和查找的单词列表。这样来看，这个程序是相当简单的，实际上只是一个美化了的grep脚本。但是，当单词列表包含的不是单词而是正则表达式的时候，这个脚本就变得十分强大了。例如，先来看一下下面的由Don Ranta所写的正则表达式：

```
[a-zA-Z0-9._-]+@(([a-zA-Z0-9_-]{2,99}\.)+[a-zA-Z]{2,4})|((25[0-5]|2[0-
4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-
4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9]))
```

除非你非常精通正则表达式，否则这看起来很像是一串垃圾文本。但是，这个正则表达式是十分强大的，它能找出各种形式的E-mail地址。

我们看看这个正则表达式的效果。针对这个例子，我们将把一个Google Group查询“@yahoo.com” email的结果保存在一个叫做results.html的文件中，同时我们也把前面的正则

表达式写在一行保存在叫做wordfile.txt的文件中。如图4-13所示，我们可以使用用一种类似于Lynx的程序从命令行提取搜索结果。Lynx是一种常见的基于文本的Web浏览器。也可以用其他的程序来代替Lynx，例如Curl、Netcat、Telnet或者甚至是在一个标准的Web浏览器中执行“另存为”操作。要留意的是，Google的服务大都不赞成使用任何形式的自动处理。实际上，Google更偏向于让你简单地在浏览器中执行搜索，然后手动保存结果。但是，正如我们在前面讨论的那样，如果你遵守服务条款的宗旨，那么请注意不要用过分的自动处理来违反Google的免费搜索服务，这样Google才不会迁怒于你。不管怎么样，许多人最后还是决定严格地遵守服务条款。

回到我们的Google搜索，注意到其中的URL表明我们只提取了前一百条结果，这可以通过num=100参数看出来。同样，我们也可以搜索更多的E-mail地址。在把结果保存到文件results.html中之后，我们就采用ssearch.pl脚本处理results.html文件，以查找放置在wordfile.txt中的E-mail表达式。为了让结果更为精确，我们把输出用”grep yahoo | head -15 | sort -u”再做进一步处理，这样我们最多可以得到15个不同的包含单词yahoo的E-mail地址。最终结果（混乱的）如图4-13所示。

图4-13 用于搜索E-mail地址的ssearch.pl

你可以看到，这些组合命令能够相当有效地挖掘E-mail地址。如果你比较熟悉UNIX命令的话，可能已经注意到几乎不需要使用两条单独的命令。这个完整的处理过程可以很容易地组合为一条命令，即把Perl脚本修改为读取标准的输入，然后把Lynx命令的输出直接送给ssearch.pl脚本来有效地传递results.html文件。不过，以这种方式介绍这些命令开了不负责任的自动处理技术的先例，我们并不推荐这样做。

其他的正则表达式也可以做简化处理。如下的这个同样由Don Randa所写的表达式是用来搜索URL的：

```
[a-zA-Z]{3,4}[sS]?://((([\w\d\-]+\.)+[ a-zA-Z]{2,4})|((25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])))((\?|/)[\w/=+#_~&:;%\-\?\.]*)*
```

这个表达式能够搜索URL和参数，包括组成IP地址或者域名的地址。它非常擅长于处理

Google结果页面，能够返回页面中的所有链接。它不比基于API的方法好，但是它比API的方法更为简单。下面的表达式能够搜索IP地址：

```
(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-
9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]\d|[1-9])
```

我们可以用这样的正则表达式来辅助映射一个目标网络。这些技术不仅能够用来解析HTML页面，实际上也能用于各类文档。但是，要知道许多文件都是二进制形式的，这意味着在搜索它们之前先要将之转化为文本。UNIX的strings命令（通常用strings -8来实现）很适合这个工作，但是不要忘了Google内置了可以翻译多种不同类型文档的功能。如果你查找的是可视的文本，你应该选择使用Google的翻译功能；如果你查找的是不可打印文本，例如元数据，那么你需要先把原始文件下载下来，然后对它进行离线搜索。不管你是怎样实现这些技术的，你现在都应该清楚：当把Google和少许自动处理技术组合在一起使用时，它便能用做一种十分强大的信息收集工具。

4.6 Google桌面搜索

Google桌面（Google Desktop）是一种允许你在自己的本地机器中搜索文件的应用程序，可以在http://desktop.google.com上获得。目前适用于Windows Mac和Linux平台，Google桌面搜索允许用户依据使用的操作系统来搜索众多类型的文件。可以在Mac OS X操作系统中搜索以下的文件类型。

- Gmail消息
- 文本文件(.txt)
- PDF文件
- HTML文件
- Apple Mail和Microsoft Entourage E-mail
- iChat副本
- Microsoft Word、Excel和PowerPoint文档
- 音乐和视频文件
- 地址通讯录
- 系统偏好设置面板
- 文件和文件夹名

Google桌面搜索也可以在Mac OS X操作系统中搜索以下的文件类型。

- Gmail
- Outlook Express
- Word
- Excel
- PowerPoint
- Internet Explorer

- AOL Instant Messenger
- MSN Messenger
- Google Talk
- Netscape Mail/Thunderbird
- Netscape / Firefox / Mozilla
- PDF
- Music
- Video
- Images
- Zip文件

Google桌面搜索提供了许多功能，但是由于它还是beta产品，所以你应该检查Google桌面的网页以了解当前的功能列表。作为一个文档加工工具，你只要简单地从目标服务器上把内容下载下来，然后使用桌面搜索来查找这些文件即可。桌面搜索也可以截取在Internet Explorer 5及更高版本中浏览的网页。这意味着你能经常浏览以前在线浏览过的页面的老版本，即使原始页面已经发生了变化。另外，在安装了桌面搜索之后，任何在Internet Explorer中执行的在线Google搜索同时也会返回在你的本地机器中找到的结果。

4.7 总结

文档加工这一话题甚至可以花上一整本书的篇幅来讨论的。在这一章中，我们仅希望能涉及到这一话题的皮毛。擅长于此道的攻击者（黑帽或白帽）能够获取大量关于目标的信息。在这一章中，我们还讨论了配置文件、日志文件和Office文档的价值，但是显然我们也可以关注许多其他类型的文档。文档加工的关键首先是发现存在于目标服务器上的文档的类型，然后根据搜索结果的数目，精确到那些可能更有用的或者那些可能更相关的文档。根据目标的不同，可以把目标的在线状况，文档类型，许多其他的因素以及各种关键字组合在一起来搜索任何文档。

数据库hacking同样也是一个需要整本书来讨论的话题。但是，显然Google提供的信息比成熟的数据库审计更有用。登录入口，帮助文件和数据库转储均能够提供各种可循环用于审计的信息。在所有从这些信息源发现的信息中，最有效的（和破坏力最大的）可能就是源代码了。源代码提供了数据库结构的内部信息，同时还能泄露在外部评估中无法注意到的缺陷。但是，在许多情况下，需要仔细阅读源代码以判断程序中的缺陷。错误消息同样也能够给攻击者泄露大量的信息。

自动加工允许你针对一些重要的信息采用程序化的手段来搜索许多文档。当把它和极好的Google文档搜索功能结合在一起时，你就能拥有一种非常强大的信息收集武器。

4.8 快速查找解决方案

配置文件

- 配置文件能够向攻击者泄露敏感的信息。

- 尽管配置文件的文件名不同，但是仍可以通过文件扩展名来找到配置文件，如INI、CONF、CONFIG或CFG。

日志文件

- 日志文件也能够泄露敏感的信息，而且这些信息通常比在配置文件中找到的信息更新。
- 日志文件的文件名也各不相同，但是通常可以利用文件扩展名，例如LOG来找到日志文件。

Office文档

- 在许多情况下，Office文档常常是公开发布的。那些不小心发表到公共区域的文档都可能包含敏感的信息。
- 常见的Office文档文件扩展名有PDF、DOC、TXT或XLS。
- 每个文档的内容各不相同，但是像私人的（private），口令（password），备份（backup）或者管理（admin）这样的字符串意味着可能是一个敏感的文档。

数据库挖掘

- 登录入口，尤其是软件制造商提供的默认入口，是可以很容易就搜索到的，也吸引着正在搜寻特定版本或类型的软件的攻击者。单词login、welcome和copyright statements都是查找登录入口的极佳方式。
- 位于服务器端和客户端的软件都有帮助文件。这些文件会泄露关于应用程序配置和用法的信息。
- 不同的错误消息的内容都能用来剖析目标。
- 数据库转储已证明是所有数据库的发现中泄露信息最多的，因为它们包括完整的或者部分的数据库内容。可以通过数据库转储的头部字符串来搜索到，例如“#Dumping data for table”。

4.9 网站链接

- www.filext.com 一个关于文件扩展名信息的不错的资源。
- http://desktop.google.com Google桌面搜索应用程序。
- http://johnny.ihackstuff.com Google Hacking数据库的主页，在其中你可以找到更多的和本章中列出的那些搜索相近的搜索。

4.10 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：我该怎样做才能防止这种信息泄露？

答：要想为了你所负责的网站修补这种问题，首先需要检查所有能用Google搜索找到的位于你的服务器上的文档。确保用Google搜索找到的文档确定是能够公开浏览的。虽然你可能选

择用某种自动工具（参见“保护”内容相关的章节）来扫描你的网站以检查是否有数据库信息泄露，但是最好地防范方法还是在源代码上。应该隔绝外部用户访问你的数据库远程管理工具，应该对默认登录入口进行安全审查并确保已经移除了软件版本信息，而且也应该从公共服务器删除帮助文件。应当对错误消息进行裁剪以确保不泄露敏感的信息，而且应该对所有使用的应用程序执行审查。除此之外，对你的服务器做一些配置，只允许用户下载某些文件类型也是无伤大雅的。和列出不允许的文件类型比起来，列出允许的文件类型更为容易。

问：我比较关心Office文档中敏感的元数据。我能清除这些信息吗？

答：Microsoft提供了解决这一问题的Web页面：http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q223396。另外，也有一些可以自动处理这一问题的程序。其中一种是www.kklsoftware.com的ezClean。

问：许多软件都依靠包含文件来引用外部内容。根据我的理解，在这一章中讨论的包含文件，例如INC文件，也是一个隐患，因为它们经常给程序，而不是给Web访问者泄露敏感的信息。有没有什么方法可以解除包含文件的隐患吗？

答：包含文件确实是一个问题，这是由它们的文件扩展名导致的。如果它们使用了一个像.INC这样的扩展名，那么大多数的服务器都会以文本方式显示这些文件，这会泄露敏感数据。可以考虑防止用户下载.INC文件（或者是任何包含文件所用的扩展名）。服务器的这种修改可以防止浏览器显示这些文件的内容，而后台处理程序仍然可以访问它们。

问：我们的软件使用.INC文件来保存数据库连接字符串。有其他可替代的办法吗？

答：把扩展名改为.PHP，这样其中的内容就不会显示出来了。

问：该怎样避免Google黑客下载我们的某个应用程序数据库呢？

答：阅读该应用程序的文档。一些编写不是很合理的软件定义了一些固定的路径，但是它们可以允许你把文件放在Web服务器的docroot目录之外。

第5章 Google在信息收集框架中扮演的角色

5.1 简介

做黑客的理由多种多样。当我们中的大多数人听到黑客一词时就会联想到计算机和网络安全，但是律师、销售人员和警察从某种本质上来说也是黑客。黑客实际上是一种心境以及一种思考方式，而不是一种物理属性。人们为什么要做黑客呢？有两种动机，不过一个特殊的原因是因为他们想了解街上普通老百姓所不了解的。由此引出另一个动机。知识就是力量——我们可以查看到别人在做什么而不让他们知晓。黑客的主要内容就是对这种对知识的探索，假设我们有Google、一台分布式超级计算机，可以读取所有已知信息，有一个简单的欺骗性用户界面，那么我们只需要等待几秒钟来回答所有的查询。这看起来几乎就像是Google是为黑客量身定做的。

本书的第1版公开了很多黑客（渗透测试人员）常用来获取那些有可能在常规安全评估（例如，查找网络、域、E-mail地址等）帮助自己的信息的技术。在这样一个常规安全测试（或者笔试）的过程中，目标几乎都是破坏安全措施并且访问受限的信息。然而，信息可以通过将相关的信息碎片组装成一个更大的、更完整的信息来获取。当然，这并非适用于所有的信息。笔者在Google上查找你的超级机密的双重加密文档的可能性非常小，但是我们可以打赌我们最终会通过从诸如Google的公共源中收集大量信息来获取它。

如果你在阅读本书，那么你很可能会对信息挖掘感兴趣，并期望通过以有趣的方式使用搜索引擎来获取大多数信息。在本章中，我希望能向你展示有用的、巧妙的方式来达成此目的。

5.2 自动搜索原则

计算机帮助自动执行那些单调乏味的任务。更巧妙的自动化工作甚至可以完成上千个不同人的同时工作所不能完成的工作。但是那些手动不能完成的工作不可能自动化。如果你想编写一个程序来执行某些工作，那么你需要手动完成整个过程，并保证该过程每次都能执行。自动化执行一个有缺陷的过程没有任何意义。一旦手动过程能很顺利地执行，就可以使用算法来将该过程“翻译”成一个计算机程序。

让我们来看一个实例。某用户很喜欢查找包含andrew@syngress.com这种E-mail地址的Web站点。一般是先打开Google网页，然后在输入框中键入E-mail地址。搜索结果如图5-1所示。

可以看到三个不同的站点列出了这个E-mail地址，它们分别是：g.bookpool.com、www.networksecurityarchive.org和book.google.com。搜索者可能会觉得它并非是那个E-mail地址出现的唯一几个站点，并且可能会记得曾经在哪里看到过将此E-mail地址列为andrew at syngress dot com。当该用户将这个搜索放到Google中时，可以得到如图5-2所示的不同的结果。

图5-1 搜索E-mail地址的简单搜索

图5-2 扩展查询

很明显，查询没有用引号，因此会得到一些不正确的结果。添加引号后会得到如图5-3所示的结果。

通过对查询进行不同的定制，现在可以得到一个新的结果：taosecurity.blogspot.com。搜索查询操作有了结果，用户也找到了另一个站点引用。

如果我们将这个过程分成几个部分，可以看到实际上它是由很多不同的步骤组成的。几乎所有的搜索都遵循以下的几个步骤。

- 确定原始搜索关键字。
- 扩展搜索关键字。

- 从数据源获取数据。
- 解析数据。
- 在后期过程中，将数据处理为信息。

接着让我们就这几个部分仔细地分析一下。

图5-3 添加引号的搜索结果

5.2.1 原始搜索关键字

前面所讲实例的目标是查到引用了指定E-mail地址的网页。这个步骤非常直截了当，但是清晰地定义目标大概算是所有搜索中最不相同的部分了。好的搜索不可能得到一个不清晰的目标。在进行自动化搜索时，手动搜索中的原则也同样适用：无用信息输入，无用信息输出。

工具与陷阱

无用信息输入，无用信息输出

计算机的“思考能力”很差，但是很擅长“数字游戏”。不要寄希望于计算机能够代替你思考，因为这样做的话，你会得到十分让你懊恼的结果。“无用信息输入，无用信息输出”简单地陈述了这样一个道理，如果你在一开始便向计算机里输入了无用的信息，那么你也只能从计算机那里得到无用的信息输出。没有经验的搜索引擎用户通常会忽视这一经验原则。

某些情况下，目标需要被打破。对于广义的目标而言尤其如此，例如尝试查找那些在荷兰奶酪工厂工作的人的E-mail地址。在这样的情形下，最少有一个子目标——你需要先明确奶酪工厂。请确认目标已经清晰地定义，接着便要开展一组中心关键字的搜索。在某些情况下，你需要关注某个单一查询的结果以明确地将搜索的方向确定为某个适宜的起始搜索关键字。看到查询的结果后，我通常想：“我从来没有想到我的查询会返回这些结果。如果每次都能自动形成稍有不同的查询，那么我将得到很多有趣的信息。”

最终，唯一能真正地限制你从搜索引擎获得信息的东西便是你自己的想像力，发现何种查询能起作用的最佳途径是你的经验。

5.2.2 扩展搜索关键字

在前面的实例中，用户快速地领会到可以通过将原始查询分解为一组稍有区别的查询来得到更多的结果。扩展搜索关键字是人类的天性，自动化搜索的真正本领就在于它能思考人类的工作过程并且能将该过程翻译成某种算法。通过将搜索的标准形式程序化地更改为很多不同的搜索，我们把自己从手工重复性劳动中解脱出来，更重要的是，从必须牢记所有的扩展技巧的工作中解脱出来。让我们来看一下其中的一小部分扩展技巧。

1. E-mail地址

很多站点都会模糊E-mail地址以迷惑数据挖掘程序。这样做是有很好的理由的：大多数据挖掘程序都会搜索站点以便为垃圾邮件群发者收集E-mail地址。如果你想收到大量的垃圾邮件，不妨在邮件列表中直接张贴出你的E-mail地址而不进行模糊化处理。虽然站点能自动模糊化处理E-mail地址是一件好事，不过它也会让你过上与Web搜索者一样的艰难生活。幸运的是，还有很多方式可以做到这一点，不过，那些垃圾邮件群发者同样知晓这些方法。

在查找E-mail地址时，我们可以使用以下扩展。E-mail地址andrew@syngress.com可以进行如下的扩展。

- andrew at syngress.com
- andrew at syngress dot com
- andrew@syngress dot com
- andrew_at_syngress.com
- andrew_at_syngress dot com
- andrew_at_syngress_dot_com
- andrew@syngress.remove.com
- andrew@_removethis_syngress.com

注意@符号可以写为多种形式，例如：-(at)，_at_或者-at-。点“.”也一样。你也可以看到很多人在E-mail中添加“remove”或者“removethis”。最终，它会演变成二八事件——当执行这些扩展中排在前面的20%的扩展搜索时，你可以找到80%的地址。

到此，你可能会觉得你永远都不可能查找完地址的所有情况（你或许是对的）。不过，隧道的尽头还是有一丝曙光的。Google会忽略搜索中的特定的字符。搜索andrew@syngress.com和“andrew syngress com”会返回相同的结果。@符号以及点都会被忽略。因此当扩展搜索关键字时，不用包括这两个符号，因为你只是在浪费搜索资源。

工具与陷阱

确定E-mail地址

下面是一个用于在E-mail地址存在之时快速确定地址的搜索方式。虽然这可能并不适

用于所有的邮件服务器，但是它可用于大多数的邮件服务器——包括Gmail。让我们来看一下这种方式：

- 步骤1 查找邮件服务器

```
$ host -t mx gmail.com
gmail.com mail is handled by 5 gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt1.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt2.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 50 gsmtp163.google.com.
gmail.com mail is handled by 50 gsmtp183.google.com.
```

- 步骤2 挑选出一个服务器，并且远程登录到端口25

```
$ telnet gmail-smtp-in.l.google.com 25
Trying 64.233.183.27...
Connected to gmail-smtp-in.l.google.com.
Escape character is '^]'.
220 mx.google.com ESMTP d26si15626330nfh
```

- 步骤3 伪装简单邮件传送协议（Simple Mail Transfer Protocol，SMTP）

```
HELO test
250 mx.google.com at your service
MAIL FROM: <test@test.com>
250 2.1.0 OK
```

- 步骤4a 正面测试

```
RCPT TO: <roelof.temmingh@gmail.com>
250 2.1.5 OK
```

- 步骤4b 负面测试

```
RCPT TO: <kosie.kramer@gmail.com>
550 5.1.1 No such user d26si15626330nfh
```

- 步骤5 完成

```
quit
221 2.0.0 mx.google.com closing connection d26si15626330nfh
```

通过查看来自邮件服务器的响应我们现在可以肯定roelof.temmingh@gmail.com存在，而kosie.kramer@gmail.com不存在。我们可以以同样的方式确定其他E-mail地址的存在。

注意

在Windows平台上你需要使用nslookup命令查找某一个域内的E-mail服务器。你可以这样来查找：nslookup -Cqtype=mx gmail.com

2. 电话号码

相对于E-mail地址有一组确定的格式而言，电话号码就不尽相同了。似乎没有记录电话号

码的标准方式。让我们假设你有一个南非的电话号码，该号码为012 555 1234。该号码可能会以各种不同的形式出现。

- 012 555 1234（本地）
- 012 5551234（本地）
- 012555124（本地）
- +27 12 555 1234（带国家区号）
- +27 12 5551234（带国家区号）
- +27 (0)12 555 1234（带国家区号）
- 0027 (0)12 555 1234（带国家区号）

获取所有结果的一种方式就是查看数据最有效的部分“555 1234”和“5551234”。不过，这种方式有一个缺陷——你可能会发现很多完全不同的国家都有相同的号码，所以查询提供了误报。

通过Google的数字范围（numrange）操作符在一个特定的范围内查找包含的电话号码是一种很有趣的方式。这种搜索的方式就是先指定起始数字，接着键入“..”，再键入终止数字。让我们来看一下在现实生活中这种方式是如何工作的。假设我要查看搜索带区号的电话号码+1 252 793有哪些结果。你可以使用数字范围操作符来限定该查询，如图5-4所示。

图5-4 查找电话号码范围

我们可以清楚地看到这些结果全部包含位于北卡罗莱纳州中的指定范围的数字。我们将在本章看到将结果限制在一个特定区域的功能有多么的强大。

3. 人

查找某人信息的最佳方式就是Google他。如果你没有在Google上搜索过自己，那么你就是一个另类。有很多搜索人的方式，而且它们大都很简单明了。如果你不能马上搜到结果也没关系，还有很多种选择。假定要查找Andrew Williams，你可能使用以下搜索：

- "Andrew Williams"
- "Williams Andrew"
- "A Williams"
- "Andrew W"
- Andrew Williams
- Williams Andrew

注意，最后两个搜索没有使用引号。这会查到类似于"Andrew is part of the Williams family"的短语。

因为互联网上可能有很多名为Andrew William的人，所以搜索名字Andrew William肯定会得到大量的误报。正因为如此，你需要添加尽可能多的附加搜索关键字。例如，你可以尝试这样的搜索："Andrew Williams" Syngress publishing security。另一个减少误报的技巧就是将站点限制在一个固定的国家。如果Andrew住在英国，那么添加site:uk操作符会有助于得到更精确的结果。但是要记住的是，接下来的搜索将限定在英国。如果Andrew真的来自英国，但是公布的站点的终端位于其他的顶级域（TLD），那么搜索不可能返回来自这些站点的结果。

4. 获取大量结果

有些时候，你可能更希望获取大量的结果，而不是某些特定的结果。例如，你需要查找位于特定TLD中的所有Web站点或者E-mail地址。这时，你需要将搜索与能够完成以下两件事的关键字结合起来：超出1000个结果的限制，并且提升每次搜索的结果的数量。让我们来看一下如图5-5所示的在****.gov域中查找Web站点的实例。

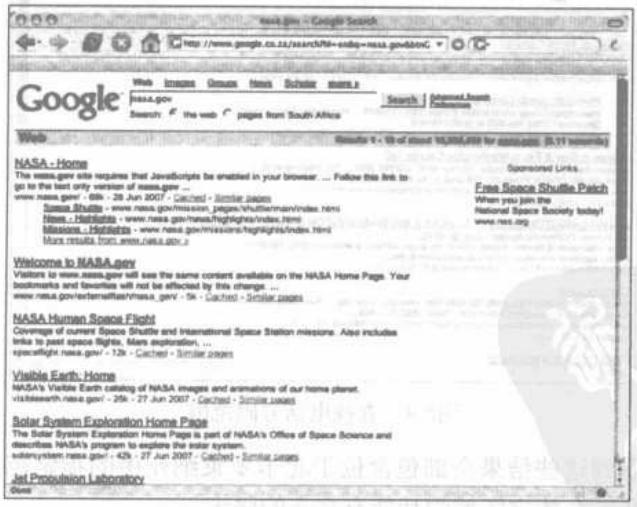


图5-5 搜索一个域

你将从该查询中搜到1000多个站点，因为很可能从一个站点搜索出多个结果。换句话说，如果500个页面位于一个服务器，500个页面位于另一个服务器，你将只能得到两个站点结果。同样，你也会从那些不在****.gov域中的站点中搜索到结果。如何获得更多的结果并且将搜索

限制在****.gov域中呢？那就是结合查询与关键字以及其他操作符。来看一下这个查询site:****.gov - www.****.gov。该查询意味着可以在****.gov域中的站点内查找任何结果，但是这些结果都不位于它们的主站点中。尽管查询能很好地工作，但是它也只能获得最多1000个结果。我们可以在各个查询上添加几个常用的关键字。在此，我们的搜索主旨是使用那些能将少于1000个结果的站点增加到最初的1000个结果的单词。尽管无法保证可以排除其他站点，我们还是可以考虑添加关键字如about、official、page、site等。尽管Google称诸如the、a、or等单词会在搜索中被忽略掉，但是我们仍然能看到当我们将这些单词与site:操作符结合起来时得到的结果不同。图5-6所示的结果说明Google确实会重视查询中的这些“会被忽略的”单词。

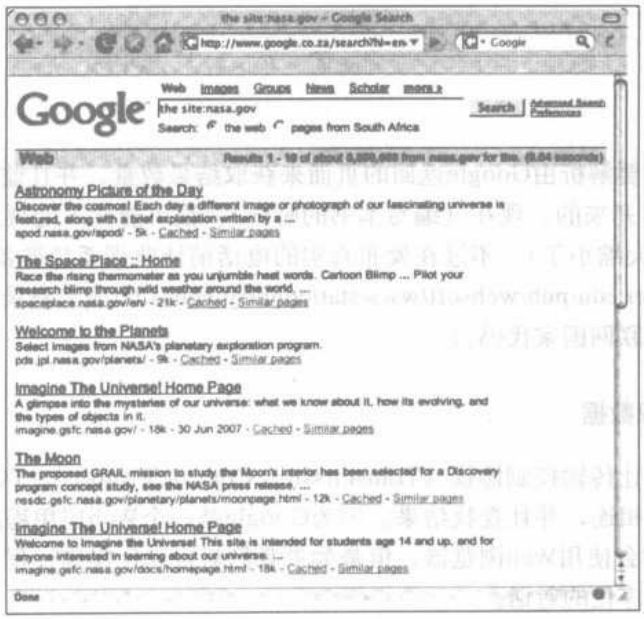


图5-6 使用site操作符搜索域

5. 更多组合

在把搜索主旨确定为查找大量结果时，你可能需要将搜索与那些可以搜索更理想结果的关键字相结合。例如，在查找E-mail地址时，可以添加关键字如contact、mail、e-mail、send等。当查找电话号码时，则可以添加诸额外的关键字如phone、telephone、contact、number、mobile等。

6. 使用“特殊的”操作符

基于要从Google中查找的内容，我们可能会需要使用一些其他操作符。假设我们要查看某个Web站点上有哪些Microsoft Office文档。我们知道可以使用filetype:操作符来指定一种特定的文件类型，但是每次查询只能指定一种类型。因此，我们需要自动化每次查询各种Office文件类型的过程。不妨询问Goolge以下几个问题：

- filetype:ppt site:www.****.gov
- filetype:doc site:www.****.gov
- filetype:xls site:www.****.gov

- filetype:pdf site:www.****.gov

要牢记的是在特定的情况下，这些扩展可以使用布尔逻辑符再次结合。在搜索Office文档时，filetype:ppt或者filetype:doc site www.****.gov的搜索效果也很好。

记住，我们可以将site:操作符更改为site:****.gov，后者可以在****.gov域中的任意Web站点中搜索结果。我们还可以以其他方式使用site:操作符。假设有一个能在不同国家的各大站点里查找单词iPone出现的频率的程序。如果要查看的是荷兰、法国、德国、比利时以及瑞士，那么可以将查询扩展为以下形式：

- iphone site:nl
- iphone site:fr
- iphone site:de
- iphone site:be
- iphone site:ch

此时，我们仅需要解析由Google返回的页面来获取结果数量，并且监测iPhone活动是如何慢慢地经过西欧扩展开来的。现在（编写本书的时候），很可能你不能得到一个有意义的结果（因为该活动已经大大缩小了），不过在发布真实的电话前让监视系统准备待绪会更有用。（可以登录http://ftp.ics.uci.edu/pub/websoft/wwwstat/country-codes.txt或者登录Google来搜索所有国家代码列表以查看互联网国家代码。）

5.2.3 从数据源获取数据

至少，我们需要让传输控制协议（Transmission Control Protocol，TCP）与我们的数据源（Google Web站点）相连，并且查找结果。因为Google是一个Web应用程序，所以我们要连接端口80。通常，我们会使用Web浏览器，但是如果我们对自动化操作过程很感兴起，那么则需要能与Google进行程序化的对话。

1. 自行抓取——请求并接收响应

这是获取结果的另一种灵活的方式。你完全可以控制整个过程，并且可以进行诸如设置结果数量这类的工作——使用应用编程接口（Application Programming Interface，API）永远也不可能做到这一点。但是这也是劳动力强度最密集的工作。不过，一旦你进行了该工作，你的顾虑就被完全打消了，并且会开始着手调节参数。

警告

大多数Web应用程序都不允许抓取（scraping）。Google在使用条款（Terms of Use，TOU）中也不允许抓取，除非你已经通过这些条款的批准。以下摘自www.google.com/accounts/TOS：

“5.3你同意不以任何除了通过Google提供的界面外的方式访问（或者尝试访问）任何服务，除非你已经与Google签署了单独的协议，被明文允许访问。你明确同意不通过任何自动方式（包括使用脚本或者Web爬行者）来访问（或者尝试访问）任何服务，并且你保证你会遵守服务提出的任何robots.txt文件里列出的使用说明。”

在开始前，我们需要找出向Web站点提问或者提出查询的方式。如果你像平常那样向Google询问一些事（如本例中的单词test），那么返回的URL将会显示如下：

```
http://www.google.co.za/search?hl=en&q=test&btnG=Search&meta=
```

第一个斜杠之后的那一部分比较有用：search?hl=en&q=test&btnG=Search&meta=。这是一个使用&分隔开来的GET请求、参数以及值。在这个请求中，我们传递了4个参数：

- hl
- q
- btnG
- meta

这些参数的值与参数之间采用等号符号分隔。hl参数指的是“home language（母语）”，这里被设置为English（英语）。q参数指的是“question（问题）”或者“query（查询）”，该参数被设置为我们的查询“test”。其他两个参数不太重要（至少现在不是太重要）。我们的查询将返回10个结果。如果我们将首选项设置为返回100个结果，那么我们可以得到如下的GET请求：

```
http://www.google.co.za/search?num=100&hl=en&q=test&btnG=Search&meta=
```

请注意此处的附加参数；num被设为100。如果我们请求结果的第2个页面（例如结果101－200），请求语句如下：

```
http://www.google.co.za/search?q=test&num=100&hl=en&start=100&sa=N
```

此处要注意两点。参数传递的顺序，以及添加了start参数。start参数告诉Google从哪个页面开始获取结果，num参数则告诉它我们希望得到多少结果。因此，如此推理，要想得到301－400的结果，我们应该采用类似于以下请求：

```
http://www.google.co.za/search?q=test&num=100&hl=en&start=300&sa=N
```

让我们尝试输入该请求，并看看能得到什么，如图5-7所示。

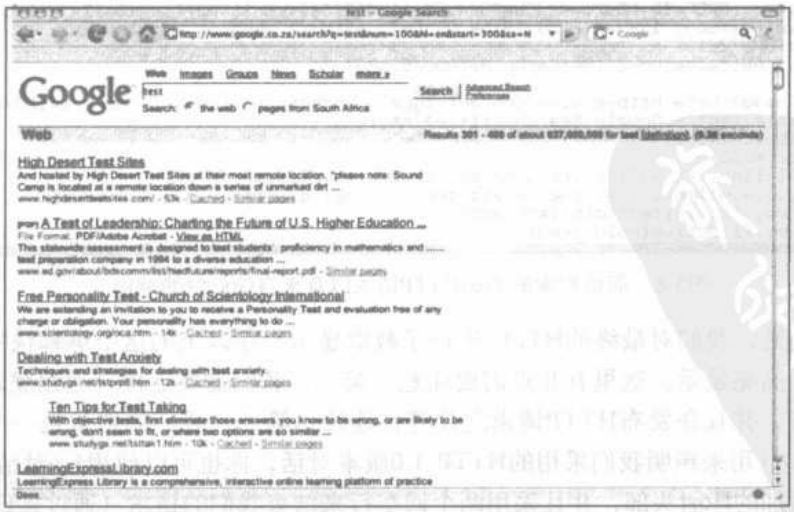


图5-7 从第3页搜索100个结果

这看上去不错。假设我们正在查找一些更为复杂的东西，此时又会发生什么情况呢。搜索“testing testing 123” site:uk在如下的查询中会生成如下查询：

```
http://www.google.co.za/search?num=100&hl=en&q=%22testing+testing+123%22+site%3A
uk&btnG=Search&meta=
```

此处发生了什么？让我们来分析一下。num参数被设置成100。btnG和meta参数可以被忽略。site操作符不会生成额外的参数，但是位于提问或者查询之中。这个提问是%22testing+testing+123%22+site%3Auk。实际上%22就是引号（“）的十六进制代码形式。%3A是冒号（:）的代码形式。一旦我们使用非代码形式来替代这些代码符号，我们就可以还原到原始的查询：“testing testing 123” site:uk。

因此，当对字符进行编码时，以及当你要使用未编码形式时，你会做何许判定？这是它自己的事，但是根据经验，在编码那些不在A－Z，a－z，0－9范围内的字符时，你不可能犯错。编码可以程序化地执行，但是如果你对此很感兴趣了，你可以通过在UNIX终端上键入“man ASCII”，通过在Google上查找ASCII十六进制编码或者通过访问http://en.wikipedia.org/wiki/ASCII来查看所有的编码字符。

既然我们已经知道如何公式化阐述请求了，不妨让我们一起准备将该请求发送到Google，并获取一个应答。注意，服务器将会以HTML的方式回答。在这个最简单的形式里，我们可以直接远程登录到Google的Web服务器，并且手动发送请求。图5-8说明了该工作方式：

```
Mips:~ roeloftemmingh$ telnet www.google.com 80
Trying 64.233.183.103...
Connected to www.l.google.com.
Escape character is '^]'.
GET /search?hl=en&q=test&btnG=Search&meta= HTTP/1.0
Host: www.google.com

HTTP/1.0 200 OK
Date: Mon, 02 Jul 2007 11:55:47 GMT
Content-Type: text/html; charset=ISO-8859-1
Cache-Control: private
Set-Cookie: PREF=ID=65d2ba4ed6bd9544:TM=1183377347:LM=1183377347:S=T2xjyi3xSSKmD
cdR; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Server: GWS/2.1
Via: 1.1 netcachejhb-2 (NetCache NetApp/5.5R6)

<html><head><meta http-equiv="content-type" content="text/html; charset=ISO-8859
-1"><title>test - Google Search</title><style><!--
div,td{color:#000}
.f{color:#666}
.flc,.fl:link,.ft a:link,.ft a:hover,.ft a:active{color:#77c}
a:link,.w,a.w:link,.w a:link,.q:visited,.q:link,.q:active,.q{color:#00c}
a:visited,.fl:visited{color:#551a8b}
a:active,.fl:active{color:red}
```

图5-8 简单搜索的原始HTTP请求以及来自Google的响应

为了简洁起见，我们对最终的HTML进行了截取显示。在以上的这个屏幕快照中，打印输出的命令采用了高亮显示。这里有几点需要注意。第一，我们需要连接（远程登录）到端口号为80的Web站点，并且在发布HTTP请求之前等待连接。第二，我们的请求是一个GET，它后面跟着HTTP/1.0（用来声明我们采用的HTTP 1.0版本对话，你也可以使用1.1对话）。最后一件事是注意我们添加的Host头部，并且采用两个回车行来结束我们的请求（通过按Enter键两次）。服务器使用HTTP头部（两个回车行之前的部分）以及一个包含了HTML实体的程序体（以

<html>开始的部分）来应答。

这看似很繁琐，但是既然我们知道了请求的内容，便可以基于它来生成自动操作了。让我们一起来尝试使用Netcat来完成此工作。

背景知识

Netcat

人们称Netcat是TCP/Internet Protocol (IP)的瑞士军刀。对于黑白两道而言，它都是一个极其有用的工具；从漏洞利用获取反解的shell（黑帽）到帮助网络管理员分析协议（白帽）。在本例中，我们将使用它来向Google的Web服务器发送一个请求，并且在屏幕上显示最终的HTML。你可以通过Google搜索“netcat download”（netcat下载）来获取UNIX以及Microsoft Windows的Netcat版本。

本章将不再讲述Netcat的各种转变以及使用。我们只使用Netcat来向Google发送请求，并且获取响应。在将Netcat程式化之前，不妨先来看一下如下的命令以及它们对应的输出。

```
$ echo "GET / HTTP/1.0";echo "Host: www.google.com"; echo
GET / HTTP/1.0
Host: www.google.com
```

注意，最后的echo命令（后面空无一物的那个）在HTTP请求的后面添加了必要的回车行(CRLF)。为了将它挂到Netcat上，并且将它与Google站点相连，我们可以进行以下操作：

```
$ (echo "GET / HTTP/1.0";echo "Host: www.google.com";echo)|nc www.google.com 80
```

该命令将得到如下的输出：

```
HTTP/1.0 302 Found
Date: Mon, 02 Jul 2007 12:56:55 GMT
Content-Length: 221
Content-Type: text/html
```

为了简洁起见，我们截掉了输出的后部分内容。注意，我们在echo命令的两侧添加了括号()，并且使用管道符号（|）来将命令挂到Netcat之上。Netcat将从端口80连接到www.google.com，并且将管道符号左侧的输出命令发送到服务器。将Netcat和echo一起挂起的特殊方式可以在UNIX中使用，但是如果Windows下使用它就需要进行一些改动了。

还有其他（更容易）的方式可以获得相同的结果。先来看一下wget命令（可以访问httP://xoomer.alice.it/hherold/来获取wget的Windows版本）。wget是一个很强大工具，使用它向Web服务器发送请求就有点像是让火箭专家维修微波炉一样简单。你只需简单地键入wget ®Ch即可查看wget能做的所有工作。如果我们想使用wget来获取查询结果，可以使用以下形式：

```
wget http://www.google.co.za/search?hl=en&q=test -O output
```

输出结果如下：

```
--15:41:43-- http://www.google.com/search?hl=en&q=test
=> `output'
Resolving www.google.com... 64.233.183.103, 64.233.183.104, 64.233.183.147, ...
```

```
Connecting to www.google.com|64.233.183.103|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
15:41:44 ERROR 403: Forbidden.
```

该命令的输出是Google不太关注自动化过程的首要暗示。什么地方出问题了呢？HTTP请求的头部有一个名为“User-Agent”域。该域由请求Web页面的应用程序（通常是浏览器，也可能是诸如wget的“强盗”）来填写，该域被用来识别浏览器或者程序。wget生成的HTTP头部如下：

```
GET /search?hl=en&q=test HTTP/1.0
User-Agent: Wget/1.10.1
Accept: */*
Host: www.google.com
Connection: Keep-Alive
```

你可以看到User-Agent被填写为Wget/1.10.1。这有一个问题。Google会检查头部中的该域，并且判定你是否正在使用可以用来自动处理的工具。Google不喜欢自动搜索查询，而且会返回在HTTP错误代码403，Forbidden（禁用）。幸运的是，这并不是世界的末路。因为wget是一个灵活的程序，你可以设置它将如何在User Agent域中介绍自己。因此，我们所需要做的一切就是告诉wget要将它自己介绍成非wget的其他的东西。你只需要进行一些辅助的转换就可以轻易地做到此。让我们来看一下，当我们告诉wget要把自己介绍为“my_diesel_driven_browser”时，头部的内容应该如何编写。我们假设命令的内容如下：

```
$ wget -U my_diesel_drive_browser "http://www.google.com/search?hl=en&q=test"-O output
```

合成后的HTTP请求的头部如下：

```
GET /search?hl=en&q=test HTTP/1.0
User-Agent: my_diesel_drive_browser
Accept: */*
Host: www.google.com
Connection: Keep-Alive
```

注意已经更改后的User-Agent。现在，命令的输出形式如下：

```
--15:48:55-- http://www.google.com/search?hl=en&q=test => 'output'
Resolving www.google.com... 64.233.183.147, 64.233.183.99, 64.233.183.103, ...
Connecting to www.google.com|64.233.183.147|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
[ <=> ] 17,913 37.65K/s
15:48:56 (37.63 KB/s) - 'output' saved [17913]
```

位于该文件中的该查询的HTML称为‘output’。该实例了一个非常重要的概念——更改User-Agent。Google有一个巨大的不被允许的User-Agent列表。

自动化Web请求的另一个流行程序被称为curl，你可以通过访问http://fileforum.betanews.com/detail/cURL_for_Windows/966899018/1来攻取它的Windows版本。针对安全套接字层（Secure Sockets Layer，SSL）的使用，你可能需要从其他地方获取文件libssl32.dll。请在同一

目录里保存EXE和DLL文件。与wget一样的是，你需要设置User-Agent才能使用它。默认的curl行为将会把来自查询的HTML直接返回到标准输出。下面是一个使用带有可选User-Agent的curl从一个简单的查询返回HTML的实例。命令如下所示：

```
$ curl -A zoemzoemspecial "http://www.google.com/search?hl=en&q=test"
```

该命令的输出是未经过处理的HTML响应。注意已经更改的User-Agent。

Google也会使用Lynx基于文本的浏览器的用户代理，该浏览器会尝试编译HTML，而不需要用户直接与HTML打交道。对于诸如从查询获取大量结果的快速黑客行为而言，这种做法非常有用。先来看一下如下命令：

```
$ lynx -dump "http://www.google.com/search?q=google" | grep Results | awk -F "of
about" '{print $2}' | awk '{print $1}'
1,020,000,000
```

显然，使用诸如sed、grep、awk等的UNIX命令使得在紧要关头使用带转储参数的Lynx成为了必然的选择。

有很多其他的命令行工具可以被用来向Web服务器发送请求。本章将不再列出这些不同的工具清单。大多数情况下，你都需要更改User-Agent以便能够与Google进行对话。你也可以使用你喜欢的编程语言来自行编译请求，并且使用套接字连接到Google。

2. 自行抓取——解析结果

在上一节中，我们学到如何向Google提问，以及如何从服务器获取HTML。因为它还算有用，所以如果我们仅以HTML堆结束的话，它就不可能真的那么有用了。为了了解HTML的意义，我们需要能获取单个的结果。在每次抓取操作中，获取单个的结果都是任务的最杂乱的部分。解析结果的第一步就是查看是否有一个结果返回的结构。如果存在一种结构，那么我们可以从该结构中将数据解析为不同的结果。

来自FireFox的FireBug扩展（https://addons.mozilla.org/en-US/firefox/addon/1843）可以被用来很容易地将HTML代码映射为可视化结构。在FireFox中查看Google结果页面以及在FireBug中查看的部分结果的效果如图5-9所示。

在FireBug中，每个结果摘要均以HTML代码 开始。记住这一点后，我们便可以以一个非常简单PERL脚本开始，该脚本将仅提取摘要的第一部分。先来看一下如下的代码：

```
1 #!/bin/perl
2 use strict;
3 my $result=`curl -A moo "http://www.google.co.za/search?q=test&hl=en"`;
4 my $start=index($result,"<div class=g>");
5 my $end=index($result,"<div class=g",$start+1);
6 my $snippet=substr($result,$start,$end-$start);
7 print "\n\n".$snippet."\n\n";
```

在脚本的第3行，我们从外部调用curl来将一个简单请求的结果放入到$result变量中（提问/查询是test，并且我们获取了第一个10条结果）。在第4行中，我们创建了一个标量（$start），该标量包含了“ ”令牌第一次出现的位置。在第5行中，我们看到了令牌的第二

次出现，看到了摘要的结尾（这同时也是第二个摘要的开始）以及我们把位置赋给了$end。在第6行中，我们将会从整个HTML代码块中把第一个摘要剪切出来，在第7行中，我们将输出并显示它。让我们看一看它是如何工作的：

```
$ perl easy.pl
% Total % Received % Xferd Average Speed Time Time Time Current
                                Dload Upload Total Spent Left Speed
100 14367 0 14367 0   0 13141 0 --:--:-- 0:00:01 --:--:-- 54754

<div class=g><a href="http://www.test.com/" class=l><b>Test</b>.com Web Based
Testing Software</a><table border=0 cellpadding=0 cellspacing=0><tr><td
class="j"><font size=-1>Provides extranet privacy to clients making a range of
<b>tests</b> and surveys available to their human resources departments. Companies
can <b>test</b> prospective and <b>...</b><br><span class=a>www.<b>test</b>.com/ -
28k - </span><nobr><a class=fl
href="http://64.233.183.104/search?q=cache:S9XHtkEncW8J:www.test.com/+test&hl=en&ct
=clnk&cd=1&gl=za&ie=UTF-8">Cached</a> - <a class=fl href="/search?hl=en&ie=UTF-
8&q=related:www.test.com/">Similar pages</a></nobr></font></td></tr></table></div>
```

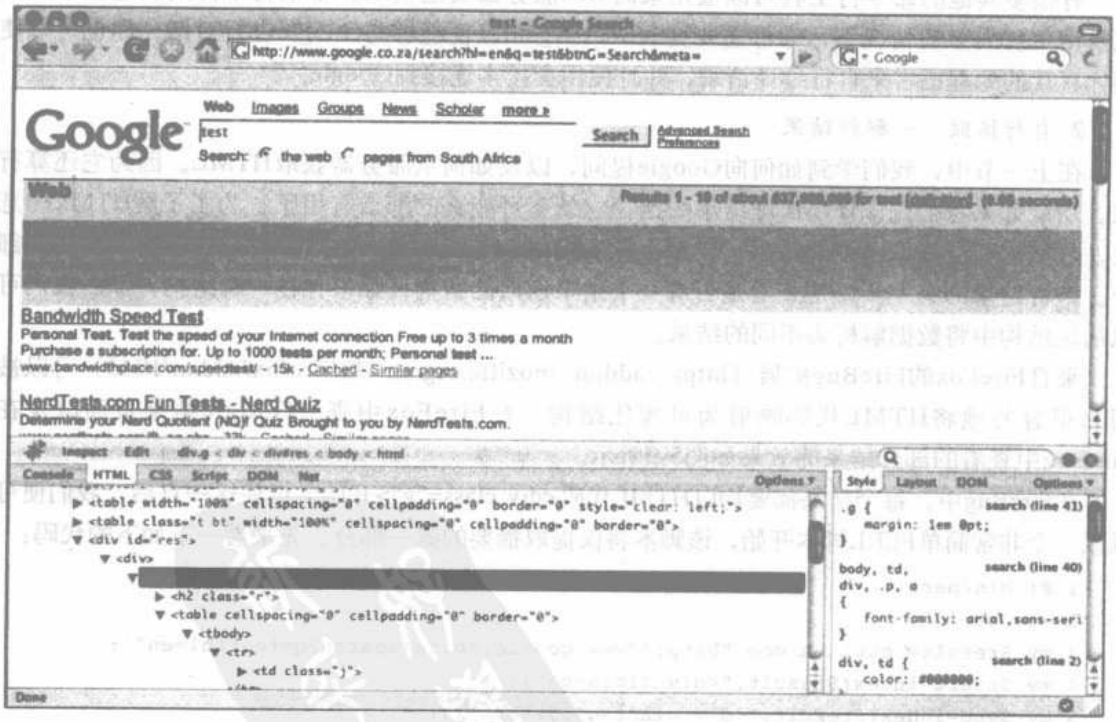


图5-9 使用FireBug查看Google搜索结果

当我们将它与浏览器显示的内容相对比时，它看上去是对的。现在，该脚本需要以某种方式完成整个HTML，并且提取所有的摘要。先来看一下以下PERL脚本：

```
1 #!/bin/perl
2 use strict;
```

```
3 my $result=`curl -A moo "http://www.google.com/search?q=test&hl=en"`;
4
5 my $start;
6 my $end;
7 my $token="<div class=g>";
8
9 while (1){
10 $start=index($result,$token,$start);
11 $end=index($result,$token,$start+1);
12 if ($start == -1 || $end == -1 || $start == $end){
13  last;
14 }
15
16 my $snippet=substr($result,$start,$end-$start);
17 print "\n-----\n".$snippet."\n----\n";
18 $start=$end;
19 }
```

虽然脚本有一点复杂，但是它实际上仍然是很简单的。因为我们要多次使用“ ”，所以在这个脚本里，我们将它放入一个令牌中。这样一来，当Google因其他事宜调用它时更改会变得容易。在第9行到第19行中，我们创建了一个循环，该循环将持续查看令牌的存在，直到它再也没有找到。如果它没有找到令牌（第12行），然后循环就只是存在而已。在第18行中，我们将该位置从我们开始搜索（令牌）的位置移至我们要结束先前搜索的位置。

运行该脚本后会生成要被发送到标准输出的不同的HTML摘要。但是它最好是如此有用。我们真正需要的是从摘要中提取URL、标题和总结。对于此，我们需要一个接收4个参数的函数：一个包含起始令牌的字符串，一个包含结束令牌的字符串，一个用来说明从何处查找的标量以及一个包含我们要在其中搜索的HTML的字符串。我们要让该函数返回提取的部分，以及我们在传递的字符串中放入的新位置。这样的函数的结构应该如下所示：

```
1 sub cutter{
2   my ($starttok,$endtok,$where,$str)=@_;
3   my $startcut=index($str,$starttok,$where)+length($starttok);
4   my $endcut=index($str,$endtok,$startcut+1);
5   my $returner=substr($str,$startcut,$endcut-$startcut);
6   my @res;
7   push @res,$endcut;
8 push @res,$returner;
9 return @res;
10 }
```

现在，我们有了这么一个函数，我们可以检查HTML，并且决定如何从每个摘要中提取URL、总结和标题。执行此操作的代码需要被定位在主循环中，并且结构如下所示。

```
1 my ($pos,$url) = cutter("<a href=\"","\"",0,$snippet);
2 my ($pos,$heading) = cutter(">","</a>",$pos,$snippet);
3 my ($pos,$summary) = cutter("<font size=-1>","<br>",$pos,$snippet);
```

注意，为什么URL是我们在摘要中遇到的第一件事。URL本身是超级链接，而且通常以“”并以“”结束。最后出现的是总结（summary），它通常以“<font size=-1>”开始，以“
”结尾。将它们放在一起我们可以得到如下的PERL脚本：

```
#!/bin/perl
use strict;
my $result=`curl -A moo "http://www.google.com/search?q=test&hl=en"`;

my $start;
my $end;
my $token="<div class=g>";

while (1){
  $start=index($result,$token,$start);
  $end=index($result,$token,$start+1);
  if ($start == -1 || $end == -1 || $start == $end){
    last;
  }
  my $snippet=substr($result,$start,$end-$start);
  my ($pos,$url) = cutter("<a href=\"","\"",0,$snippet);
  my ($pos,$heading) = cutter(">","</a>",$pos,$snippet);
  my ($pos,$summary) = cutter("<font size=-1>","<br>",$pos,$snippet);

  # remove <b> and </b>
  $heading=cleanB($heading);
  $url=cleanB($url);
  $summary=cleanB($summary);

  print "--->\nURL: $url\nHeading: $heading\nSummary:$summary\n<---\n\n";
  $start=$end;
}

sub cutter{
  my ($starttok,$endtok,$where,$str)=@_;
  my $startcut=index($str,$starttok,$where)+length($starttok);
  my $endcut=index($str,$endtok,$startcut+1);
  my $returner=substr($str,$startcut,$endcut-$startcut);
  my @res;
  push @res,$endcut;
  push @res,$returner;
  return @res;
}

sub cleanB{
  my ($str)=@_;
```

```
    $str=~s/<b>//g;
    $str=~s/<\/b>//g;
    return $str;
}
```

注意，Google会在结果中高亮显示搜索关键字。因此，我们可以在结果里删除“cleanB”子程序中的和。让我们来看一下该脚本是如何工作的，如图5-10所示。

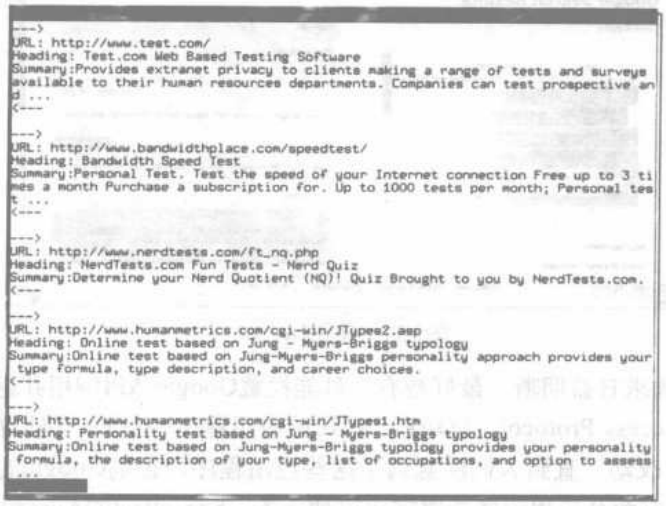


图5-10 运行中的PERL脚本

它似乎运行得不错。如果第一次它运行得不理想，只需通过调试以及优化便能达到更好的效果。

3. Dapper

虽然手动抓取是获取结果的最灵活的方式，但是它仍旧是一件非常艰辛和杂乱的工作。因此，我们肯定还有更简单的方式。Dapper站点（www.dapper.net）支持用户创建一种他们称之为Dapp的东西。这些Dapp是很小的程序，可以从任何站点抓取信息，并将这些抓取的数据以几乎任意的一种格式传递（例如，XML、CSV、RSS等）。Dapper的优点在于编写Dapp很便利，可以通过可视化的界面编写。虽然Dapper在抓取不少站点都表现得不错，但是它却不可能像我们期盼的Google搜索的方式那样工作。不同人创建的也会返回不同的结果。Dapper很有前景，应该引起人们的重视与研究，如图5-11所示。

4. Aura/EvilAPI

Google过去常常会提供一个允许用户与Google引擎程序化对话的API。首先，你需要签约该服务，并且接收一个密钥。你可以将该密钥与其他参数一起传递给某个Web服务，该Web服务将会以良好的XML结果的形式返回数据。每天，这个标准的密钥都常常会用上近1,000次。过去很多工具都使用该API——现在也仍旧有一些工具使用该API。这种方式过去的工作量非常之大，不过，自2006年12月5日起，Google已经不再签发新的API密钥了。旧的密钥仍旧可用，并且API仍旧保留在原处（它知道密钥工作了多长时间），不过，新用户将不能读取它。现在，

Google提供了一个很有趣的AJAX界面，但是已经不允许从脚本或者应用程序来自动搜索了（并且它的某些主要的特性已经丧失了）。并不是所有的特性都已经丧失。

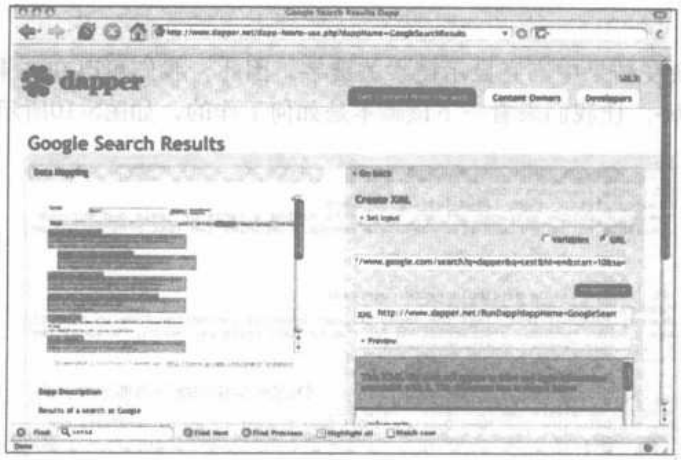


图5-11 使用Dapper

对API替代品的需求日益明晰。最好是有一种能拦截Google API调用并返回简单对象存取协议（Simple Object Access Protocol，SOAP）XML的应用程序——基于仍旧在使用的API的应用程序不需要进行任何改动。直到人们注意到了这些应用程序，否则Google的终端不会出现任何变化。谢天谢地，还有两个应用程序能够完成这种任务：SensePost的Aura和Sitening的EvilAPI。

将EvilAPI（http://sitening.com/evilapi/h）作为PERL脚本安装在Web服务器上。接着，定义了Web服务提供的功能（以及在何处查找该功能）的GoogleSearch.wsdl文件应该被修改为指向你的Web服务器。

在费力获取了Web服务器上运行的PERL脚本后（设想这有两个版本的PERL），Sitening提供了一个可供你测试API脚本的测试网关。再次修改WSDL文件并指向它们的站点后，开始运行示例脚本。Sitening看上去仍旧像是不打算开工的样子。因为“Google经常会把它们记入黑名单”，所以它们的网关看起来“几乎就像是有故障一样”。“基于PERL的脚本代码与本章前面列举的代码是如此的像，以至于自行抓取看起来要比费劲地让所有的代码正常运行起来要更容易。尽管如此，如果你有大量的Google API信任的合法代码，你还是应该尝试使用Sitening。

SensePost的Aura（www.sensepost.com/research/aura）是另一个执行同样功能的代理。目前，它只在Windows（以.NET编码）上运行，但是据SensePost的内部消息称，很快Java版本的也将发布了。该代理通过改变主机表的方式工作，因此api.google.com指向本地机。接着，该代理将解析向Web服务发出的请求，并替代你执行抓取。当前，Aura与本地主机绑定在一起（换句话说，就是它不支持外部连接），但是我们坚信Java版本将支持外部连接。假设在Windows中，示例代码通过Aura无法工作，同样也无法通过一个来自UNIX机器的中继连接工作。那么眼下，示例代码的完整性就要得到质疑。但是，当使用旧的API密钥测试时，它运行得十分完好。最后，我们不得不Aura来测试Wikto的Googler部分，谢天谢地，该连接工作做得很棒。

API兼容产品的底线就是按预定的计划使用它们时，它们确实能很好地工作。但是本地设

计的脚本则需要更多的关注。注意，当你可以在不费力的情况下手动抓取站点，则不要花太多的时间来使用兼容产品。手动抓取的方式也很灵活。

5. 使用其他搜索引擎

不管你是否相信，除了Google外，世界上还有很多搜索引擎！MSN引擎也支持API，并且很值得一用。既然本书的书名没有被定为“MSN Hacking与渗透测试”，那么我们就将MSN API留做读者的课后习题。

5.2.4 解析数据

假设目前所有的东西都已经各就其位，处于要连接到数据源（本例中是Google）的状态，我们正在询问正确的问题，并且已经将那些会告诉我们的结果的东西以简洁的纯文本写出。现在，我们不打算过多地担忧它出现的方式。它可以使用代理API获取，可以自行抓取获取，或者从某些提供者那获取。本节将只讲解你可以利用返回数据处理的事情。

为了进入最佳的状态，不妨先来扪心自问作为一个人，你要使用结果来做什么。你可以要扫描结果来查找E-mail地址、Web站点、域、电话号码、地址、名字和姓氏。作为一个人，你也能将某些上下文放入结果中。这里的观念是我们可以把人类的逻辑放到程序中。另外，计算机很擅长做重复的工作，而且不会感到疲惫和厌烦，或者要求加薪。一旦我们将这些逻辑分类整理好，我们便可以添加诸如计算获取的每个结果有多少个、确定提问对应的结果的可信度是多少以及返回数据与源提问有多少的关联这样的其他有趣的事情。

1. 解析E-mail地址

有很多种从纯文本中解析E-mail地址的方式，其中大多数方式基于正则表达式。正则表达式就像你不情愿与之交谈的古怪的叔叔，但是当你深入了解他，便会觉得他很酷很有趣。你害怕正则表达式也是情理之中的事，不过了解一点关于它的情况可以让你的生活更简单。如果你是一个正则表达式的宗师级人物，你也许能创建一个oneliner regex来从纯文本中有效地解析E-mail，不过要是只有我了解的话就很糟糕了，我们要放轻松并且尽量只使用基本实例。让我们看一下我们如何在PERL程序中使用它。

```
use strict;
my $to_parse="This is a test for roelof\@home.paterva.com - yeah right blah";
my @words;
#convert to lower case
$to_parse =~ tr/A-Z/a-z/;

#cut at word boundaries
push @words,split(/ /,$to_parse);

foreach my $word (@words){
  if ($word =~ /[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}/) {
    print $word."\n";
  }
}
```

看起来一切都很顺利，但是现实生活中可能有一些问题。脚本基于单词之间的空格将文本分割成了多个单词。但是，如果文本是“Is your address roelof@paterva.com”又会怎么样呢？脚本失效了。如果我们将@符号、下划线（_）以及短划线（-）转换成字母记号，接着删除所有的符号，并且将字母记号转换为源值，程序就又起效了。让我们来看下一段代码：

```
use strict;
my $to_parse="Hey !! Is this a test for roelof-temmingh\@home.paterva.com? Right
!";
my @words;

print "Before: $to_parse\n";
#convert to lower case
$to_parse =~ tr/A-Z/a-z/;

#convert 'special' chars to tokens
$to_parse=convert_xtoX($to_parse);
#blot all symbols
$to_parse=~s/\W/ /g;
#convert back
$to_parse=convert_Xtox($to_parse);
print "After: $to_parse\n";

#cut at word boundaries
push @words,split(/ /,$to_parse);

print "\nParsed email addresses follows:\n";
foreach my $word (@words){
  if ($word =~ /[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}/) {
    print $word."\n";
  }
}

sub convert_xtoX {
  my ($work)=@_;
  $work =~ s/\@/AT/g; $work =~ s/\./DOT/g;
  $work =~ s/_/UNSC/g; $work =~ s/-/DASH/g;
  return $work;
}

sub convert_Xtox{
  my ($work)=@_;
  $work =~ s/AT/\@/g; $work =~ s/DOT/\./g;
  $work =~ s/UNSC/_/g; $work =~ s/DASH/-/g;
  return $work;
}

Right -let's see how this works.
```

```
$ perl parse-email-2.pl
Before: Hey !! Is this a test for roelof-temmingh@home.paterva.com? Right !
After: hey is this a test for roelof-temmingh@home.paterva.com right

Parsed email addresses follows:
roelof-temmingh@home.paterva.com
```

一切顺利，但是代码仍会遇到失效的情况。如果代码读到了“My e-mail address is roelof@paterva.com”会怎样？注意到E-mail地址后的句点了吗？解析后的地址仍会保留句点。幸运的话，也可以使用一个简单的替代法则来修复，将点空格顺序更改为两个空格。在PERL中有：

```
$to_parse =~ s/\. / /g;
```

有了这一句代码，我们解析有效的E-mail地址（以及大约5%的无效地址）的效率将高达99%。诚然，该脚本不够优雅、完美和赏心，但是它十分有效！

你是否还记得我们在前些小节里如何处理E-mail地址的扩展名吗？我们现在需要进行完全相反的操作。也就是说，如果我们找到“andrew at syngress.com”文本，我们需要了解它是否真是一个E-mail地址。这样做的弊端是我们会导致误报。设想有这么一段文本“you can contact us at paterva.com”。如果我们将at转换回@，我们将会解析一个全称为reads us@paterva.com的E-mail。但是也许喜大于忧，通常，搜索到的真实的E-mail地址要多于假的E-mail地址。（这也基于域。如果域属于一个公司，通常在它们的名字后面添加.com，例如amazon.com。在你获取某些有意义的东西之前，你还可能会获得一些误报。）此外，我们还要搜索到那些包括_remove_或removethis记号。

在PERL中做这些事情简直是小菜一碟。我们仅需要在解析程序前添加这些“翻译”即可：

```
sub expand_ats{
  my ($work)=@_;
  $work=~s/remove//g;
  $work=~s/removethis//g;
  $work=~s/_remove_//g;
  $work=~s/\(remove\)//g;
  $work=~s/_removethis_//g;
  $work=~s/\s*(\@)\s*/\@/g;
  $work=~s/\s+at\s+/\@/g;
  $work=~s/\s*\(at\)\s*/\@/g;
  $work=~s/\s*\[at\]\s*/\@/g;
  $work=~s/\s*\.at\.\s*/\@/g;
  $work=~s/\s*_at_\s*/\@/g;
  $work=~s/\s*\@\s*/\@/g;
  $work=~s/\s*dot\s*/\./g;
  $work=~s/\s*\[dot\]\s*/\./g;
  $work=~s/\s*\(dot\)\s*/\./g;
  $work=~s/\s*_dot_\s*/\./g;
  $work=~s/\s*\.\s*/\./g;
  return $work;
}
```

这些替换被绑定来捕获大量E-mail地址，但是也可能会导致误报。让我们先来运行一下程序，并且查看它如何使用测试数据。

```
$ perl parse-email-3.pl
Before: Testing test1 at paterva.com
This is normal text. For a dot matrix printer.
This is normal text...no really it is!
At work we all need to work hard
  test2@paterva dot com
  test3 _at_ paterva dot com
  test4(remove) (at) paterva [dot] com
  roelof @ paterva . com
  I want to stay at home. Really I do
```

之后：测试test1@paterva.com this is normal text.for a.matrix printer.this is normal text...no really it is @work we all need to work hard test2@paterva.com test3@paterva.com test4 @paterva . com roelof@paterva.com i want to stay@home.really i do。可得：

```
Parsed email addresses follows:
test1@paterva.com
test2@paterva.com
test3@paterva.com
roelof@paterva.com
stay@home.really
```

因为运行了测试，你可以看到程序捕获了5个测试E-mail地址中的4个，并且包括了一个误报。基于该程序，误报的比率已经可以为我们所接受，因为使用可视化查看即可快速地注意到它们。另外，二八法则也适用于此；你只需要20%的努力即可捕获80%的E-mail地址。如果你乐于做一些后期处理，你应该查看你查找到的该E-mail地址是否以一些不知名的TLD（参见下节）结尾。但是，通常如果你要捕获所有的E-mail地址（以所有不明晰的格式出现），你便可以清楚地了解到是花大把的力气来查找还是选择处理大量的误报。

2. 域和子域

幸运的是，如果你愿意做一些假设的话，域和子域要更容易解析一些。主机名和域名之间有什么区别呢？你如何区别这两部分呢？这看起来像是一个愚蠢的问题。很明显，www.paterva.com是一个主机名，paterva.com是一个域，因为www.paerva.com有一个IP地址，而paterva.com没有。但是google.com域（以及很多其他的域）也会解析为一个IP地址。而且你知道google.com是一个域。如果我们从fpd.gsfc.****.gov处获得了一个Google链接又怎么样呢？它是主机名还是域呢？抑或某些其他的CNAME？你也许本能地会在名字前添加www.，并查看它是会被解析为一个IP地址。如果这样可行的话，它便是一个域。但是如果域中没有www条目会怎样呢？答案是什么？

域中需要一个名称服务器条目。主机名就不必有名称服务器条目了，实际上它很少有名称服务器条目。如果我们做此假设，便可以将域和主机分辨开来了。剩下的看起来就很容易了。我们只需将Google URL域按句点分割成几块，并将它还原在一起即可。让我们来看一个实例：

站点fpd.gsfc.****.gov。它没有名称服务器，因此我们可以安全地忽略fpd部分，并采用gsfc.****.gov结束。从这里我们可以获取域：

- gsfc.****.gov****.gov
- gov

这里有我们想要的更多的事。具有代表性的是，我们对TLD甚至是子TLD不感兴趣。如果你想更轻松地将它们筛选出来（TLD和子TLD列表可以通过访问www.neuhaus.com/domaincheck/domain_list.htm来获取）。在查找域时，我们还可以做另一件有趣的事。我们可以使用找到的任何新的信息递归地调用脚本。域搜索脚本的输入通常应该是一个域，对吗？如果我们将域****.gov输入到脚本中，我们就会仅限于得到1,000个结果。如果我们的脚本找到域gsfc.****.gov，便可以将它反馈回同一脚本，以便在它的子域（该子域可以给我们更深层次的子域）中找到另1000个新的结果。最后，在再也找不到任何新的子域时，便可以终结脚本运行了。

另一个肯定可行的获取域而不需要执行主机/域的检查的方式就是来后期处理挖掘所得的E-mail地址。因为几乎所有的E-mail地址都位于域上（而非主机上），所以E-mail地址可以简单地从@符号后截断，并用于类似的样式。

3. 电话号码

电话号码的解析很难维持在一个可接受的误报率之内。这是因为没有记录电话号码的标准方法。一些人会在号码中添加国家区号，但是在地方站点中（或者邮寄列表中）却很少添加国家区号。即使添加了国家区号，也是通过使用加号（例如，+44）或者使用本地国际拨号方式（例如，0044）来添加的。这变得更糟了。在很多情况下，如果城市区号以0开头，那么在添加国家区号时就会将这个零忽略（例如，+27 12 555 1234与012 555 1234相对）。然而又有一些人把零放在圆括号中以表示，当从国外拨打电话时，不要拨它（例如，+27 (0)12 555 1234）。更糟糕的是，很多欧洲国家喜欢将最后4个阿拉伯数字分成两组（例如，012 12 555 12 34）。当然，这就是那些以特定方式记忆数字的人们，他们破坏了所有的格式，并且让人几乎无法判断出哪一部分是国家区号、哪一部分是城市区号，以及这是在城市的哪一个区域（例如，+271 25 551 234）

幸运的是，日期与电话号码大不相同。先来看一段这样的文本“From 1823-1825 1520 people couldn't parse telephone numbers”。更好的情况恐怕要数诸如“Andrew Williams: 1971-04-01 - 2007-07-07”这样的时间范围了。既然人们在处理E-mail地址时发现误报并不那么困难，所以你需要成为一名当地居民来区分布隆迪的水管工的电话号码以及盗窃网络的ISBN号。那么，是否所有的力气都白费了？不尽然。有两个解决方案：一个较难实现但是花费较低，另一个容易实际但是要付出较大代价。在前而那个较难实现但是花费较低的方案中，我们将会应用所有我们可能想到的电话号码的逻辑，并且要忍耐误报。在另一个容易（它也并非有那么容易）的方案中，我们可以从商家那里购入国家、城市和地区区号。让我们先来看一下第一种较难实际的解决方案。

自动化最强大的原理就是如果你可以想出一种类似于人类工作的工作方式，那么你就一定可以用代码实现它。在你编写不出你要做的工作之时，就是自动化无法实现之时。如果我们可以将了解到的电话号码的所有情况编写为一个算法，那么我们就有了正确执行它的方法了。以

下是一些我通常用来判断数字是否是一个真正的电话号码的规则。

- 当且仅当数字以+开头时，方才将00转换为+。
- 删除(0)的实例。
- 字长在9到13之间。
- 至少要有一个空格（低容差时可选）。
- 不能包含两个（或更多的）单个数字（例如，2383 5 3 231要被清除）。
- 不能看起来像日期（多种格式）。
- 数字里带有加号情况只能出现在数字字首。
- 第1个空格前要少于4个数字（除非它以+或者0开头）。
- 邻近的位置中没有ISBN字样的字符串。
- 从最后一个数字反推到最开始的那个数字，并将它们以+××××××−××××−××××格式写出。

要想找出那些与以上规则的数字并非易事。我没有使用正则表达式而是使用一个嵌套循环来完成此工作。嵌套循环依次计算数字和可接受符号（加号、短划线以及空格）的个数。一旦它满足这个条件：后面跟着大量不可接受的符号的特定数量的可接受字符，结果被发送到校验程序（该校验程序运用了上面列出的所有规则）。如果得到证实，结果就会被重新打包来尝试获取正确的格式。

当然，这种方式也不常有效。事实上，近20%的结果都是误报。但是该方法很少能错过发现真正电话号码的机会，而且更重要的是，它并不需要花什么钱。

还有更多好办法来做到此。如果我们有一个列出了所有国家和城市区号的列表，我们就应该能总结出格式以及证实一个数字序列是否真的是电话号码。这样的列表确实出在，不过并不是公开的，它无法通过公共域来获取。图5-12中显示的就是一个示例数据库的截屏（CSV）：

图5-12 电话的城市和地区区号示例

如果它是手机号或者地区号以及城市名的话，那么我们不但得到了号码，也知道了国家、供应商。图5-12中的数字来自西班牙，而且有6个数字长。我们现在需要查看列表中的哪些数字与我们解析所得的数字最匹配。因为我没有完整的数据库，所以我没有该数据库的区号，但是我猜想你将会需要编写一个通过比对解析得到的数字与列表中的数字来测定第一对数字之间的差距的程序。因为同样的数字可能是多个国家的号码，并且如果在Web页面上列举它们的时

候也没有显示国家区号，那就不可能判定出它们位于哪个国家。

可以访问www.numberingplans.com来购买数据库，但是将数据库卖给任何人的程序都相当严格。他们也会提供一个漂亮的查看界面（限制为每天一组查看），界面不只用来查看电话号码。不过，明天还有机会。

5.2.5 后期处理

即使我们从我们的数据源得到了有效的数据，仍需要对数据进行某些后期处理。也许你需要统计出你找到的每个结果的数量，以便能定期地对数据进行分类。在下一节，我们将来看一下你应该考虑做哪些事。

1. 按相关性对结果进行分类

如果在我们搜索“Andrew Williams”时需要解析一个E-mail地址，该E-mail地址肯定要比我们搜索“A Williams”时获得的E-mail地址要有趣一些。实际上，我们在前面小节中进行的部分扩展搜索几乎没有什么效果。因此，我们需要的是一个实现搜索“置信度”的方法。这实际上并不没有那么难。只需要简单地将该置信度指数指定给你解析的每个结果。

还有其他一些获取最相关的结果以列在结果列表顶部的方法。另一种方法就是简单地查看结果出现的频率。如果你解析的E-mail地址andrew@syngress.com比任何其他的E-mail地址的10倍还多，那么很可能是这个E-mail地址比那些只出现两次的E-mail地址要更相关。

还有一个方法是查看结果与源搜索关键字的关联性有多大。结果andrew@syngress.com看起来非常像Andrew Williams的E-mail地址。写出关联类型的算法很困难。像这样的关联程序实例的内容可能如下：

```
sub correlate{
  my ($org,$test)=@_;
  print " [$org] to [$test] : ";
  my $tester; my $beingtest;
  my $multi=1;
  #determine which is the longer string
  if (length($org) > length($test)){
    $tester=$org; $beingtest=$test;
  } else {
    $tester=$test; $beingtest=$org;
  }
  #loop for every 3 letters
  for (my $index=0; $index<=length($tester)-3; $index++){
    my $threeletters=substr($tester,$index,3);
    if ($beingtest =~ /$threeletters/i){
      $multi=$multi*2;
    }
  }
  print "$multi\n";
  return $multi;
}
```

该程序将两个字符串中的较长的那个分割成几个由三个字母组成的部分，并且将这些部分与其他（更短的）字符串相比较。对于每个匹配的部分而言，结果返回值都将翻倍。这绝对不是一个“标准的”相关函数，但是却可以达到预期的效果，因为基本上我们所需要的东西是可以识别出E-mail地址各个看上去与名或者姓相似的部分。让我们来运行一下，并且简单地查看一下它是如何工作的。在此，我们将“权衡”一下以下E-mail地址的结果以及“Roelof Temmingh”的最初搜索结果：

```
[Roelof Temmingh] to [roelof.temmingh@abc.co.za] : 8192
[Roelof Temmingh] to [rtemmingh@abc.co.za] : 64
[Roelof Temmingh] to [roeloft@abc.co.za] : 16
[Roelof Temmingh] to [TemmiRoe882@abc.co.za] : 16
[Roelof Temmingh] to [kosie@temmingh.org] : 64
[Roelof Temmingh] to [kosie.kramer@yahoo.com] : 1
[Roelof Temmingh] to [Tempest@yahoo.com] : 2
```

它们看起来工作得不错啊，将第1个地址评为最佳的，两个包含完整的姓的地址排在第二位。看到算法不知道哪个是用户名、哪个是域真是一件有意思的事。你可能应该通过简单地在@符号处将E-mail地址分开并且仅比较前面的部分来更改。从另一方面来看，查看那些看起来像是名或姓的域也是很有趣的事。

有两种权衡结果的方法。第一种方法是查看源搜索关键字与结果页面上的解析结果之间的差距。换句话来说，如果E-mail地址紧接着搜索关键字出现，那么很有可能的是它比远离搜索关键字20段开外的E-mail地址更相关。第二种方法是通过查看那些给出结果的重要（或者流行）站点。这意味着来自某个更为流行的站点的结果要比那些来自Google搜索结果第5页上的结果要更相关。幸运的是，通过只查看Google结果，我们可以轻易地实现这两个必要的条件。Google摘要仅包含围绕在我们查找的关键字周围的文本，因此我们保证有一些相似（除非解析的结果与解析的结果通过“...”分隔）。站点的重要性或者流行程度可以通过站点评级（Pagerank）来获取。通过将值赋予基于结果中的位置的站点（例如，如果站点出现在结果中的首位或者更靠后一些），我们可以获取站点重要性的一个相当近似的值。

此处要注意一点。这些不同的因素需要被小心地平衡。事情朝着坏方向发展真的非常快。假设Anrew的E-mail地址是whipmaster@midgets.com，但是来自E-mail地址中的记录是他通常使用的化名“WhipMaster”。刚开始，与原始关键字关联的搜索（假定我们搜索的是Andrew Williams）并不会得到一个空值。同样的，我们可以选择只为每个匹配的三个字母单词提高10%的索引率，而使用代码的话则只会以100%的提高。但是这正是自动操作的特性，也是为什么这些工具类型最终只是辅助而非替代人类的原因。

2. 非摘要内容

这是另一种我们可以做的后期处理类型，但是它会占用大量的带宽以及处理工作量。如果我们将挖掘的工作扩展到实际返回页面（例如，不只是摘要），我们可能得到更多的结果，并能够做其他一些有趣的事情。这里的想法是从Google结果获取URL，下载整个页面，将它转换为纯文本（尽我们最大的努力），并且对文本执行挖掘算法。在某些情况下，该扩展值得努力（想像一下查找E-mail地址并且查找一个包含一个雇员及他们的E-mail地址列表的页面。好大一

个金矿！）。当仅在摘要中查找某些单词和短语时，它也允许解析单词和短语。

对整个页面中的单词或者短语进行解析和分类最好留给专家来进行（比如Google方面的博士），但是没有人否认我们可以进行某些基本的处理。开始，我们将查找所有页面的单词频率。我们将使用最常用的普通单词（例如，the、and和friends）来结束。我们可以使用很多提供了特定语言中的最常用的十个单词列表中的某一个来筛选出这些单词。最后的文本会为我们提供所有页面中哪些单词是最常用的一般观念；换句话说，是“这是关于什么”的理念。我们可以通过简单地将这些单词连接起来扩充为短语。下一步将是查看那些在单一页面中使用频率不是很高的、但是当在多个页面中查找时会有较高频率的单词或者短语。换句话说，我们查找的是那些仅在一个文档（或者Web页面）中使用一次或者两次的单词，但是这些单词会在所有不同的页面中出现。此处的观念是这些单词或者短语会给出有关该主题的特定信息。

3. 呈现结果

因为很多搜索都会使用扩展，因此，得到了很多搜索以及很多我们需要最后将所有子结果合并到单一结果的Google页面的抓取。通常，它会是一个结果列表，我们需要通过结果的相关性来对结果进行分类。

5.2.6 数据挖掘的应用

1. 略有一点兴趣

让我们来看一下一些可以用来查找E-mail地址的基本挖掘。在我们进行更有趣的实例之前，让我们先来看一下，所有不同的抓取/解析/权衡技术是否能真正地工作。位于ww.paterva.com之上的Evolution的Web界面通常可以实现前面提及的所有技术（以及一些其他的神秘的商业秘密）。让我们先来看一下Evolution是如何工作的。

一开始，我们需要决定前要查看哪类实体（事物）。假定我们在查找Andrew Williams的E-mail地址，我们需要将类型设置为“Person”，并且将函数（或者换算）设置为“toEmailGoogle”，因为我们要让Evolution在Google上为Andrew搜索E-mail地址。点击搜索提交按钮之前的状态如图5-13所示。

图5-13 Evolution准备就绪

点击提交按钮之后便会显示如图5-14所示的结果。

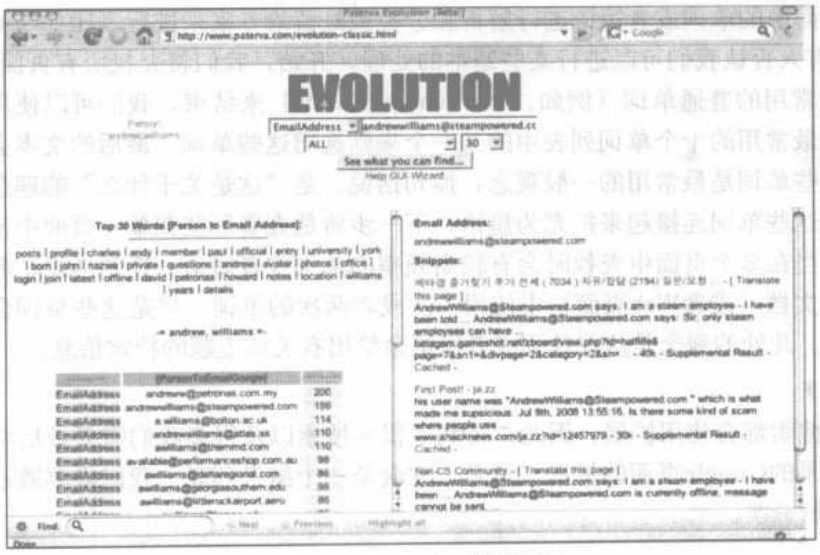


图5-14 Evolution结果页面

这里有几点需要注意的事情。第一是Evolution针对该查询为我们提交了结果页面中查找到的排行在前30位的单词。第二是结果将按照它们间的关联索引来分类，并且当你把鼠标移到它们的上方时，系统会给出相关的摘要：它能在哪找到以及相应地填充搜索框。最后，你应该注意到，这里没有Andrew在Syngress的地址任何相关信息，而只是告诉你互联网上提及过不只一个Andrew Williams。为了更精确地查找在Syngress工作的Andrew Williams，我们可以添加一个额外的搜索关键字。可以通过添加另一个逗号，并指定一个额外的关键字。因此，它变成了“Andrew, Williams, syngress”。结果如图5-15所示。

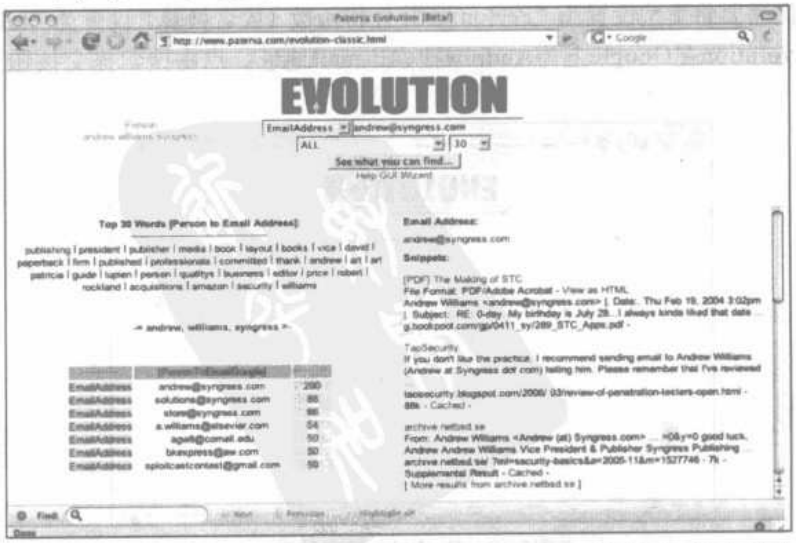


图5-15 当添加一个额外的搜索关键字后，Evolution可以搜到更好的结果

有趣的是，需要注意的是这里有三种不同的Andrew的E-mail地址编码（也就是说，andrew@syngress.com、Andrew at Syngress dot com以及Andrew (at) Syngress.com）。他在Elsevier的另一个E-mail地址也被找出来了。

假定我们想要查找某个特定域（例如，****.gov）中的大量地址。我们会将类型设置为“Domain（域）”，键入域****.gov，将结果设为100，并且选择“ToEmailAtDomain”。最终，位于****.gov域上的所有可用E-mail地址将会如图5-16所示。

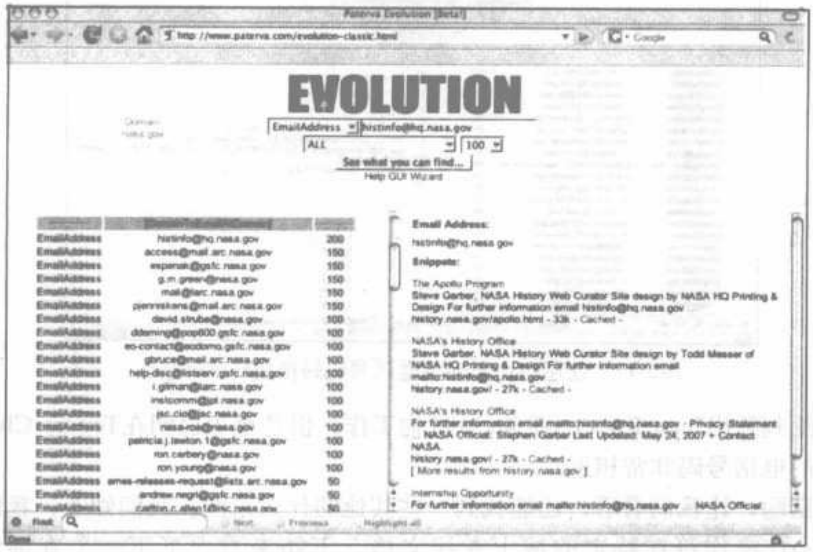


图5-16 使用Evolution挖掘E-mail地址

当鼠标移至结果的上方，界面会自动地针对下一条搜索做好准备（例如，更新类型和值）。图5-16展示的是预载了前一个搜索的结果的界面。

我们可以通过相似的方法来使用Evolution获取电话号码。这些号码可以是大批号码也可以是特定号码。这完全取决于它的使用方式。

2. 最感兴趣

到目前为止，我们采用的实例都很乏味。让我们一同来查看一些三字母的联邦机构证券，以给实例添加一些乐趣。你不要认为在×××.gov（我们用来代替机构的名字）工作的那些类似间谍般的人会列出他们的E-mail地址。让我们来看一下我们可以使用工具找到些什么内容。我们可以通过在×××.gov域上搜索以及查看我们可以从此解析到什么电话号码来开始查找。在Evolution中我们提交了域×××.gov，并且将换算设置为“ToPhoneGoogle”。结果看起来不是非常的好，但是通过查看地区区号以及城市区号，我们了解到一组以703 444开头的号码。这是我们经常用来掩盖机构真实名字的一个仿制的扩展实例，但是这些号码确实与真实机构的Web站点相关。这是一个很好的开始。在任何情况下，我们都不能肯定整个分机都属于他们，但是我们可以来试一下。同样地，我们需要搜索以703 444开头的电话号码，并且解析与这些号码相关的E-mail地址、电话号码以及站点名。我们寄希望于某位类似间谍般的人在列出他的

私人E-mail地址的同时也会列出他的办公电话。我们可以通过将实体（Entity）类型设置为“Telephone”、键入“+1 703 444”（忽略电话号码的最后4位）、将结果设置为100，并且使用组合“ToEmailPhoneSiteGoogle”来着手。结果如图5-17所示。

图5-17 使用Evolution将电话号码转换成E-mail地址

这虽然无法判断出Jean Roberts在为xxx机构工作，但是至少说明在Tennis Club中列出的电话号码与机构的电话号码非常相近。

继续这个话题，让我们看看，还能找到一些其他的什么东西。我们知道，我们可以通过设置filetype和site操作符在特定的域上查找文档。不妨来看一下下一个查询filetype:doc site:xxx.gov，如图5-18所示。

图5-18 查找某个域上的文档

虽然结果页面中列出的文档并不那么令人激动，但是文档中的元信息可能有用。ServerSniff.net站点提供了一个很有用的页面，该页面上有很多可以用来分析有趣的元数据的文

档（www.serversniff.net/fileinfo.php）。

一旦你进行更深入地了解，它就会变得真的很有趣了。在Evolution中完成几次点击之后，你可以在www.clator.com上找到Clator Butler Web Consulting，并且了解Clator Butler先生是David Wilcox（艺术家）论坛的管理员。当在Evolution上找到“Clator Butler”并且将换算设置为“ToAffiLinkedIn”时，我们找到了如图5-19所示的Clator Butler上的LinkedIn的分析。

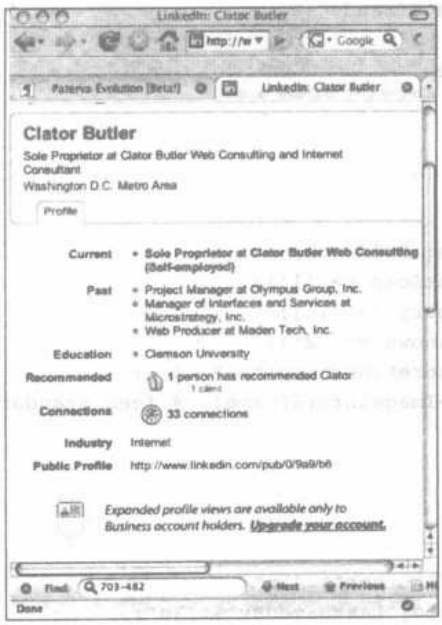


图5-19 官方文档作者的LinkedIn的分析

提取文档和分析文档的过程可以实现自动化吗？当然！刚开始时，我们可以创建一个用来查找Office文档（(.doc，.ppt，.xls，.pps）的URL的scraper（抓取程序）。接着，我们需要下载文档，并且让它完成元信息解析程序。最后，我们可以提取有趣的位，并对它做一些后期处理。我们已经有一个抓取程序（参见前面的小节），因此，我们只需要一些会从文件中提取元信息的东西。ServerSniff.net的Thomas Springer非常友善，他为我提供了文档的信息脚本源。经过细微的改变后，脚本如下：

```
#!/usr/bin/perl

# File-analyzer 0.1, 07/08/2007, thomas springer
# stripped-down version
# slightly modified by roelof temmingh @ paterva.com
# this code is public domain - use at own risk
# this code is using phil harveys ExifTool - THANK YOU, PHIL!!!!
# http://www.ebv4linux.de/images/articles/Phil1.jpg
use strict;
use Image::ExifTool;
```

```
#passed parameter is a URL
my ($url)=@ARGV;

# get file and make a nice filename
my $file=get_page($url);
my $time=time;
my $frand=rand(10000);
my $fname="/tmp/".$time.$frand;

# write stuff to a file
 open(FL, ">$fname");
 print FL $file;
 close(FL);

# Get EXIF-INFO
 my $exifTool=new Image::ExifTool;
 $exifTool->Options(FastScan => '1');
 $exifTool->Options(Binary => '1');
 $exifTool->Options(Unknown => '2');
 $exifTool->Options(IgnoreMinorErrors => '1');
 my $info = $exifTool->ImageInfo($fname); # feed standard info into a hash

# delete tempfile
unlink ("$fname");

my @names;
print "Author:".$$info{"Author"}."\n";
print "LastSaved:".$$info{"LastSavedBy"}."\n";
print "Creator:".$$info{"creator"}."\n";
print "Company:".$$info{"Company"}."\n";
print "Email:".$$info{"AuthorEmail"}."\n";

exit; #comment to see more fields

foreach (keys %$info){
   print "$_ = $$info{$_}\n";
}
sub get_page{
   my ($url)=@_;
   #use curl to get it - you might want change this
   # 25 second timeout - also modify as you see fit
   my $res=`curl -s -m 25 $url`;
   return $res;
}
```

将该脚本保存为docinfo.pl。你将会注意到为了使用该脚本，你需要使用一些PERL库，特别是Image::ExifTool库，该库通常被用来从文件中获取元数据。该脚本使用curl来从服务器下载页面，因此你也需要curl。curl被设置为25秒超时。在一个较慢的链接中，你也许需要将它设置得更高一些。让我们来看一下该脚本如何工作：

```
$ perl docinfo.pl http://www.elsevier.com/framework_support/permreq.doc
Author:Catherine Nielsen
LastSaved:Administrator
Creator:
Company:Elsevier Science
Email:
```

该脚本在文档中查找5个域：Author、LastedSavedBy、Creator、Company和AuthorEmail。这还有很多其他的有趣的域（诸如被用来创建文档的软件）。就它自身而言，该脚本只是稍微有趣而已，当将它与抓取程序结合起来，并对结果做一些后期处理时，它就会变得非常强大。让我们来稍加修改该现成的抓取程序，得到如下所示的新程序：

```
#!/usr/bin/perl
use strict;

my ($domain,$num)=@ARGV;
my @types=("doc","xls","ppt","pps");
my $result;
foreach my $type (@types){
  $result=`curl -s -A moo
"http://www.google.com/search?q=filetype:$type+site:$domain&hl=en&
num=$num&filter=0"`;
  parse($result);
}
sub parse {
  ($result)=@_;
  my $start;
  my $end;
  my $token="<div class=g>";

my $count=1;
while (1){
  $start=index($result,$token,$start);
  $end=index($result,$token,$start+1);
  if ($start == -1 || $end == -1 || $start == $end){
    last;
  }

  my $snippet=substr($result,$start,$end-$start);
  my ($pos,$url) = cutter("<a href=\"","\"",0,$snippet);
  my ($pos,$heading) = cutter(">","</a>",$pos,$snippet);
  my ($pos,$summary) = cutter("<font size=-1>","<br>",$pos,$snippet);

  # remove <b> and </b>
  $heading=cleanB($heading);
  $url=cleanB($url);
  $summary=cleanB($summary);
```

```
    print $url."\n";
    $start=$end;
    $count++;
  }
}

sub cutter{
  my ($starttok,$endtok,$where,$str)=@_;
  my $startcut=index($str,$starttok,$where)+length($starttok);
  my $endcut=index($str,$endtok,$startcut+1);
  my $returner=substr($str,$startcut,$endcut-$startcut);
  my @res;
  push @res,$endcut;
  push @res,$returner;
  return @res;
}

sub cleanB{
  my ($str)=@_;
  $str=~s/<b>//g;
  $str=~s/<\/b>//g;
  return $str;
}
```

将该脚本保存为scraper.pl。该抓取程序将域和数字作为参数。该数字是要返回结果的数量，但是多页支持并不包括在代码中。然而，修改脚本以从Google中抓取多个页面简直就是小儿科。注意，抓取程序被修改来查找一些普通的Microsoft Office格式，并且使用site:domain_parameter filetype:XX搜索关键字来循环搜索。现在，所有我们需要的就是那些将所有的东西放在一起并且会对结果做一些后期处理的东西。代码看起来如下所示：

```
#!/bin/perl
use strict;
my ($domain,$num)=@ARGV;

my %ALLEMAIL=(); my %ALLNAMES=();
my %ALLUNAME=(); my %ALLCOMP=();

my $scraper="scrape.pl";
my $docinfo="docinfo.pl";
print "Scraping...please wait...\n";
my @all_urls=`perl $scraper $domain $num`;
if ($#all_urls == -1 ){
  print "Sorry - no results!\n";
  exit;
}
my $count=0;
foreach my $url (@all_urls){
```

```
    print "$count / $#all_urls : Fetching $url";
    my @meta=`perl $docinfo $url`;
    foreach my $item (@meta){
      process($item);
    }
    $count++;
}

#show results
print "\nEmails:\n-------------\n";
foreach my $item (keys %ALLEMAIL){
  print "$ALLEMAIL{$item}:\t$item";
}
print "\nNames (Person):\n-------------\n";
foreach my $item (keys %ALLNAMES){
  print "$ALLNAMES{$item}:\t$item";
}
print "\nUsernames:\n-------------\n";
foreach my $item (keys %ALLUNAME){
  print "$ALLUNAME{$item}:\t$item";
}
print "\nCompanies:\n-------------\n";
foreach my $item (keys %ALLCOMP){
  print "$ALLCOMP{$item}:\t$item";
}

sub process {
  my ($passed)=@_;
  my ($type,$value)=split(/:/,$passed);
  $value=~tr/A-Z/a-z/;
  if (length($value)<=1) {return;}
  if ($value =~ /[a-zA-Z0-9]/){
    if ($type eq "Company"){$ALLCOMP{$value}++;}
    else {
      if (index($value,"\@")>2){$ALLEMAIL{$value}++; }
      elsif (index($value," ")>0){$ALLNAMES{$value}++; }
      else{$ALLUNAME{$value}++; }
    }
  }
}
```

脚本首次运行scraper.pl时，会使用传递给它的域和结果的数量作为参数。它在数组中捕获处理的输出（URL列表），接着为每个URL运行docinfo.pl脚本。接着，脚本的输出被送到下一个处理过程，在这个处理过程中，会执行某些基本检查以查看它是否是公司名称、E-mail地址、用户名或者人名。这些输出被存储在单独的哈希表中，以备后用。当所有的一切安排妥当时，脚本会显示每条收集好的信息以及它在所有页面中出现的次数。它真有效吗？让我们来看一看：

```
# perl combined.pl xxx.gov 10
Scraping...please wait...
0 / 35 : Fetching http://www.xxx.gov/8878main_C_PDP03.DOC
1 / 35 : Fetching http://***.xxx.gov/1329NEW.doc
2 / 35 : Fetching http://***.xxx.gov/LP_Evaluation.doc
3 / 35 : Fetching http://*******.xxx.gov/305.doc
... <cut>

Emails:
-------------
1: ***zgpt@***.ksc.xxx.gov
1: ***ikrb@kscems.ksc.xxx.gov
1: ***ald.l.***mack@xxx.gov
1: ****ie.king@****.xxx.gov

Names (Person):
-------------
1: audrey sch***
1: corina mo****
1: frank ma****
2: eileen wa****
2: saic-odin-**** hq
1: chris wil****
1: nand lal****
1: susan ho****
2: john jaa****
1: dr. paul a. cu****
1: *** project/code 470
1: bill mah****
1: goddard, pwdo - bernadette fo****
1: joanne wo****
2: tom naro****
1: lucero ja****
1: jenny rumb****
1: blade ru****
1: lmit odi****
2: **** odin/osf seat
1: scott w. mci****
2: philip t. me****
1: annie ki****

Usernames:
-------------
1: cgro****
1: ****
1: gidel****
1: rdcho****
```

```
1: fbuchan****
2: sst****
1: rbene****
1: rpan****
2: l.j.klau****
1: gane****h
1: amh****
1: caroles****
2: mic****e
1: baltn****r
3: pcu****
1: md****
1: ****wxpadmin
1: mabis****
1: ebo****
2: grid****
1: bkst****
1: ***(at&l)

Companies:
----------
1: shadow conservatory
[SNIP]
```

为了保护那些被怀疑的政府机构的身份，公司列表被撤消了，但是脚本看起来工作得不错。该脚本可以被很容易地修改来抓取（很多页面中）更多结果、提取更多的域以及获取其他文件类型。那么，到底那家被称为“Shadow Conservatory”的未经检查过的公司是哪一家呢？

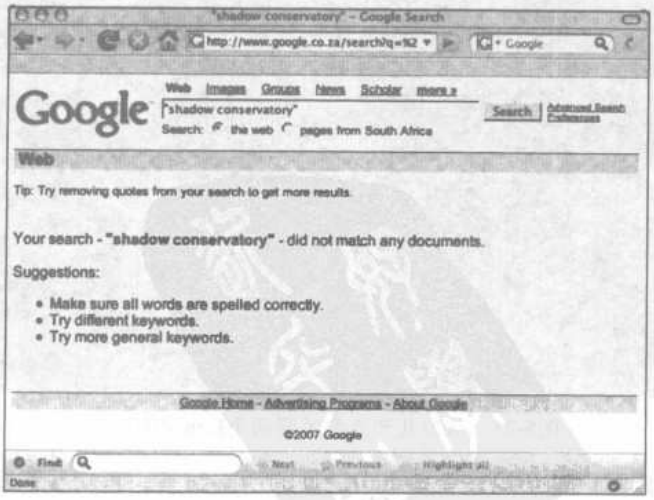


图5-20 “Shadow Conservatory”对应的零搜索结果

在查找使用了哪一种（以及是否使用了）用户名格式时，该工具都能收到很好的工作效果。

试考虑一下从某处挖掘到的用户名列表：

```
Usernames:
-------------
1: 79241234
1: 78610276
1: 98229941
1: 86232477
2: 82733791
2: 02000537
1: 79704862
1: 73641355
2: 85700136
```

从该列表中可以清楚地看到8位数字被用做用户名。在攻击的后期，该信息就应该非常有用了。

3. 更进一步

有时候，你会在要将某个输出用做另一个过程的输入的情况下停下来。该过程可能是另一个搜索，或者它也可以是那些看起来像是在社会网络中搜索E-mail的过程，将DNS名转换为域，解析DNS名字或者确定E-mail账户的存在。我将怎么样将两个E-mail地址链接起来呢？先来看一下Johnny的E-mail地址johnny@ihackstuff.com以及我先前的SensePost的E-mail地址roelof@sensepost.com。为了将这两个地址链接起来，我们可以先从搜索其中的一个E-mail地址并且提取站点、E-mail地址和电话号码来着手。一旦我们有了结果，我们可以对其他的E-mail地执行相同的操作，接着对比他们来查看是否存在任何公共的结果（或者节点）。在本实例中，存在多个公共节点，如图5-21所示。

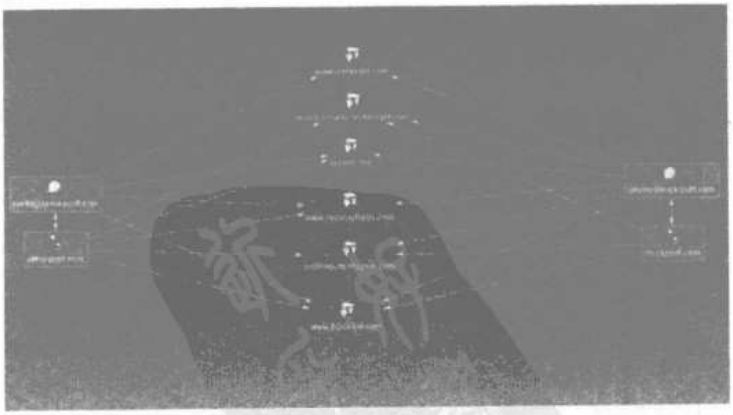


图5-21 从公共节点源关联两个E-mail地址

如果不匹配，我们可以遍历第一个E-mail地址的所有结果，再次提取E-mail地址、站点以及电话号码，接着针对第二个地址重复以上的操作希望这些是公共节点。

不只包含搜索的更复杂的程序又如何呢？你能通过简单地查找公共信息来获取五角大厦的数据中心的位置吗？先看一下图5-22。

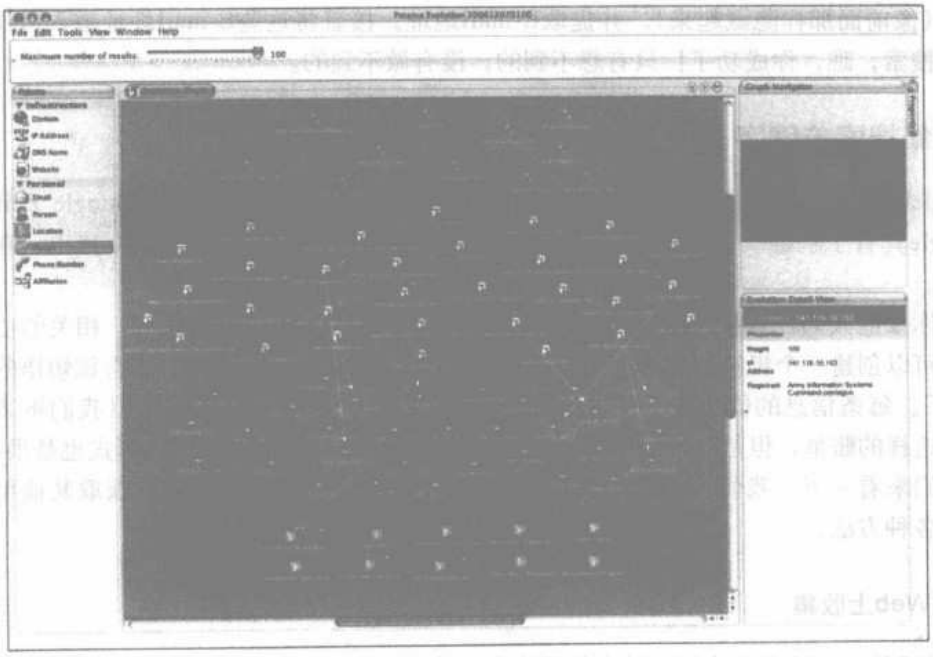


图5-22 使用公共信息获取数据中心的地理位置

这里发生了什么？虽然它看起来相当复杂，但是实际上不尽然。要获取图中显示位置的程序如下所示：

- 在pentagon.mil中挖掘出一个E-mail地址（没有在屏幕快照中显示出来）。
- 从该E-mail地址中提取域（曾在域及子域挖掘小节中提及过）。结果是屏幕快照顶部的节点。
- 从子域执行强力DNS查找，主要查找公共DNS名。结果是屏幕快照第二层节点。
- 为每个域添加MX记录的DNX名。
- 一旦将所有的DNS名解析为IP地址。就得到了屏幕快照中的第三层节点。
- 从IP地址获取地理位置，即最后一层节点。

你可以从屏幕快照中了解几个有趣的事情。第一个是位置，南非，它与www.pentagon.mil链接。这是因为使用了Akamai的结果。查看运行如下：

```
$ host www.pentagon.mil
www.pentagon.mil is an alias for www.defenselink.mil.edgesuite.net.
www.defenselink.mil.edgesuite.net is an alias for a217.g.akamai.net.
a217.g.akamai.net has address 196.33.166.230
a217.g.akamai.net has address 196.33.166.232
```

同样的，应用程序将IP位置视为位于南非，事实也就是如此。用图表的方式显示这些关系的应用程序（如上面的屏幕快照所示）正是Evolution图形用户界面（Graphical User Interface，GUI）客户机程序，可以从Paterva站点中获取。

当把数据与搜索以及其他方式联合起来使用时，那些可以被创建的应用程序的数量将无止境。想知道你的哪一位邻居正在Myspace网站上挂着吗？简单。搜索你的电话号码，忽略最后

4位数字（像前面那样隐藏起来），并提取E-mail地址。接着将这些E-mail地址放入MySpace中进行个人搜索，瞧，你成功了！只有想不到的，没有做不到的。

5.3 收集搜索关键字

Google收集搜索关键字的功能非常强大。如果你对此有置疑，不妨访问Google ZeitGeist页面。Google具有了解每个与互联网相连的每个人的心思的能力。它们可以逐字读出（在线）人种的思想。

如果你知道人们在查找什么，你就可以提供给他们（也就是说卖给他们）相关的信息。事实上，你可以创建一个粗糙的经济模型。探索某个短语的数量是“需求”，包含该短语的页面数是“供给”。每条信息的价格均与需求除以供给的值相关。虽然Google可能（我们祈祷）从来没有拿过这样的账单，但是看到它们将需求和供给添加为结果页面上的索引形式也是很有趣的。

让我们来看一下，我们可以做些什么来获取这一功能。本节将着重考察获取其他用户搜索关键字的各种方法。

5.3.1 在Web上收集

2006年8月，AOL为某个Web站点的研究员发布了2千万搜索记录。这些数据不仅包括搜索关键字，也包括搜索时间，用户点击的链接以及与用户名相关的数据。这意味着虽然你看不到用户名或者E-mail地址，你也仍能找到用户在何时进行了搜索以及他们在搜索什么。这个收集涉及了三个月内的大约658 000用户（大约占所有搜索的1.5%）的数据。该数据在互联网上快速地扩散了开来。源数据在当天就被删除了，但是这已经太晚了。

通过数据手动搜索很无趣。不久之后，那些你可以从中搜索到其他人的搜索关键字的泄漏站点会突然出现，一旦你找到了一些有趣的东西，你将会看到此人执行的所有的其他的搜索。这个关于某人私生活的秘密查看被证实是非常流行的，后期创建的站点允许用户列出有趣的搜索以及根据他们的搜索来剖析他们。这些剖析可以被用来确定至少一个用户的身份。以下是对curityfocus.com上贴出的某篇文章中进行的一个分析：

The New York Times combed through some of the search results to discover user 4417749, whose search terms included, “homes sold in shadow lake subdivision gwinnett county georgia” along with several people with the last name of Arnold.This was enough to reveal the identity of user 4417749 as Thelma Arnold, a 62-year-old woman living in Georgia. Of the 20 million search histories posted, it is believed there are many more such cases where individuals can be identified. ...Contrary to AOL's statements about no personally-identifiable information, the real data reveals some shocking search queries. Some researchers combing through the data have claimed to have discovered over 100 social security numbers, dozens or hundreds of credit card numbers, and the full names, addresses and dates of birth of various users who entered these terms as search queries.

站点http://data.aolsearchlog.com为所有的搜索关键字提供了一个界面，也显示了被收集到

的部分剖析结果，如图5-23所示。

图5-23 允许你搜索AOL搜索关键字的站点

虽然该站点会让你忙活几分钟，但是它确实包含了那些你不了解的人的搜索术语，不过数据是旧的、静态的。是否有一种查看地更实时的搜索的方式（动态方式）？

5.3.2 自行收集

1. 搜索关键字

当你搜索某些资料时，查询都会定位到Google计算机。你每次在Google上执行搜索时，它们都会查看你是否通过了一个cookie。如果你没有通过cookie，它们会指示你的浏览器设置一个。该浏览器将被指示来为每个对任何Google系统（例如，*.google.com）的后续请求通过cookie，并且继续该操作直到2038年。因此，从同一台笔记本电脑在两个不同国家发出的两个搜索，即便是它们相隔两年之久，也依然会发出同样的cookie（假定该cookie的存储从来没有被清除过），Google将了解到搜索来自同一个用户。该搜索不必横穿整个网络，因此如果我在该搜索横穿网络的时候截获它的话，我便可以阅读到它。该技术被称为“嗅探（sniffing）”。在前面的小节中，我们已经讲解了如何对Google提出请求。让我们来看一下，较少的cookie请求是怎么样的、Google如何来设置cookie：

```
$ telnet www.google.co.za 80
Trying 64.233.183.99...
Connected to www.google.com.
Escape character is '^]'.
GET / HTTP/1.0
Host: www.google.co.za

HTTP/1.0 200 OK
Date: Thu, 12 Jul 2007 08:20:24 GMT
Content-Type: text/html; charset=ISO-8859-1
Cache-Control: private
```

```
Set-Cookie:
PREF=ID=329773239358a7d2:TM=1184228424:LM=1184228424:S=MQ6vKrgT4f9up_gj;
expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.co.za
Server: GWS/2.1
Via: 1.1 netcachejhb-2 (NetCache NetApp/5.5R6)

<html><head>....snip...
```

注意Set-Cookie部分。ID部分是较有意思的部分。其他的cookie（TM和LM）包括cookie的诞生日期（从1970年起的很短的时间内），以及偏好上次更改的时间。ID会一直保持不变直到你清除了浏览器中存储的cookie。这意味着每个来自你浏览器的后续请求都会包括cookie。

如果我们有一种阅读Google流量的方法，便可以使用cookie来识别来自同一浏览器的后续搜索。有两种方式来查看定位到Google的请求。第一种方法包括沿着流量创建嗅探器，嗅探器会监视定位到Google的请求。第二种方法更容易一些，但是包括那些几乎确定已经在原位的基础结构：使用代理。流量代理的方式有两种。用户可以手动地在浏览器上设置代理，也可以在上行数据流的某处透明地进行。有了透明的代理后，用户通常不会意识到流量被发送到代理，并且该过程通常不会得到用户的允许或者悉知。同样，用户也没有办法接通或者断开代理。默认情况下，所有定位到端口80的流量都会被窃听并发送到代理。在很多这样的设置中，其他的端口也会被窃听，通常是那些标准的代理端口，如3128，1080和8080。因此，即使你在浏览器中设置了代理，流量在到达手动配置的代理前会被窃听，并且发送给透明的代理。这些透明的代理通常会用于网络中的边界，比方说你的ISP互联网关或者靠近你公司的互联网连接处。

一方面，我们拥有能够提供良好机制来跟踪搜索关键字的Google，另一方面，我们拥有一些很棒的可以收集和记录流量的透明设备。这看似是数据挖掘的一个完美组合。

让我们来看一下如何将一些为我们达成此工作的资源聚集到一起。开始时，我们需要配置一个代理来记录整个请求的头部以及GET参数，并且接收来自透明网络重定向的连接。要想做到此，你可以使用仅有3个对常备的标准设备配备进行修正的流行的Squid代理。以下是你需要的三条语句：

第一条是告诉Squid接收为来自端口3128上的透明重定向的连接：

```
http_port 3128 transparent
```

第二条是告诉Squid记录完整的HTTP请求头部：

```
log_mime_hdrs on
```

最后一条是告诉Squid记录GET参数，而不仅是主机及路径：

```
strip_query_terms off
```

完成该设置并且运行Squid代理之后，还需要做的唯一一件事是将流量发送给它。这可以采用很多方式进行，并且通常会在防火墙处进行。假设你在运行带有所有支持此的核心选项设置的FreeBSD（并且Squid代理也位于同一框中），下面的这行语句将会把所有发送到端口80的流量定位到Squid框中。

```
ipfw add 10 fwd 127.0.0.1,3128 tcp from any to any 80
```

也可以为其他操作系统和/或防火墙创建相似的配置。在Google上查找“transparent proxy network configuration”（透明代理网络的配置），并选择一个合适的。有了这一设置，我们将可以窃听所有的位于防火墙后的Web流量。虽然有很多有趣的可以从其他类型的Squid日志中捕获到的信息，我们也只会关注与Google相关的请求。

只要你的透明代理已各就其位，你就会看到流入的请求。下面的代码是来自刚进行完短语“test phrase”搜索的简单搜索后代理日志的一行：

```
1184253638.293 752 196.xx.xx.xx TCP_MISS/200 4949 GET
http://www.google.co.za/search?hl=en&q=test+phrase&btnG=Google+Search&meta= -
DIRECT/72.14.253.147 text/html [Host: www.google.co.za\r\nUser-Agent: Mozilla/5.0
(Macintosh; U; Intel Mac OS X; en-US; rv:1.8.1.4) Gecko/20070515
Firefox/2.0.0.4\r\nAccept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,ima
ge/png,*/*;q=0.5\r\nAccept-Language: en-us,en;q=0.5\r\nAccept-Encoding:
gzip,deflate\r\nAccept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\nKeep-Alive:
300\r\nProxy-Connection: keep-alive\r\nReferer: http://www.google.co.za/\r\nCookie:
PREF=ID=35d1cc1c7089ceba:TM=1184106010:LM=1184106010:S=gBAPGByiXrA7ZPQN\r\n]
[HTTP/1.0 200 OK\r\nCache-Control: private\r\nContent-Type: text/html; charset=UTF-
8\r\nServer: GWS/2.1\r\nContent-Encoding: gzip\r\nDate: Thu, 12 Jul 2007 09:22:01
GMT\r\nConnection: Close\r\n\r]
```

注意，搜索关键字将以“q”参数的值“test+phrase”出现。另外，还要注意被设置为“35d1cc1c7089ceba”的ID cookie。不管随后的搜索关键字是什么，cookie的值都将会保持不变。在以上的文本中，发出该请求的IP数字也会被列出来（不过大部分是以x形式列出的）。此后，就剩下一个实现的问题了，即创建一个可以析出搜索关键字、IP地址和cookie的系统，并且将该系统放入数据库中以进行更深入的分析。像这样的系统将静静地收集每天流入、流出的搜索关键字。

在SensePost期间，我编写了一个非常简单的（但未经优化的）应用程序来做到此，并将该应用程序称为PollyMe（www.sensepost.com/research/PollyMe.zip）。该应用程序看起来与进行AOL搜索的Web界面的工作方式相似，不同的是你在搜索你已经收集到的日志。正如AOL界面那样，你可以搜索搜索关键字，找出搜索者的cookie值，并且查看有其他与该值相关的搜索。除些之外，你还可以查看同一时期内用户访问过的其他的站点。该应用程序甚至可以支持你在访问过的URL中搜索关键字。

工具和技巧

如何发掘透明的代理

在某些情况下，了解你是否位于透明的代理之后很有用。有一个快速了解此情况的方法。远程登录到位于你网络之外的几个随机IP地址上的端口80。如果你获取了每次的连接，那么你就位于一个透明的代理之后。（注意：在进行该测试时不要尝试使用私有的IP地址范围。）

另一个方法就是查找Web站点地址，接着远程登录到IP数字，发布一个GET/HTTP/1.0（没有Host:头部），并且查看响应。某些代理使用Host:头部来确定你需要连接的地方，如果

没有它你就会得到一个错误。

```
$ host www.paterva.com
www.paterva.com has address 64.71.152.104

$ telnet 64.71.152.104 80
Trying 64.71.152.104...
Connected to linode.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 400 Bad Request
Server: squid/2.6.STABLE12
```

我们不但可以了解我们正处于透明代理的状态中，还可以看到使用的代理的类型和服务器。注意第二种方法不适用于所有的代理，特别是那些很多ISP使用的更大的代理。

2. Gmail

收集搜索关键字并且基于它来剖析人是一件有趣的事，但是也仅仅是如此而已。更有趣的是他们的信箱里发生的事情。虽然这超出了本书的探讨范围，但是还是让我们一同来看一下我们可以使用代理设置和Gmail来做什么。在我们进一步研讨这个问题之前，你需要了解（大多数）Web应用程序的工作方式。在成功登录到Gmail之后，cookie被传递给Web浏览器（使用与普通搜索相同的方式），该cookie被用来标识你。如果没有cookie，那么你将必须为你要浏览的每个页面提供用户名和密码，因为HTTP是一个无国籍的协议。因此，当你登录到Gmail时，Google唯一能使用来识别你的就是cookie。当你的证书通过SSL传递给Google时，会话的其他部分也会畅通无阻地进行（除非你将它强加给SSL，这不是默认的行为），这意味着你的cookie一直在畅通无阻地传播。被用来识别我的cookie畅通无阻，我的整个请求（包括内含cookie的HTTP头部）都会记录在我不了解的某处的透明代理上。

眼下，你可能想了解其中的要点。众所周知，未加密的E-mail的传递将不受约束，而且那些位于上行数据流的人可以阅读它。但是还是有一个细微的差别。嗅探E-mail允许你读取E-mail本身。Gmail cookie允许你读取用户的Gmail应用程序，该应用程序允许你读取地址簿，会赋予你搜索旧的收取的邮件和发出的邮件的能力，会赋予你像用户那样发送E-mail，允许你读取用户的日历、搜索历史记录（如果启用了的话），会赋予你通过内置的Gmail聊天工具与他人在线聊天的能力，等等。这样看来，这里确实存在着很大的区别。同样，只要在ISP处提及“sniffer”单词，所有的警报都会解除。但是要调节代理又完全不同了。

让我们来看一看这是如何进行的。进行过数次试验后，我们发现真正需要用来在Gmail上假扮某人的唯一的cookie就是GX cookie。因此，通常要做的事就是成为连接到代理的网络上的透明的代理用户，等待接收一些Gmail流量（浏览器登录到Gmail并向应用程序发出频繁的请求，所有的请求都会传送GX cookie），分析GX cookie，并且制作正确的请求来获取用户的联系列表，接着在他或她的E-mail信箱中搜索某些有趣的短语。

获取地址簿的请求如下：

```
GET /mail?view=cl&search=contacts&pnl=a HTTP/1.0
Host: mail.google.com
Cookie: GX=xxxxxxxxxx
```

搜索邮箱的请求如下：

```
GET /mail?view=tl&search=query&q=__stuff_to_search_for___ HTTP/1.0
Host: mail.google.com
Cookie: GX=xxxxxxxxxxx
```

GX cookie必须是你从Squid日志中挖掘到的GX。你需要对接受数据进行必要的解析，不过，有用的东西遍地是。对此类搜索进行自动化处理只是小事一碟。事实上，邪恶的管理员可以完成更进一步的搜索。他或她可以挖掘到用户地址簿并给列表中的所有的人都发送一封E-mail，接着等待他们阅读E-mail，挖掘他们的GX，再开始这个过程。Google将专门去辨认那写看似无害的E-mail是如何变成病毒的（当然，它不可能是真正的病毒，但是当防火墙后部有足够大的网络时，它将拥有一些与蠕虫相同的特性）。

提示

并非Google专属的事

眼下，你也许会想，这是Google必须要做的事。但是当你仔细考虑一下，你就会发现这是所有Web应用程序的事。它们可以应用的唯一现实的解决方案就是确保整个会话发生在SSL层之上，SSL在计算机性能方面的能力非常强大。其他的Web邮件提供商也会遭受同样的问题。唯一不同的是他们的应用程序的特性数量与Gmail不同（可能用户基数更小），这使得它们的目标也不那么大了。

放心。尽管对于ISP网络管理员来说做这些事是可行的，但是受严厉的保密法的制约，他们也不太可能去这么做。在大多数国家，你要遵守法律来获取合法的窃听（例如，嗅探你自己的流量并且阅读你自己的E-mail）。作为一个使用者，你必须要意识到：当你需要对某些东西严格保密的话，就需要对它合理地加密。

5.3.3 甜言蜜语

假设你在运行一个代号为“Sookha”的超级机密项目。没有人知道这个项目名称。如果某人在Google上查找单词Sookha，那么你应该在不让搜索者察觉你所了解的事实情况下获知。你能做的就是注册一个使用Sookha作关键字的关键字广告。这样做的目的就是关键字广告不但可以告诉你别人在何时点击了你的广告，而且还能告诉你有多少痕迹被展示（显现）了出来，别人对该单词搜索过多少次。为了不惊动你的潜在搜索者，你应该为广告选择一种不会引人注意的方式。下面的屏幕快照（如图5-24所示）就说明了这样的广告的设置。

一旦别人搜索你的关键字，广告就会出现，并且很可能不会引人注意。但是，在管理控制台你可以看到留下了一个痕迹，并且你可以有把握地说：“我找到了组织的纰漏。”

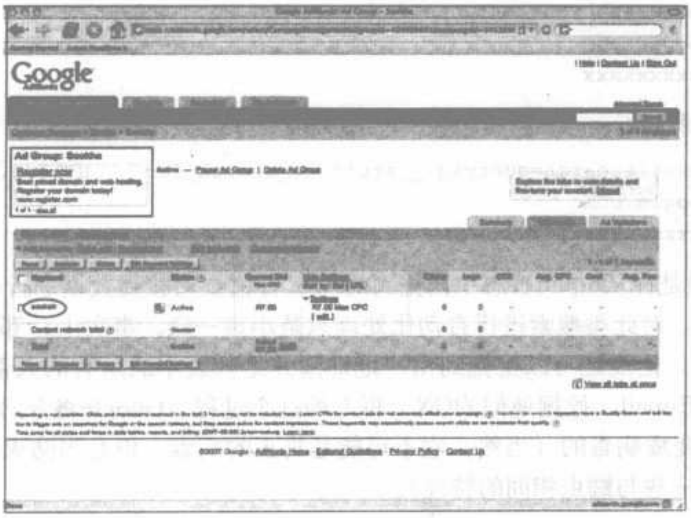


图5-24 自称为甜言蜜语的关键字广告

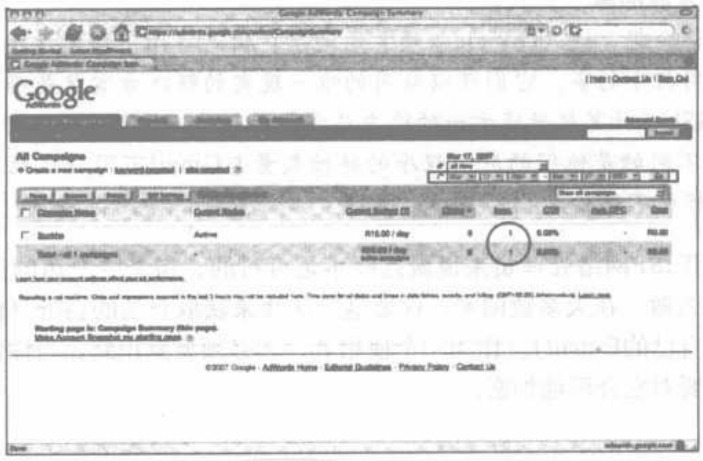


图5-25 显示了一个单一痕迹的关键字广告控制面板

5.3.4 引用者

另一种查找人们搜索的方式是查阅引用者：到达你的Web站点的请求的标题。当然这是有限的。这里的观念是别人在Google上查找资料，你的站点要在结果列表上列出，它们会点击定位到你的站点的链接。虽然这对于那些没有流量或者是低流量的站点而言不是特别有用，但是对于那些访问非常流行的站点的人而言，该方法非常好用。它的实际工作方式是什么样呢？你访问的每个站点都知道你访问过的前一个站点。它会被放在HTTP头部中作为引用者发送。当别人访问Google时，一旦用户到达你的站点，他们的搜索关键字会作为URL的一部分出现（因为它是一个GET请求）并且传送到站点。这为你提供了在他们到达你的站点之前查看搜索内容

的可能，对于营销人员而言，这一点非常有用。

通常，来自Google搜索的Apache日志条目的内容可能看起来如下所示：

```
68.144.162.191 - - [10/Jul/2007:11:45:25 -0400]"GET /evolution-gui.html HTTP/1.1"
304 - "http://www.google.com/search?hl=en&q=evolution+beta+gui&btnG=Search"
"Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.8.1.4) Gecko/20070515
Firefox/2.0.0.4"
```

我们可以从该条目中发现在用户到达我们的页面之前，他正在Google上搜索“evolution beta gui”，之后他最后访问的是“/evolution-gui.html”页面。很多分析Web日志的应用程序都能自动为日志提取关键字，并且为你提供了一个很好的关键字及关键字搜索频率列表。

有没有一种使用它来随意挖掘搜索关键字的方法呢？不见得会有。最好的选择（事实上并非那么实用）就是创建一个具有各种类型的内容的流行站点，并且查看你是否可以吸引访问者因某种唯一的原因来挖掘他们的搜索关键字。另外，你肯定还会为这些访问者提供更好用途，而不只是他们用于搜索的关键字。

5.4 总结

本章，我们学习了使用Google挖掘有用信息的多种方法。当你具有自动执行特定操作的能力时，搜索能力将得到进一步的提升。本章向你展示了如何使用简单的脚本来获取自动操作的功能。此外，当你拥有了收集信息位并拼凑完整信息的方法时（例如，不只是搜索，还有执行带有挖掘信息的额外的函数），一切就变得有趣了。本章介绍的工具和技巧仅是数据收集（或挖掘）这种巨大的冰山的一角。但愿它会开启你的关于可以获取哪些资料的思维之门。这一想法永远都无法完全详尽阐述每一种方法，但是它可以保证你的思维朝着正确的方向发展，并激发创新思想。如果本章激励了你，让你能够用你自己的脚本来进行攻击，并能够展现一些令人惊讶的东西，那么本章节就实现了它的用途了（而我会很愿意从你那听到这个好消息）。

第6章 搜索漏洞利用与查找目标

6.1 简介

漏洞利用（Exploit）代码是黑客使用的一种工具。他们设计用来洞察一个目标，大部分黑客都有许多可以随意支配的漏洞利用。一些称为零天（zero day）或者叫做0day的漏洞利用在某段时期以内是不公开的，不过它们最终会公布，发布到新闻组或者网站上供世界分享。有许多的网站都被用来分发漏洞利用代码，所以能够相当简单地利用强大的Google来搜索这些工具。而搜索潜在的目标就稍微有些困难了，即使许多流行的Web应用程序的安全顾问包含了设计用来搜索潜在目标的Google查询。

在这一章中，我们将探索搜索漏洞利用代码与潜在有漏洞的目标的方法。这些严格来讲并不属于“黑暗面”的练习，因为安全专家也经常在漏洞评估中使用公开的漏洞利用代码。但是，只有黑帽才会在没有经过事先允许的情况下使用那些工具来攻击系统。

6.2 搜索漏洞利用代码

有数不清的成千上万个网站在给普通公众提供漏洞利用。黑帽通常在黑客社区中给同伴黑帽提供漏洞利用。白帽提供漏洞利用则是用于消除在评估过程中自动工具的误报。例如remote exploit和vulnerable exploit这样的简单搜索是通过关注在安全社区中常用的术语来搜索漏洞利用。其他的搜索，例如inurl:0day不再像以前一样有用，但是旧的备用搜索，例如inurl:sploits仍然能起到很好的效果。问题是大多数安全人员不仅仅在互联网上查找漏洞利用；他们经常是在一些网站上查找主流的工具，只有当他们在收藏的网站中找不到某些工具时，才会求诸于搜索引擎。当在Web中搜索某个特定的安全工具时，应该首当其冲使用Google来搜索。

搜索公开的漏洞利用站点

一种搜索漏洞利用代码的方法是，首先关注漏洞利用程序源代码的文件扩展名，然后搜索代码中的特定内容。由于源代码是难以阅读的机器代码的文本表示，所以Google非常适合于这类搜索任务。例如，大量漏洞利用是用C语言编写的，这些源代码大都以.c为扩展名。当然，一个filetype:c c搜索会返回将近500 000条结果，这意味着我们必须让搜索更为精确。查询filetype:c exploit会返回大约5000条结果，其中的大部分确实是我们要找的类型的程序。要明白这些都是保存含有单词exploit的C源代码的流行网站，所以把返回的列表作为一个收藏夹列表是个不错的主意。使用页面提取技术，我们可以通过运行一个如下的UNIX命令从Google结果页面中分离出这些站点：

```
grep cached exp | awk -F" -" '{print $1}' | sort -u
```

使用良好的、老式的剪切和粘贴或者例如lynx -dump这样的命令就可以很好地捕捉这些页面。我们利用这种方法在Google中提取了20条结果，精选了其中的一些，如下面的列表所示。

download2.rapid7.com/r7-0025
securityvulns.com/files
www.outpost9.com/exploits/unsorted
downloads.securityfocus.com/vulnerabilities/exploits
packetstorm.linuxsecurity.com/0101-exploits
packetstorm.linuxsecurity.com/0501-exploits
packetstormsecurity.nl/0304-exploits
www.packetstormsecurity.nl/0009-exploits
www.0xdeadbeef.info
archives.neohapsis.com/archives/
packetstormsecurity.org/0311-exploits
packetstormsecurity.org/0010-exploits
www.critical.lt
synnergy.net/downloads/exploits
www.digitalmunition.com
www.safemode.org/files/zillion/exploits
vdb.dragonsoft.com.tw
unsecure.altervista.org
www.darkircop.org/security
www.w00w00.org/files/exploits/

Google搜索背景知识

Google取证

Google也是一种能够执行数字取证的很好的工具。如果在一台被攻陷的机器中发现了一个可疑的工具，那么一种标准的通常做法是用一个UNIX命令来运行这个工具以得到这个程序中可以阅读的文本，如strings -8。这样通常能够得到类似于工具的用法文本等信息，我们就可以利用这些信息在Google中进行查询以查找相似的工具。虽然如今出现了越来越多的混淆程序，但是如果合理地把strings和Google组合起来，那么它们仍旧是非常强大的，能够揭开一台被攻陷的机器中的大量的可疑工具的神秘面纱。

6.3 通过常见代码字符串搜索漏洞利用

由于网页会以各种方式显示源代码，所以一个源代码列表可能实际上是任何类型的文件扩展名。例如，一个PHP页面可能会生成一个C文件的文本视图，这样，从Google的角度来看，

其文件扩展名就是.PHP而不是.C。

另外一种搜索漏洞利用代码的方法是关注源代码自身内部常见的字符串。一种实现此结果的方法是关注常用的包含文件的引用或者头文件引用。例如，许多C程序都包含标准的输入/输出库函数，通常都是在源代码里添加一个include语句，例如#include <stdio.h>。例如像“#include <stdio.h>” exploit这样的查询能够搜索包含单词exploit的C源代码，而不管文件的扩展名。这个查询会捕捉到在HTML文档显示的代码（和代码段）。我们可以把这个搜索进一步扩大，让它包含带有友好的用法提示语句的程序，例如“#include <stdio.h>” usage exploit返回的结果如图6-1所示。

图6-1 搜索带有非标准扩展名的漏洞利用代码

这个搜索可以返回许多条结果，几乎所有的结果都包含漏洞利用。使用遍历技术（或者简单地点击网站的主页）就能够得到其他的漏洞利用或者工具。注意，这些结果大都是HTML文档，而它们在前面的filetype:c查询中都被排除了。有许多搜索使用常见代码字符串的源代码的方法，但是不可能能用某一种方法就可以找到所有的源代码。有些代码使用这种技术就可以很方便地找到，而有些则需要一点查询优化技术。表6-1给出了一些使用常见字符串搜索源代码的建议。

表6-1 用常见字符串搜索源代码

语言	扩展名（可选）	样例字符串
asp.net(C#)	Aspx	“<%@ Page Language=“C#”” inherits
asp.net(VB)	Aspx	“<%@ Page Language=“vb”” inherits
asp.net(VB)	Aspx	<%@ Page LANGUAGE=“JScript”
C	C	“#include <stdio.h>”
C#	Cs	“using System;” class

（续）

语言	扩展名（可选）	样例字符串
c++	Cpp	“#include “stdafx.h””
Java	J、JAV	class public static
JavaScript	JS	“<script language=“JavaScript” >”
Perl	PERL、PL、PM	“#!/usr/bin/perl”
Python	Py	“#!/usr/bin/env”
VBScript	.vbs	“<%@ language=“vbscript” %>”
Visual Basic	Vb	“Private Sub”

在使用这张表时，filetype搜索是可选的。在大多数情况下，你会发现可以很容易地找到这些样例字符串，所以你是不会错过这些扩展名的。

6.4 使用Google代码搜索查找代码

Google代码搜索（www.google.com/codesearch）可以用来搜索公开源代码。除了允许查询包括强大的正则表达式外，代码搜索还引入了特殊的操作符，部分操作符见表6-2。

表6-2 Google代码搜索操作符

操作符	说 明	示 例
file	搜索指定的文件类型。参数可以包括文件名、扩展名或者完整的路径名	file:js
package	在一个指定的通常以URL或者CVS服务器名称列出的包中搜索	package:linux.*.tar.gz buggy
lang	搜索以特定语言编写的代码	lang: “c++”
license	搜索基于特定证书编写的代码	license:gpl

代码搜索是前面章节提及的技术补充。例如，在表6-1中我们使用Web搜索关键字”#include <stdio.h>”来查找用C编程语言编写的程序。该搜索很有效，它能够在忽略文件扩展名的情况下查到C代码。只需将引号删除即可将相同的查询重新格式化为代码搜索查询，如图6-2所示。

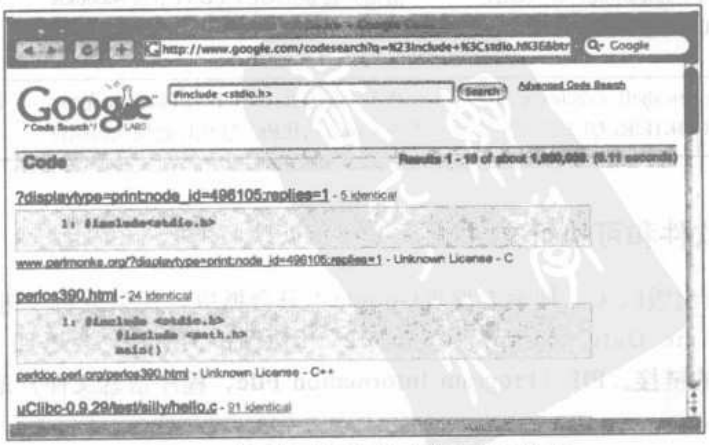


图6-2 代码搜索被用来查找头部字符串

如果我们要查找C代码，不妨使用代码搜索查询lang:c或者lang:c++。尽管它看起来很像是在通过文件扩展名搜索，但是它要比文件扩展名搜索更高级。Google的代码搜索十分胜任分析代码（不考虑扩展名）来判定代码编写采用的编程语言的工作。让我们来看一下图6-2中的第二条结果。正如摘要清晰显示的那样，这是一段嵌入到HTML文件中的C代码，这可以通过文件名perlos390.html知晓。

正如很多研究学者和博客人士报道的那样，Google代码搜索也可以用来查找那些包含潜在的易受攻击的漏洞的软件，见表6-3。

表6-3 搜索易受攻击的代码的Google代码搜索查询

Google代码搜索查询	说 明	作 者
lang:php (echo|print).*\$_(GET|POST|COOKIE|REQUEST	显示传递给GET/POST或者cookie的不可信任的变量的代码。经典的XSS（跨平台脚本）缺陷	Ilia Alshanetsky
<%=.*getParameter*	因HTML编码的用户输入而允许Java中存在XSS的代码	Nitesh Dhanjani
lang:php echo.*\$_SERVER\['PHP_SELF']	因为PHP_SELF回应造成的XSS缺陷	
echo.*\$_(GET|POST).*	以上查询的通用版本	Chris Shiflett
lang:php query\(.*\$_(GET|POST|COOKIE|R-EQUEST).*\)	基于用户提供的GET/POST请求创建的SQL查询。这可以是SQL注入点	Ilia Alshanetsky
.*mysql_query\(.*\$_(GET|POST).*	基于用户提供的GET/POST请求创建的SQL查询。这可以是SQL注入点。MySQL专用	Nitesh Dhanjani
lang:php "WHERE username='$_"	由于WHERE子句中的未处理的输入引发的SQL注入	Chris Shiflett
.*executeQuery.*getParameter.*	由于使用不可信任的用户输入引起的SQL查询的执行引发的Java代码中的SQL注入	Stephen de Vries
lang:php header\s*\("Location:.*\$_ (GET|P-OST|COOKIE|REQUEST).*\)	由用户提供的GET/POST请求和cookie创建的代码输入。这可以允许恶意代码的执行	Ilia Alshanetsky
lang:php (system|popen|shell_exec|exec)\s*\(\$_(GET|POST|COOKIE|REQUEST).*\)	将不可信任的GET/POST数据传递给系统来执行的代码。这允许远程代码执行	Ilia Alshanetsky

6.5 搜索恶意软件和可执行文件

自本书的第1版出版以来，搜索者发现Google不只会抓取，还会分析二进制文件或者可执行文件。查询"Time Date Stamp: 4053c6c2"（如图6-3所示）会返回一个连接到名为Message.pif的程序的链接。PIF（Program Information File，程序信息文件）是一类Windows可执行文件。

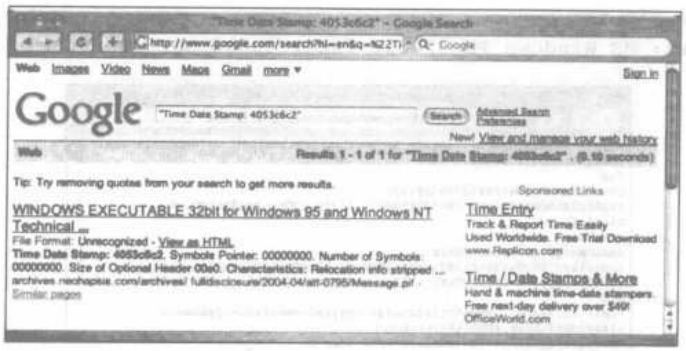


图6-3 Google分析可执行文件

既然可执行文件是机器（而非人类）可读的，那么在搜索结果的摘要中看到文字就很出乎意料了。然而，摘要文字是Google对二进制文件的分析结果。单击该结果的“以HTML格式查看”（View as HTML）的链接会显示文件的完整分析，如图6-4所示。列出的信息看起来很重要是因为列出的信息却是很重要。

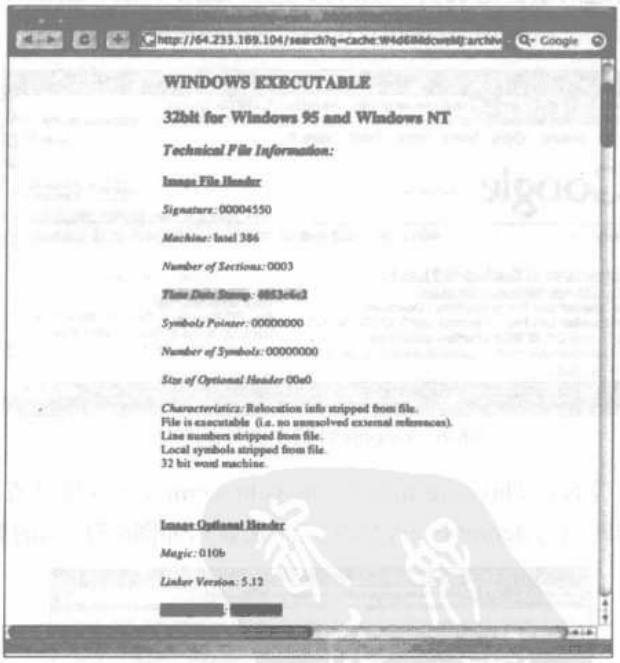


图6-4 Google分析二进制文件

单击文件链接（不是HTML链接）之后，你的浏览器将会产生如图6-5所示的效果。

二进制文件照道理不应该在浏览器中显示。不过，如果我们右击文件链接并且选择“另存为”（Save As）来将它保存在我们本地的上，我们可以对该文件进行基本分析，以查证事实真相。例如，在安装了Linux或者Mac OS X的机器上运行file命令可以揭示Message.pif确实是一个Windows执行文件：

```
$ file Message.pif.txt
Message.pif.txt: MS Windows PE 32-bit Intel 80386 GUI executable not relocatable
```

图6-5 二进制浏览器垃圾

因此，Google获取并分析了它在Web上找到的二进制文件。那又怎么样呢？首先，看到Google的介于是一件有趣的事。这表明Google的性能在逐步扩展。例如，现在Google已经能认出恶意软件了。先来看一下图6-6中所示的对Backup4all备份软件的搜索。

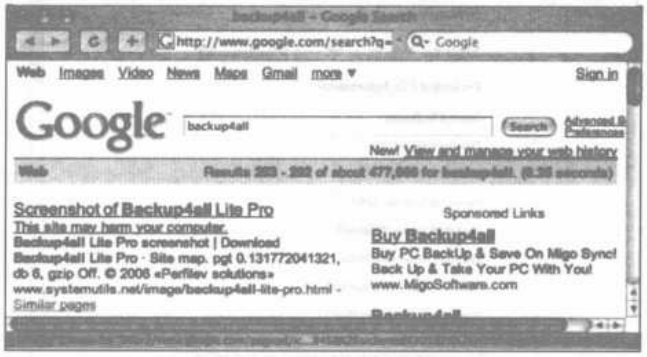


图6-6 Google对恶意软件的警告

注意，站点下方的警告：This site may harm your computer（该站点可能包含恶意代码）。单击该文件链接不会跳转到systemutils.net URL，而是显示如图6-7所示的警告页面。

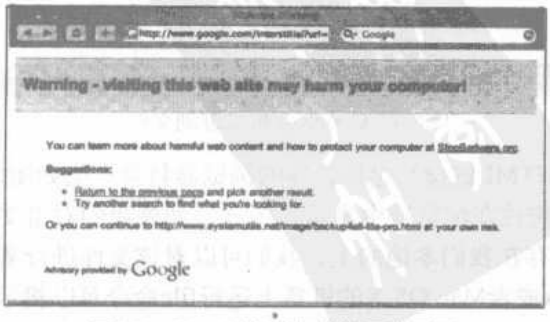


图6-7 Google的恶意软件屏蔽页面

这当然是一个便利的功能，但是因为本书是介绍Google Hacking的，而不是介绍Google拯救全球互联网冲浪者的计划的，所以我们只用深入地了解事情的黑暗本质：Google可以用来搜索仍然活跃的恶意软件。正如Websense在2006年宣布的那样，该功能可以通过关注个别文件的具体细节来放大到搜索非常特别的可执行文件，例如，时间戳（Time Stamp）、尺寸（Size）和进入点（Entry Point）域。H.D.Moore将此推进了一步，并且创建了一系列的恶意软件搜索引擎，相关信息可访问：http://metasploit.com/research/misc/mwsearch，如图6-8所示。

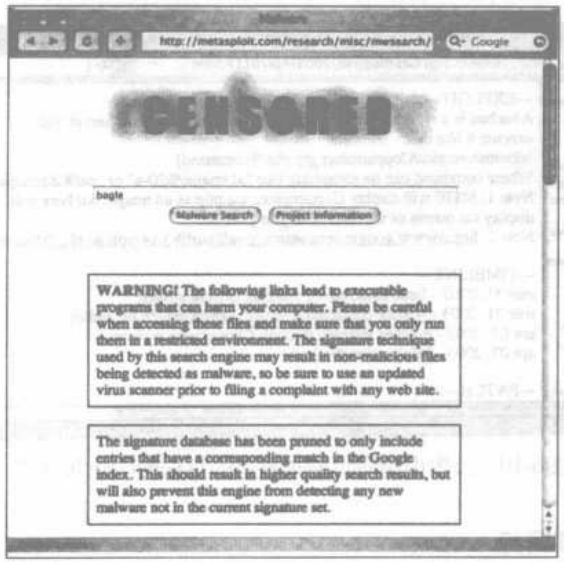


图6-8 基于Google二进制搜索的H.D.Moore恶意搜索引擎

例如，搜索bagle会找到如图6-9所示的几个结果。

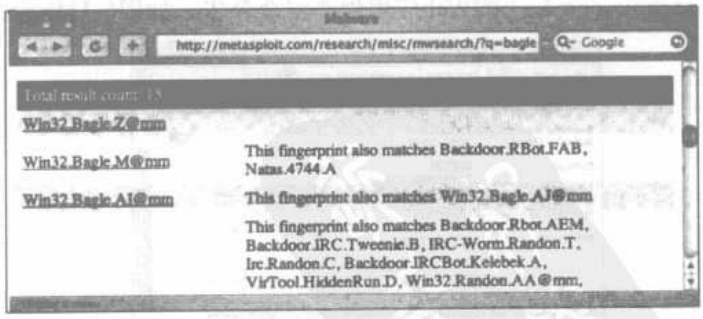


图6-9 搜索bagle的恶意软件搜索

单击搜索结果中的第二个结果会跳转到Google中的“Time Date Stamp: 4053c6c2”“Size of Image: 00010000”“Entry Point: 0000e5b0”“Size of Code: 00005000”的Web搜索结果页面——一个非常长的独特地描述了Win32.Bagle.M蠕虫的二进制签名查询。该查询的Google结果页面如图6-3所示。还记得这个文件吗？它是那个我们成功地下载并且安放到计算机桌面的文件！

因此，尽管Google的二进制分析功能的意向是好的，但是经验丰富的攻击者仍旧可以使用它来进行恶意的攻击。

6.6 搜索易受攻击的目标

越来越多的攻击者使用Google来搜索具有某个特定漏洞的基于Web的目标。实际上，公开的漏洞公告中通常都含有潜在漏洞目标的Google链接，如图6-10所示。

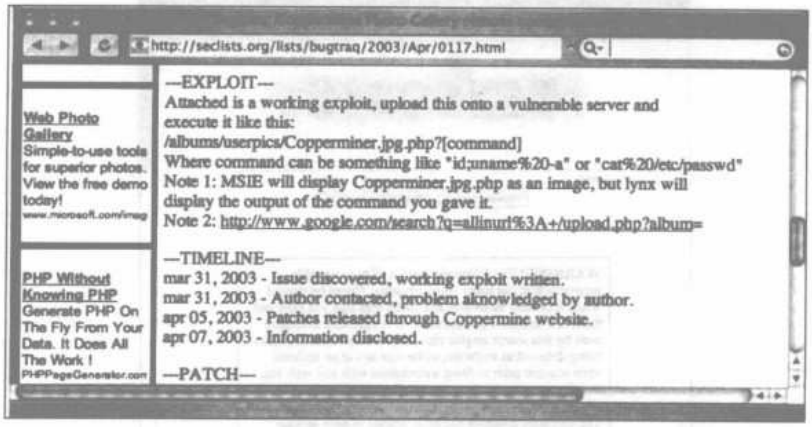


图6-10 公告中指向可能被攻击的目标的Google链接

6.6.1 利用演示页面搜索目标

我们将在这一节中了解到，搜索易受攻击的目标的过程是相当直接的。同样，在下一节中，我们也将了解到有些时候在这一过程中也会碰到一些困难。我们来看一个在2004年10月10日发表在Secunia（www.secunia.com）上的Web应用程序安全公告，如图6-11所示。

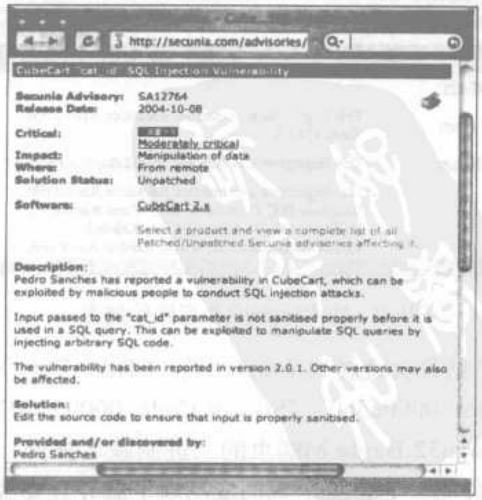


图6-11 典型的Web应用程序安全公告

这个特别的公告公布了一个受影响的软件制造商的网站链接。并非所有的公告都会列出这样的链接，但是一个快速的Google查询就可以帮助你搜索到这些制造商的页面。因为我们的目标是创建一个查询字符串来搜索Web中易受攻击的目标，所以制造商的网站就是研究目标产品网页的最好的地方。和许多软件制造商的网站一样，图中所示的CubeCart站点显示了产品演示的链接以及运行该产品的活动站点，如图6-12所示。

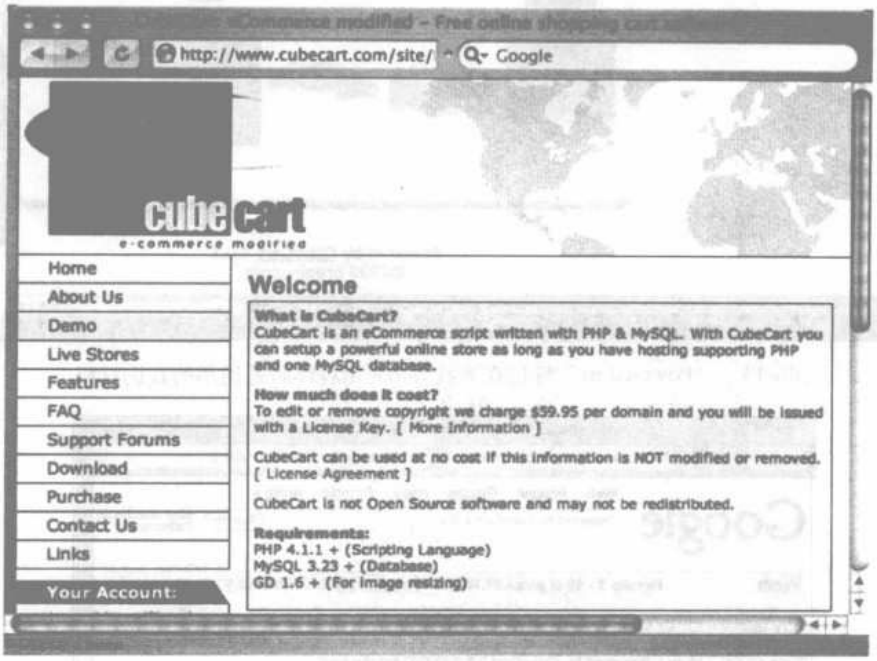


图6-12 制造商网站通常会提供产品演示

在写作这一节时，该站点的演示页面是无效的，但是其中的活动站点列表是有效的。活动的站点通常更为合适，因为我们可以了解到网站的各种变形的显示方式。例如，一些管理员可能会修改制造商提供的网页格式以匹配整个站点的风格。这类修改会影响目的在于制造商提供的页面格式的Google搜索的效果。

仔细浏览图6-12中的活动站点，我们发现大部分站点看起来都很相似，而且几乎在每个站点的主页底部都有一个“powered by”消息，如图6-13中的（变化很大的）例子所示。

在这个例子中，该页面在脚注部分显示了“Powered by CubeCart 2.0.1”。由于在安全公告中列出的易受攻击的软件版本正是CubeCart 2.0.1，所以我们不需要再做其他什么工作就能够创建一个可以搜索到Web中易受攻击的目标的查询。其最终的查询字符串“Powered by CubeCart 2.0.1”返回了27 000多个可能易受攻击的目标，如图6-14所示。

把这些网站列表和在Secunia安全公告中发布的漏洞利用工具结合起来，攻击者就可以访问到很可能被攻陷的在线零售商的许多信息，这样也可能会泄露许多客户的敏感信息，如地址、购买的产品以及支付细节。

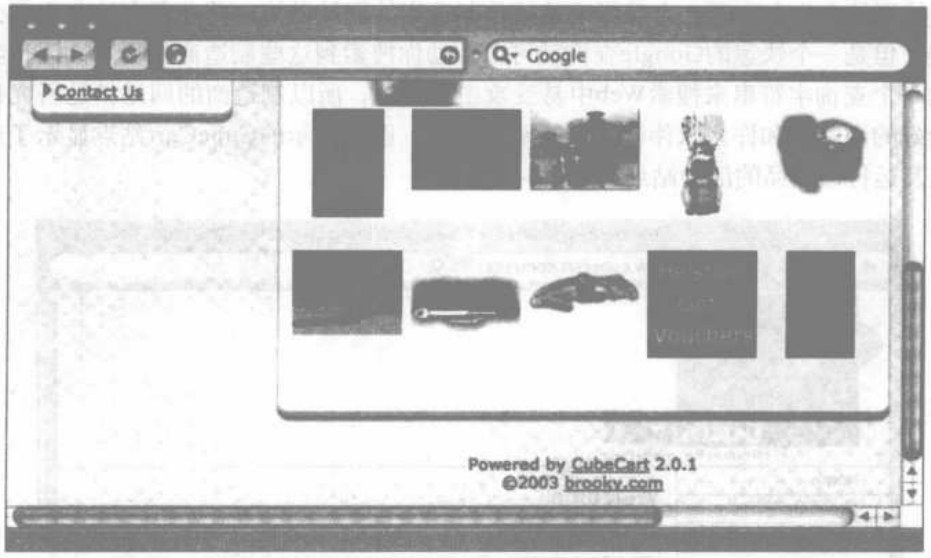


图6-13 “Powered by”标记在查找Web应用程序时是通用的查询材料

图6-14 一个搜索易受攻击的CubeCart站点的查询

6.6.2 利用源代码搜索目标

在某些情况下，构建一个好的查询并不容易，虽然我们会看到，组合查询和构建中的查询几乎相同。虽然这种方法更为冗长（可以灵活的剪裁），但是它却表明了一个典型的检测用于搜索易受攻击目标的有效查询的过程。在这里我们来看看黑客是如何使用程序的源代码在Google

中搜索相应的软件的。例如，如图6-15所示的是一个针对CuteNews程序发布的安全公告。

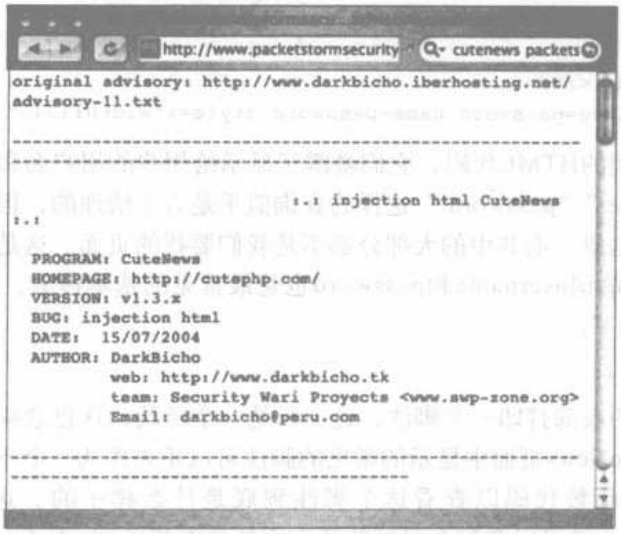


图6-15 CuteNews公告

正如在安全公告中解释的那样，攻击者可以利用一个精心构造的URL从一个易受攻击的目标中获取信息。为了找到最好的用于搜索可能易受攻击的目标的查询字符串，我们可以浏览软件制造商的网站以查找相应的软件的源代码。在无法找到源代码的情况下，攻击者可能会选择下载该软件，然后在他能控制的机器上运行以获得可能的查询字符串。在这个例子中，CuteNews的1.3.1版本恰好可以从作者的网站中下载下来。

在下载完软件并且适当地解压缩之后，第一件事就是观察显示给浏览者的主页。这个例子中的特殊软件使用PHP文件来生成网页。图6-16给出了顶级CuteNews目录的内容。

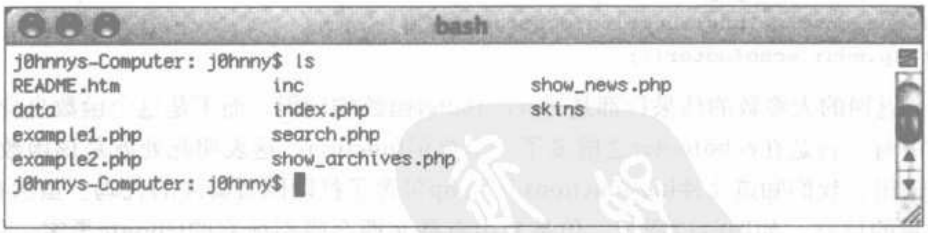


图6-16 CuteNew 1.3.1所包含的文件

在这个包的主目录中所列出的所有文件中，index.php最有可能是顶级页面。对index.php整个文件进行分析可以发现第156行最吸引人。

```
156 // If User is Not Logged In, Display The Login Page
```

第156行给出了一个典型的有用的注释。这个注释表明该段代码会显示一个登录页面。在登录页面中继续查找，即可看到第173～178行：

```
173 <td width=80>Username: </td>
```

```
174 <td><input tabindex=1 type=text
name=username value='$lastusername' style=\"width:134\"></td>
175 </tr>
176 <tr>
177 <td>Password: </td>
178 <td><input type=password name=password style=\"width:134\"></td>
```

这几行代码是典型的HTML代码，它们暴露了显示给用户的用户名和口令提示。基于这段代码，诸如“username:”“password:”这样的查询似乎是合乎情理的，但是事实上这个查询却返回了超过上百万条结果，而其中的大部分都不是我们要找的页面。这是因为查询中的冒号被Google忽略了，而且单词username和password也是最常见的基本搜索。我们的查找将继续到index.php的第191行如下：

```
191  echofooter();
```

这一行将在网页的底部打印一个脚注。这一行是一个函数，这也意味着它在整个程序中被多次调用。在许多CuteNews页面中显示的常见的脚注可以用来作为一个不错的基本查询。我们需要找到echofooter函数代码以查看这个脚注到底是什么样子的。运行一个例如grep -r echofooter *这样的命令就可以在每个目录的所有文件中查找单词echofooter。这会返回许多结果，简化输出如下：

```
j0hnnys-Computer: j0hnny$ grep -r echofooter *
inc/about.mdu: echofooter();
inc/addnews.mdu: echofooter();
inc/categories.mdu:echofooter();
inc/editnews.mdu: echofooter();
inc/editnews.mdu: echofooter();
inc/editusers.mdu: echofooter();
inc/functions.inc.php: echofooter();
inc/functions.inc.php:// Function: echofooter
inc/functions.inc.php:function echofooter(){
inc/help.mdu: echofooter();
```

该命令返回的大多数的结果行都是对echofooter函数的调用，而不是这个函数自身的定义。但是，其中有一行是在echofooter之前多了一个单词function，这表明此处就是该函数的定义。根据这段输出，我们知道文件inc/functions.inc.php包含了打印网页脚注的代码。虽然在这个函数中有大量的信息，如图6-17所示，但是其中有些东西会吸引所有的Google黑客。例如，第168行表明应首先打印出版权，然后在脚注中打印词组“Powered by”。

一个类似于“Powered by”这样的词组在搜索特定的目标时会十分有用，这是因为它们的高度唯一性。在“Powered by”词组之后是一个连接到http://cutephp.com/cutenews/的链接以及字符串$config_version_name，它列出了CuteNews程序的版本号。为了给Google提供一个非常特殊的“Powered by”搜索，攻击者必须猜测会显示的确切版本号（回忆一下，版本号为1.3.1）或者在源代码中查找实际版本号。另外，grep命令可以帮助我们快速地搜索这一字符串。我们可以直接搜索这个字符串或者在字符串后面加上一个等于号（=）以查找它在何处定义。命令

grep -r "\$config_version_name =" *可以实现这个搜索：

```
johnny-longs-g4 root$ grep -r "\$config_version_name =" *
inc/install.mdu:\$config_version_name = "CuteNews v1.3.1";
inc/options.mdu: fwrite($handler, "<?PHP \n\n//System
Configurations\n\n\$config_version_name =
\"$config_version_name\";\n\n\$config_version_id = $config_version_id;\n\n");
johnny-longs-g4 root$
```

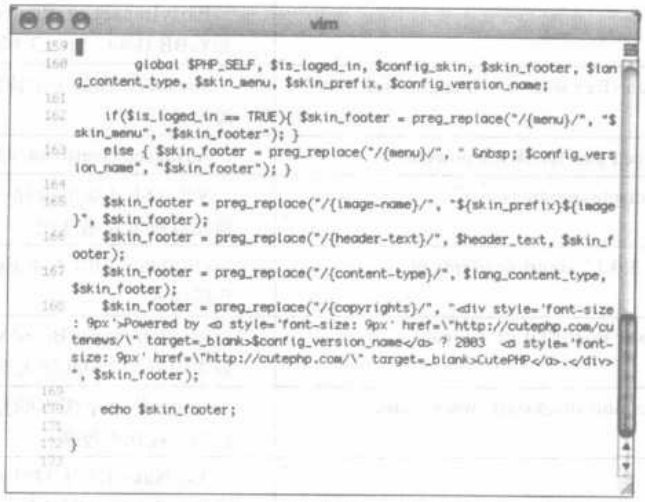


图6-17 echofooter函数泄露了可能的查询字符串

正如此处所显示的，版本号为CuteNews v1.3.1。把这两块脚注合并在一起就可以创建出一条特别的字符串："Powered by CuteNews v1.3.1"。它同样也创建了一条非常棒的Google查询，如图6-18所示。这个非常特殊的查询可以返回近乎完美的结果，结果表明将近500个站点运行了易受攻击的CuteNews 1.3.1版本的软件。

图6-18 一个完整的漏洞搜索

有太多这种技术的例子，以至于无法把它们都列出来，但是考虑到本书后面的传统风格，表6-4列出了一些设计用来搜索运行了可能易受攻击的Web应用程序的目标的例子。这些例子都取自Google Hacking数据库。

表6-4 取自GHDB的易受攻击的Web应用程序例子

Google查询	漏洞说明
inurl:custva.asp	EarlyImpact productcart包含多个漏洞版本，如YaBB Gold - Sp 1.3.1等
"Powered by mnoGoSearch–free web search engine software"	mnGoSearch的多个特定版本包含一个缓冲区溢出漏洞
intitle:guestbook "advanced guestbook 2.2 powered"	Advanced Guestbook v2.2有一个SQL注入漏洞
filetype:asp inurl: "shopdisplayproducts.asp"	VP-ASP（虚拟编程—ASP）版本包含多个跨站点脚本攻击漏洞
"Powered by: vBulletin * 3.0.1" inurl:newreply.php	vBulletin 3.0.1不能正确清除输入，允许恶意代码注入
"Powered by Invision Power Board(U) v1.3 Final"	Invision Power Board v.13 Final在它的ssi.php脚本中有一个SQL注入漏洞
"powered by sphider" -exploit-ihackstuff -www.cs.ioc.ee	spider的多个版本的搜索引擎脚本都允许任意的远程代码包含
inurl:gotoURL.asp?url=	Asp Nuke 1.2、1.3和1.4版不能清除输入变量，因此会生成一个SQL注入问题
inurl:comersus_message.asp	Comersus Open Technologies Comersus Cart的多个特定版本都含有包括XSS在内的多个漏洞
ext:pl inurl:cgi intitle: "FormMail *" - "*Referrer" - "* Denied" -sourceforge -error -cvs -input	FormMail的多个特定版本都有配置问题和非法引用者检查
inurl: "dispatch.php?atknodetype" | inurl:class.at	Achievo的多个特定版本允许远程代码执行
"Powered by Gallery v1.4.4"	Gallery v1.44包含可能允许远程攻击者执行恶意脚本的漏洞
"Powered by Ikonboard 3.1.1"	IkonBoard 3.1.1包含薄弱的用户输入验证，允许攻击者评估任意的Perl并运行任意的命令
inurl:/cgi-bin/index.cgi inurl:topics inurl:viewca	WebAPP的多个特定版本包含一个严重的反向目录遍历漏洞
inurl: "/becommunity/community/ index.php?pageurl="	E-market的多个特定版本允许任意的代码注入
"Powered *: newtelligence" ("dasBlog 1.6" | "dasBlog 1.5" | "dasBlog 1.4" |" dasBlog 1.3")	据称，DasBlog 1.3-1.6易受HTML注入的影响
"Powered by DCP-Portal v5.5"	DCP-Portal 5.5易受SQL注入
"FC Bigfeet" -inurl:mail	TYPO3的多个特定版本允许演试登录
filetype:cgi inurl:tseekdir.cgi	Turbo Seek的多个特定版本允许文件枚举
filetype:php inurl:index.php inurl: "module=subjects" inurl: "func=*" (listpages| viewpage | listcat)	PostNuke Modules Factory Subjects模块的多个特定版本都包含SQL注入漏洞

（续）

Google查询	漏洞说明
filetype:cgi inurl:pdesk.cgi	PerlDesk的多个特定版本都包含多个漏洞
“Powered by IceWarp Software” inurl:mail	IceWarp Web Mail 5.2.8之前的版本都包含多个输入验证漏洞
intitle: “MRTG/RRD” 1.1* (inurl:mrtg.cgilinurl:14all.cgitraffic.cgi)	MRTG 1.1.*版本允许部分文件枚举
inurl:com_remository	Mambo的ReMOSitory模块的多个特定版本都有SQL注入漏洞倾向
intitle: “WordPress > * > Login form” inurl: “wp-login.php”	WordPress的多个特定版本都包含XSS漏洞
inurl: “comment.php?serendipity”	Serendipity的多个特定版本都易受SQL注入的攻击
“Powered by AJ-Fork v.167”	AJ-Fork v.167易泄露完整路径
“Powered by Megabook *” inurl :guestbook.cgi	MegaBook的多个特定版本都有多个HTML注入漏洞倾向
“Powered by yappa-ng”	yappa-ng的多个特定版本都包含认证漏洞
“Active Webcam Page” inurl:8080	Active WebCam的多个特定版本都包含目录遍历以及多个XSS漏洞
“Powered by A-CART”	A-CART的多个特定版本都允许客户数据库下载
“Online Store - Powered by ProductCart”	ProductCart的多个特定版本都包含多个SQL注入漏洞
“Powered by FUDforum”	FUDforum的多个特定版本都包含SQL注入问题以及文件操作问题
“BosDates Calendar System” “powered by BosDates v3.2 by BosDev”	BosDates 3.2有SQL注入漏洞
intitle: “EMUMAIL - Login” “Powered by EMU Webmail”	EMU Webmail 5.0和5.1.0都包含XSS漏洞
intitle: “WebJeff - FileManager” intext: “login” intext:PasslPAsse	WebJeff-Filemanager 1.x有一个目录遍历漏洞
inurl: “messageboard/Forum.asp?”	GoSmart Message Board的多个特定版本都易受SQL注入和XSS问题的影响
“1999-2004 FuseTalk Inc” -site:fusetalk.com	Fusetalk论坛第4版易受XSS攻击
“2003 DUware All Rights Reserved”	多个DUware产品的多个特定版本都易受SQL注入及HTML注入攻击
“This page has been automatically generated by Plesk Server Administrator”	Plesk Server Administrator（PSA）的多个特定版本都包含输入合法性错误
inurl:ttt-webmaster.php	Turbo traffic trader Nitro v1.0易受多个漏洞攻击
“Copyright ÃÂ© 2002 Agustin Dondo Scripts”	CoolPHP的多个特定版本都易受多个漏洞攻击
“Powered by CubeCart”	CubeCart 2.0.1有一个完整路径泄露和SQL注入问题
“Ideal BB Version: 0.1” -idealbb.com	据称，Ideal BB 0.1易受多个未指定的输入合法性漏洞攻击

（续）

Google查询	漏洞说明
“Powered by YaPig V0.92b”	据称，YaPiG v0.92b包含HTML注入漏洞
inurl: “/site/articles.asp?idcategory=”	Dwc_Articles的多个特定版本都易受可能的SQL注入
filetype:cgi inurl:nbmember.cgi	Netbilling nbmember.cgi的多个特定版本都包含一个信息泄漏漏洞
“Powered by Coppermine Photo Gallery”	Coppermine Photo Gallery Coppermine Photo Gallery 1.0，1.1，1.2，1.2.1，1.3，1.3.1和1.3.2都包含一个允许用户为每个照片投多个选票的设计错误
“Powered by WowBB” -site:wowbb.com	据称，WowBB的多个特定版本都会受多个输入合法性漏洞的影响
“Powered by ocPortal” -demo-ocportal.com	ocPortal的多个特定版本都会受一个远程的包含漏洞的文件的影响
“inurl:” slxweb.dll”	SalesLogix的多个特定版本都包含身份鉴定漏洞
“Powered by DMXReady Site Chassis Manager” -site: dmxready.com	DMXReady Site Chassis Manager的多个特定版本都易受两个远程的可利用输入合法性漏洞的攻击
“Powered by My Blog” intext: “FuzzyMonkey.org”	FuzzyMonkey My Blog的1.15-1.20版本都易受多个输入合法性漏洞的攻击
inurl:wiki/MediaWiki	据称，MediaWiki的1.3.1-6版本都易受跨站点脚本漏洞的攻击。这个问题会引发用户供给数据的不适合的安全处理
“inurl:/site/articles.asp?idcategory=”	Dwc_Articles v1.6之前的版本都易受SQL注入漏洞的攻击
“Enter ip” inurl: “php-ping.php”	php-ping的多个特定版本都易受远程命令执行漏洞的攻击
intitle:welcome.to.horde	Horde Mail的多个特定版本都易受几个漏洞的攻击
“BlackBoard 1.5.1-f | ÃÂ© 2003-4 by Yves Goergen”	据称，BlackBoard Internet Newsboard System v1.5.1易受包含漏洞的远程文件的攻击
inurl: “forumdisplay.php” + “Powered by: vBulletin Version 3.0.0..4”	据称，vBulletin 3.0.0.4易受远程SQL注入漏洞的攻击
inurl:technote inurl:main.cgi *filename=*	Technote的多个特定版本都易受远程命令执行漏洞的攻击
“running: Nucleus v3.1” -.nucleuscms.org -demo	据称，多个不确定的漏洞都会影响Nucleus CMS v3.1。
“driven by: ASP Message Board”	Infuseum ASP Message Board 2.2.1c易受多个不确定的漏洞的攻击

（续）

Google查询	漏洞说明
“Obtenez votre forum Aztek” -site:forum-aztek.com	Atztek Forum的多个特定版本都易受多个输入合法性漏洞的攻击
intext:(“UBB.threads Ã¢â©Ã¢â 6.2”|“UBB.threadsÃ¢â? ŽÃ¢ 6.3”) intext: “You * not logged *” -site:ubbcentral.com	UBB.Threads 6.2.*-6.3.*包含一个单个字符强力漏洞攻击
inurl:/SiteChassisManager/	DMXReady Site Chassis Manager的多个特定版本都易受SQL和XSS漏洞的攻击
inurl:directorypro.cgi	DirectoryPro的多个特定版本都易受目录遍历漏洞的攻击
inurl:cal_make.pl	PerlCal的多个特定版本都允许远程攻击者访问那些位于常态绑定的HTML根目录之外的文件
“Powered by PowerPortal v1.3”	据称，PowerPortal 1.3易受SQL注入攻击
“powered by minibb” -site:www.minibb.net -intext:1.7f	miniBB早于1.7f的版本都易受远程SQL注入的攻击
inurl: “/cgi-bin/loadpage.cgi?user_id=”	EZshopper的多个特定版本都允许目录遍历
intitle: “View Img” inurl:viewimg.php	viewing.php脚本的多个特定版本都不能正确地确定用户提供的path变量的输入合法性
+ “Powered by Invision Power Board v2.0.0.2”	Inivision Power Board v2.0.0-2.0.2易受SQL注入漏洞的攻击
+ “Powered by phpBB 2.0.6..10” -phpbb.com -phpbb.pl	phpbb 2.0.6-20.10易受某个SQL注入的漏洞攻击
ext:php intext: “Powered by phpNewMan Version”	PHP News Manager的多个特定版本都易受目录遍历问题的攻击
“Powered by WordPress” -html filetype:php -demo -wordpress.org -bugtraq	WordPress的多个特定版本都易受少部分SQL注入查询的攻击
intext:Generated.by.phpix.1.0?inurl:$mode=album	PHPix v1.0易受某个目录遍历漏洞的攻击
inurl:citrix/metaframexp/default/login.asp? ClientDetection=On	Citrix的多个特定版本的Web界面的普遍使用版本都包含一个XSS漏洞
“SquirrelMail version 1.4.4” inurl:src ext:php	SquirrelMail v1.4.4包含一个包含漏洞
“IceWarp Web Mail 5.3.0” “Powered by IceWarp”	IceWarp Web Mail 5.3.0包含多个跨站点脚本和HTML注入漏洞
“Powered by MercuryBoard [v1”	MercuryBoard v1包含一个不确定的漏洞
“delete entries” inurl: admin/delete.asp	AspJar的多个特定版本都包含一个缺陷，该缺陷允许恶意用户删除任意信息
allintitle:aspjar.com guestbook	ASPJar来宾簿的多个特定版本都包含一个输入合法性的漏洞
“powered by CubeCart 2.0”	Brooky CubeCart v2.0易受多个未经适合的安全处理的用户提供数据的漏洞攻击
Powered.by:.vBulletin.Version ...3.0.6	据称，vBulletin 3.0.6易受任意PHP脚本代码执行漏洞的攻击

（续）

Google查询	漏洞说明
filetype:php intitle: “paNews v2.0b4”	据称，PaNews v2.0b4易受远程PHP脚本代码执行漏洞的攻击
“Powered by Coppermine Photo Gallery” (“v1.2.2 b” | “v1.2.1” | “v1.2” | “v1.1” | “v1.0”	Coppermine Photo Gallery versions 1.0，1.1，1.2，1.2.1和1.2.2b都易受多个输入合法性漏洞的攻击，其中的部分漏洞可能会导致任意的命令执行
powered.by.instaBoard.version.1.3	InstaBoard v1.3易受SQL注入的攻击
intext: “Powered by phpBB 2.0.13” inurl: “cal_view_month.php” linurl: “downloads.php”	带有已安装的Calendar Pro MOD的phpBB 2.0.13易受SQL注入攻击
intitle: “myBloggie 2.1.1..2—by myWebland”	myBloggie v2.1.1-2.1.2易受多个漏洞的攻击
intitle: “osTicket :: Support Ticket System”	osTicket的多个特定版本都包含多个漏洞
inurl:sphpblog intext: “Powered by Simple PHP Blog 0.4.0”	Simple PHP Blog v0.4.0易受包含完整路径泄漏、XSS和其他泄漏在内的多个攻击
intitle: “PowerDownload” (“PowerDownload v3.0.2 Ã, Â©” | “PowerDownload v3.0.3 Ã, Â©”)-site:powerscripts.org	PowerDownload 3.0.2和3.0.3版本都包含一个远程执行漏洞
“portailphp v1.3” inurl: “index.php?affiche” inurl: “PortailPHP” -site:safari-msi.com	PortailPHP v1.3易受SQL注入漏洞攻击
+intext: “powered by MyBulletinBoard”	MyBB <= 1.00 RC4包含一个SQL注入漏洞
intext: “Powered by flatnuke-2.5.3” + “Get RSS News” -demo	FlatNuke 2.5.3包含多个漏洞
intext: “Powered By: Snitz Forums 2000 Version 3.4.00..03”	Snitz Forum 2000 v 3.4.03与早期的版本都易受包含XSS在内的很多漏洞的影响
inurl: “/login.asp?folder=” “Powered by: i-Gallery 3.3”	i-Gallery 3.3（可能版本更早）都易受包含目录遍历在内的很多漏洞的影响
intext: “Calendar Program Ã, Â©Copyright 1999 Matt Kruse” “Add an event”	CalendarScript的多个特定版本都易受HTML注入的影响
“powered by PhpBB 2.0.15” -site:phpbb.com	phpBB 2.0.15 Viewtopic.PHP包含一个远程代码执行漏洞
inurl:index.php fees shop link.codes merchantAccount	EPay Pro version 2.0易受目录遍历问题的影响
intitle: “blog torrent upload”	Blog Torrent的多个特定版本都包含一个口令泄露问题
“Powered by Zorum 3.5”	Zorum 3.5包含一个远程代码执行漏洞的漏洞
“Powered by FUDForum 2.6” -site:fudforum.org -johnny .ihackstuff	FUDforum 2.6易受远程任意PHP文件上传漏洞的影响
intitle: “Looking Glass v20040427” “When verifying	Looking Glass v20040427允许任意命令执行和跨站点脚本编程
phpLDAPadmin intitle: phpLDAPadmin filetype:php inurl: tree.php | inurl:login.php | inurl: donate.php (0.9.6 | 0.9.7)	phpLDAPadmin 0.9.6 - 0.9.7/alpha5（以及多个更早的版本）包含系统泄漏、远程代码执行以及XSS漏洞

（续）

Google查询	漏洞说明
“powered by ITWorking”	SaveWebPortal 3.4包含一个远程代码执行、绕过管理员核查以及远程文件包含的漏洞
intitle:guestbook inurl:guestbook “powered by Adva”	Advanced Guestbook的多个特定版本都易受HTML注入漏洞的影响
“Powered by FUDForum 2.7” -site:fudforum.org -johnny.ihackstuff	FUDforum 2.7易受远程任意PHP文件上传漏洞的影响
inurl:chitchat.php “choose graphic”	Cyber-Cats ChitCHat 2.0包含多个漏洞
“Calendar programming by AppIdeas.com” filetype:php bypass an/XSS	phpCommunityCalendar 4.0.3（和多个更早的版本）允许SQL注入、绕过登录以及XSS
“Powered by MD-Pro” | “made with MD-Pro”	MAXdev MD-Pro 1.0.73（和以前的多个版本）允许远程代码执行、XSS和路径泄露
“Software PBLang” 4.65 filetype:php	PBLang 4.65（和以前的多个版本）允许远程代码执行、管理员证书泄露、系统信息揭露、XSS和路径泄露
“Powered by and copyright class-1” 0.24.4	Class-1 Forum Software v 0.24.4允许远程代码执行
“Powered by AzDg” (2.1.3 | 2.1.2 | 2.1.1)	AzDGDatingLite V 2.1.3（和多个更早的版本）允许远程代码执行
“Powered by: Land Down Under 800” | “Powered by: Land Down Under 801” - www.neocrome.net	Land Down Under 800和900易受HTML注入漏洞攻击
“powered by Gallery v” “[slideshow]” | “images” inurl:gallery	Gallery的多个特定版本都易受脚本注入漏洞攻击
intitle:guestbook inurl:guestbook “powered by Advanced guestbook2.*” “sign the Guestbook”	Advanced Guestbook v2.*易受HTML注入漏洞攻击
“Copyright 2004 Ã, Â© Digital Scribe v.1.4”	Digital Scribe v1.4允许绕过登录、SQL注入以及远程代码执行
“Powered by PHP Advanced Transfer Manager v1.30”	PHP Advanced Transfer Manager v1.30允许底层系统泄露、远程命令执行和跨站点脚本编程
“Powered by CuteNews”	CuteNews 1.4.0（和多个更早的版本）允许远程代码执行
“Powered by GTChat 0.95” + “User Login” + “Remember my login information”	GTChat v0.95包含一个远程拒绝服务漏洞
intitle: “WEB//NEWS Personal Newsmanagement” intext:” Ã, Â© 2002-2004 by Christian Scheb—Stylemotion.de” + “Version 1.4 ” + “Login”	WEB//NEWS 1.4易受多个SQL注入漏洞的攻击
“Mimicboard2 086” + “2000 Nobutaka Makino” + “password” + “message” inurl:page=1	Mimicboard2 v086易受多个HTML注入漏洞的攻击
“Maintained with Subscribe Me 2.044.09p” + “Professional” inurl: “s.pl”	Subscribe Me Pro 2.0.44.09p易受目录遍历漏洞的攻击
“Powered by autolinks pro 2.1” inurl:register.php	AutoLinksPro v2.1包含一个远程PHP文件包含漏洞

（续）

Google查询	漏洞说明
“CosmoShop by Zaunz Publishing” inurl: “cgi-bin/cosmoshop /lshop.cgi” -johnny.ihackstuff.com -V8.10.106 -V8.10.100 - V.8.10.85 -V8.10.108 -V8.11*	Cosmoshop versions 8.10.85，8.10.100，8.10.106，8.10.108和8.11*都易受SQL注入和明码电文口令枚举的攻击
“Powered by Woltlab Burning Board” - “2.3.3” - “v2.3.3” - “v2.3.2” - “2.3.2”	Woltlab Burning Board 2.3.32版本以及2.3.3版本都易受SQL注入的攻击
intitle: “PHP TopSites FREE Remote Admin”	PHP TopSites的多个特定版本为远程用户泄露了配置数据
Powered by PHP-Fusion v6.00.109 Ã, Â© 2003-2005. -php-fusion.co.uk	PHP-Fusion v6.00.109易受SQL注入以及管理员证书泄露的攻击
“Powered By: lucidCMS 1.0.11”	Lucid CMS 1.0.11有SQL注入和绕过登录漏洞
“News generated by Utopia News Pro” | “Powered By: Utopia News Pro”	Utopia News Pro 1.1.3（和多个更早的版本）都包含SQL注入和XSS漏洞
intitle:Mantis “Welcome to the bugtracker” “0.15 | 0.16 | 0.17 | 0.18”	Mantis versions 0.19.2或者更低的版本都包含XSS和SQL注入漏洞
“Cyphor (Release:” -www.cynox.ch	Cyphor 0.19（和多个更早的版本）都允许SQL注入、公告板接管和XSS
“Welcome to the versatileBulletinBoard” | “Powered by versatileBulletinBoard”	VersatileBulletinBoard V1.0.0 RC2（和多个更早的版本）都包含多个漏洞
inurl:course/category.php | inurl:course/info.php | inurl: iplookup/ipatlas/plot.php	Moodle <=1.6允许盲目SQL注入
“Powered by XOOPS 2.2.3 Final”	XOOPS 2.2.3允许任意本地文件包含
inurl: “wfdownloads/viewcat.php?list=”	XOOPS WF_Downloads（2.05）模块允许SQL注入
“This website was created with phpWebThings 1.4”	phpWebThings 1.4包含多个漏洞
“Copyright 2000 - 2005 Miro International Pty Ltd. All rights reserved” “Mambo is Free Software released”	Mambo 4.5.2x允许远程命令执行
(“Skin Design by Amie of Intense”)|(“Fanfiction Categories” “Featured Stories”)|(“default2, 3column,Romance, eFiction”)	eFiction <=2.0包含多个漏洞
“Powered by UPB” (b 1.0)|(1.0 final)|(Public Beta 1.0b)	UPB b1.0, 1.0 final和Public Beta版本都包含几个漏洞
“powered by GuppY v4” |” Site crÃfÂ©ÃfÂ©avec GuppY v4”	Guppy <= 4.5.9允许远程代码执行和任意包含
“Powered by Xaraya” “Copyright 2005”	Xaraya <=1.0.0 RC4包含拒绝服务
“This website powered by PHPX” -demo	PhpX <= 3.5.9允许SQL注入和绕过登录
“Based on DoceboLMS 2.0”	DoceboLMS 2.0包含多个漏洞
“2005 SugarCRM Inc. All Rights Reserved” “Powered By SugarCRM”	Sugar Suite 3.5.2a & 4.0beta允许远程代码执行
“Powered By phpCOIN 1.2.2”	PhpCOIN 1.2.2允许任意远程/本地包含、盲目SQL注入和路径泄露
intext: “Powered by SimpleBBS v1.1” *	SimpleBBS v1.1包含一个允许攻击者运行SQL注入攻击的缺陷

（续）

Google查询	漏洞说明
“Site powered By Limbo CMS”	Limbo Cms <= 1.0.4.2允许远程代码执行
intext:“Powered by CubeCart 3.0.6” intitle:“Powered by CubeCart”	CubeCart 3.0.6允许远程命令执行
intext:“PhpGedView Version” intext:“final - index” -inurl:demo	PHPGedView <=3.3.7允许远程命令执行
intext:“Powered by DEV web management system” -dev-wms.sourceforge.net -demo	DEV cms <=1.5允许SQL注入
intitle:“phpDocumentor web interface”	Php Documentor < = 1.3.0 rc4允许远程代码执行
inurl:install.pl intitle:GTchat	Gtchat多个特定版本都允许未授权配置更改
intitle:“4images - Image Gallery Management System” and intext:“Powered by 4images 1.7.1”	4Images v1.7.1允许远程代码执行
(intitle:“metaframe XP Login”)|(intitle:“metaframe Presentation server Login”)	Metaframe Presentation Server的多个特定版本都允许未授权的管理员访问
“Powered by Simplog”	Simplog v1.0.2允许目录遍历和XSS
“powered by sblog” + “version 0.7”	Sblog v0.7允许HTML注入
“Thank You for using WPCeasy”	WPC.easy的多个特定版本都允许SQL注入
“Powered by Loudblog”	LoudBlog <= 0.4包含一个任意的远程包含漏洞
“This website engine code is copyright” “2005 by Clever Copy” -inurl:demo	Clever Copy <= 3.0允许SQL注入
“index of” intext:fckeditor inurl: fckeditor	FCKEditor script 2.0和2.2包含多个漏洞
“powered by runcms” -runcms.com-runcms.org	Runcms versions <=1.2易受任意的远程包含攻击
(intitle:“Flyspray setup” | “powered by flyspray 0.9.7”) -flyspray.rocks.cc	Flyspray v0.9.7包含多个漏洞
intext:“LinPHA Version” intext:“Have fun”	Linpha <=1.0允许任意的本地包含
(“powered by nocc” intitle:“NOCC Webmail”) -site:sourceforge.net -Zoekinalles.nl -analysis	NOCC Webmail的多个特定版本允许任意的本地包含、XSS和可能的远程代码执行
intitle:“igenus webmail login”	Igenus webmail允许本地文件枚举
“powered by 4images”	4images <= 1.7.1允许远程代码执行
intext:“Powered By Geeklog” -geeklog.net	Geeklog的多个特定版本包含多个漏洞
intitle:admbook intitle:version filetype:php	Admbook 1.2.2版本允许远程执行
WEBalbum 2004-2006 duda -ihackstuff -exploit	WEBalbum 2004-2006包含多个漏洞
intext:“powered by gcards” -ihackstuff -exploit	Gcards <=1.45包含多个漏洞
“powered by php icalendar” -ihackstuff -exploit	php iCalendar <= 2.21允许远程命令执行
“Powered by XHP CMS” -ihackstuff -exploit -xhp.targetit.ro	XHP CMS 0.5允许远程命令执行
inurl:*.exe ext:exe inurl:/*cgi*/	多个CGI-bin可执行文件允许XSS和HTML注入
“powered by claroline” -demo	Claroline e-learning platform <= 1.7.4包含多个漏洞
“PhpCollab . Log In” | “NetOffice . Log In” |(intitle:“index.of.” intitle: phpcollab|netoffice inurl:phpcollab|netoffice -gentoo)	PhpCollab 2.x / NetOffice 2.x允许SQL注入

（续）

Google查询	漏洞说明
intext: “2000-2001 The phpHeaven Team” -sourceforge	PHPMyChat <= 0.14.5包含一个SQL注入漏洞
“2004-2005 ReloadCMS Team.”	ReloadCMS <= 1.2.5stable允许XSS和远程命令执行
intext: “2000-2001 The phpHeaven Team” -sourceforge	phpHeaven的多个特定版本允许远程命令执行
inurl:server.php ext:php intext: “No SQL” -Released	PHPOpenChat的多个特定版本包含多个漏洞
intitle:PHPOpenChat inurl: “index.php?language=”	PHPOpenchat的多个特定版本允许SQL注入以及信息披露
“powered by phplist” | inurl: “ lists/?p=subscribe” | inurl: “lists/index. php?p=subscribe” -ubbi -bugs+phplist -tincan.co.uk	PHPList 2.10.2允许任意的本地文件包含
inurl: “extras/update.php” intext: mysql.php -display	osCommerce的多个特定版本都允许本地文件枚举
inurl:sysinfo.cgi ext:cgi	Sysinfo 1.2.1允许远程命令执行
inurl:perldiver.cgi ext:cgi	perldiver.cgi的多个特定版本都允许XSS
inurl:tmssql.php ext:php mssql pear adodb -cvs -akbk	tmssql.php的多个特定版本都允许远程代码执行
“powered by php photo album” | inurl: “main.php?cmd=album” -demo2 -pitanje	PHP相册的多个特定版本都允许本地文件枚举以及远程漏洞利用
inurl:resetcore.php ext:php	e107的多个特定版本都包含多个漏洞
“This script was created by Php-ZeroNet” “Script. Php-ZeroNet”	Php-ZeroNet v 1.2.1包含多个漏洞
“You have not provided a survey identification num	PHP Surveyor 0995允许SQL注入
intitle: “HelpDesk” “If you need additional help, please email helpdesk at”	PHP Helpdesk 0.6.16允许任意数据的远程执行
inurl:database.php | inurl:info_db.php ext:php “Database V2.*” “Burning Board *”	Woltlab Burning Board 2.x包含多个漏洞
intext: “This site is using phpGraphy” | intitle: “my phpgraphy site”	phpGraphy 0911允许XSS以及拒绝服务
intext: “Powered by PCPIN.com” -site:pcpin.com ®Cihackstuff - “works with” -findlaw	PCPIN Chat的多个特定版本都允许SQL注入、绕过登录以及任意的本地包含
intitle: “X7 Chat Help Center” | “Powered By X7 Chat” -milw0rm -exploit	X7 Chat <=2.0允许远程命令执行
allinurl:tseekdir.cgi	tseekdir.cgi的多个特定版本都允许本地文件枚举
Copyright. Nucleus CMS v3.22 . Valid XHTML 1.0 Strict. Valid CSS. Back to top -demo - “deadly eyes”	Nucleus 3.22 CMS允许任意的远程文件包含
“powered by pppblog v 0.3.(.)”	pppblog 0.3.x允许系统信息泄露
“Powered by PHP-Fusion v6.00.110” | “Powered by PHP-Fusion v6.00.2.” | “Powered by PHP-Fusion v6.00.3.” -v6.00.400 - johnny.ihackstuff	PHP-Fusion 6.00.3和6.00.4都包含多个漏洞
intitle: “XOOPS Site” intitle: “Just Use it!” | “powered by xoops (2.0)|(2.0.....)”	XOOPS 2.x允许文件改写

（续）

Google查询	漏洞说明
inurl:wp-login.php +Register Username Password "remember me" -echo -trac -footwear	Wordpress 2.x允许远程命令执行
"powered by ubbthreads"	ubbthreads的多个特定版本都很容易受文件包含的攻击
"Powered by sendcard - an advanced PHP e-card program" -site:sendcard.org	Sendcard的多个特定版本都允远程命令执行以及SQL注入
"powered by xmb"	XMB <=1.9.6 Final允许远程命令执行以及SQL注入
"powered by minibb forum software"	minibb论坛软件的多个特定版本都允许任意的远程文件包含
inurl:eStore/index.cgi?	eStore的多个特定版本都允许目录遍历

注：本表以及相关GHDB条目均由该社区的多个成员提供。在此我们将他们的名字按提供的条目的数量分别列举如下：rgod (85)，Joshua Brashars (18)，klouw (18)，Fr0zen (10)，MacUK (8)，renegade334 (7)，webby_guy (7)，CP (6)，cybercide (5)，jeffball55 (5)，JimmyNeutron (5)，murfie (4)，FiZiX (4)，sfd (3)，ThePsyko (2)，wolveso (2)，Deeper (2)，HaVoC88 (2)，l0om (2)，Mac (2)，rar (2)，GIGO (2)，urban (1)，demonio (1)，ThrowedOff (1)，plaztic (1)，Vipsta (1)，golfo (1)，xlockex (1)，hevnsnt (1)，none90810 (1)，hermes (1)，blue_matrix (1)，Kai (1)，goodvirus (1)，Ronald MacDonald (1)，ujen (1)，Demonic_Angel (1)，zawa (1)，Stealth05 (1)，maveric (1)，MERLiiN (1)，norocosul_alex R00t (1)，abinidi (1)，Brasileiro (1)，ZyMoTiCo (1)，TechStep (1)，sylex (1)，QuadsteR (1)，ghooli (1)

6.6.3 利用CGI扫描搜索目标

一种最古老的，也是读者最为熟悉的搜索易受攻击的网站服务器的技术是使用CGI扫描器。这些扫描程序将会分析一个已知的“不好的”或者易受攻击的网站文件列表，并且在网站服务器中搜索这些文件。根据各种响应代码，扫描器就可以检测出这些可能易受攻击的文件是否存在。一个CGI扫描器可以在一个数据文件中列出易受攻击的文件和目录，如下所示：

```
/cgi-bin/userreg.cgi
/cgi-bin/cgiemail/uargg.txt
/random_banner/index.cgi
/random_banner/index.cgi
/cgi-bin/mailview.cgi
/cgi-bin/maillist.cgi
/iissamples/ISSamples/SQLQHit.asp
/iissamples/ISSamples/SQLQHit.asp
/SiteServer/admin/findvserver.asp
/scripts/cphost.dll
/cgi-bin/finger.cgi
```

与直接连接到目标服务器不同的是，攻击者可以利用Google来搜索可能含有这些可能易受攻击的文件和目标的服务器，只要把其中的每一行转换为一个Google查询即可。例如，第一行

的查询就是在一个叫作cgi-bin的目录下查找文件名为userreg.cgi的文件。把这个例子转换为Google查询十分容易，如图6-19所示，查询为inurl:/cgi-bin/userreg.cgi。

图6-19 一个CGI扫描风格的查询

这个搜索找到了很多个正在运行着可能易受攻击的程序的主机。当然，我们无法保证Google检测到的程序确实是易受攻击的程序。这些正是CGI扫描器程序的最大的问题。一个文件或者目录的存在并不一定意味着它是一个易受攻击的文件或目录。同时，网络中的这类扫描器程序也没有任何缺点，每个这类程序都为许多不同的Google查询提供了可能。

还有一些其他的办法可用于CGI类型的文件的查找。例如，filetype操作符可以用来查找实际的CGI程序，即使在cgi-bin父目录之外也可以用查询filetype:cgi inurl:userreg.cgi来查找。这个查询可以找到更多的结果，但是遗憾的是，这个查询甚至更为粗略，因为只有目录cgi-bin才意味着这是一个真正的CGI程序。根据服务器的配置，userreg.cgi还可能是一个文本文件，而不是一个可执行文件，如果不是十分困难的话，那么这时如何利用程序就变得十分有趣了。

一种更为粗略地查找这个文件的方法是通过目录列表，例如查询intitle:index.of userreg.cgi。这个查询在本书写作之时没有返回结果，其理由十分明显。在网络中，目录列表并不像URL一样常见，而且再加上包含这个特定文件的目录列表就更少了。

Google搜索背景知识

利用Google进行自动CGI扫描

显然，自动化是进行高效的Google搜索所必需的方式。Wiko（来自www.sensepost.com）和Gooscan（来自http://Johnny.ihackstuff.com）这两个工具都能够自动地进行Google和CGI扫描。Wikto使用Google API，而Gooscan则不然。关于这些工具的更详细的信息参见保护的相关章节。

6.7 总结

世界上有非常多的查找漏洞利用代码的方法以至于几乎不可能把它们全部进行分类。Google可以用来搜索含有公开的漏洞利用的网站，而且在某些情况下也可以在“私有的”站点中搜索某些工具。要记住的是许多漏洞利用并不发布在网上。新的（或者0day）漏洞利用常常只在某些圈子里流传，一个有能力的攻击者总是在最后才把他的或者她的工具放到公开的网页上。如果一个工具可以网上找到，那么这个工具很有可能是加密了的或者至少也应该是有口令保护的，以防止工具的传播，这样可以给社区以警报，以保护那些潜在的目标。这并不是说新的、未公布的漏洞利用不在网上，但是坦白来讲很容易和那些知晓内情的人建立联系。同样，收藏一个不错的公开漏洞利用站点列表也没有什么不妥，而且Google非常擅长于收集这些站点，只需要一些包含单词exploit、vulnerability或者vulnerable的简单查询就可实现。Google也可以通过代码中出现的某些字符串来搜索源代码。

用Google来搜索潜在的目标是一个相当直接的过程，这只不过需要一个能够表示易受攻击的Web应用程序的唯一字符串即可。有时，这些字符串可以从制造商提供的演示应用程序中获得。其他一些时候，攻击者可能需要下载这个产品或者源代码来查找用于Google查询中的字符串。不管哪一种方法，一个公开的Web应用程序漏洞利用公告，再加上Google的强大，都给防卫者留下了很少的时间来增强易受攻击的应用程序或者服务器的安全性。

6.8 快速查找解决方案

搜索漏洞利用代码

- 可以通过关注那些常见的字符串，例如exploit或者vulnerability，来搜索公开的漏洞利用站点。如果想让结果更为精确，可以给查询加上filetype操作符，以查找以特定程序设计语言所编写的漏洞利用。
- 既可以用filetype指定文件扩展名，也可以用源代码中的常见字符串，例如“include <stdio.h>”，来查找漏洞利用代码。

Google代码搜索

- Google代码搜索（www.google.com/codesearch）可以用来搜索程序内部的代码，但是它也可以用来搜索导致漏洞的编程错误。

搜索恶意软件

- Googel的二进制搜索功能可以用来剖析可执行文件，但是它也可以用来搜索Web上的可用恶意软件。可访问http://metasploit.com/research/misc/mwsearch参看H.D. Moore[⊖]的搜索引擎。

搜索易受攻击的目标

- 攻击者可以利用软件制造商提供的应用程序的演示程序中出现的字符串来搜索潜在的目标。

⊖ H、D、Moore是BreakingPoint Systems的安全研究主管，是一名杰出的软件漏洞研究者，被誉为“黑客英雄”。——编辑注

- 攻击者也可以下载并选择性安装易受攻击的产品，从而在应用程序显示的字符串中查找特殊的字符串。
- 不管字符串是如何获得的，它都能够很容易地转换为Google查询，彻底缩短了在公开漏洞公告之后，防卫人员花在加固站点安全上的时间。

6.9 网站链接

- www.sensepost.com/research/wikto/ Wikto，一个很棒的Google和Web扫描器。
- www.cirt.net/code/nikto.shtml Nikto，一个很棒的Web扫描器。
- http://packetstormsecurity.com/ 一个很棒的工具和漏洞利用站点。
- http://ilia.ws/archives/133-Google-Code-Search-Hackers-bestfriend.htm Ilia Alshanetsky。
- http://dhanjani.com/archives/2006/10/using_google_code_search_to_fi.html Nitesh Dhanjani。
- http://shiflett.org/blog/2006/oct/google-code-search-for-securityvulnerabilities Chris Shiflett。
- http://www.securityfocus.com/archive/107/447729/30/0 Stephen de Vries。

Michael Sutton的博客：

- http://portal.spidynamics.com/blogs/msutton/archive/2006/09/26/How-Prevalent-Are-SQL-Injection-Vulnerabilities_3F00_.aspx。
- http://portal.spidynamics.com/blogs/msutton/archive/2007/01/31/How-Prevalent-Are-XSS-Vulnerabilities_3F00_.aspx。
- Jose Nazario在Google Code Search安全部分的页面：http://monkey.org/~jose/blog/viewpage.php?page=google_code_search_stats。
- Aaron Campbell 的使用Google进行的静态代码分析：http://asert.arbornetworks.com/2006/10/static-code-analysis-using-google-codesearch/。
- HD Moore的恶意软件分析：http://metasploit.com/research/misc/mwsearch。

6.10 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：CGI扫描工具已经出现几年了，而且在许多黑客的贡献之下数据库规模也不断增长。Google只是取决于它的Googlebot所抓取的站点，用它能有什么好处？它给出的结果岂不是更少？

答：虽然这些情况都是事实，但是Google能够提供某一级别的匿名性，这是因为它能够使用strip=1参数来显示缓存页面，所以攻击者（黑帽或者白帽）的IP不会被服务器记录。查看第12章中的Nikto代码，它把Google的强大和Nikto数据库结合在一起！

问：有没有通用的用于搜索已知易受攻击的Web应用程序的技术？

答：根据安全公告中的信息，把INURL:["parameter="]、FILETYPE:[ext]和INURL:[scriptname]三者组合起来进行搜索。在某些情况下，应用程序的版本信息没有出现在目标页面上。如果你打算搜索版本信息，那么要记住每个数字都应看作为一个单词，因此对于Google而言1.4.2是三个单词。这样的话，很快就达到了Google搜索10个单词的上限。

同样也要记住的是，Google要给出一条结果必须事先已经抓取了相应的网站。如果不是这种情况的话，试试使用一个更为通用的搜索，例如“powered by XYZ”来搜索运行某一特定软件家族的页面。

第7章　简单有效的安全性搜索

7.1　简介

虽然本书通篇出现了上百条Google搜索，但是有时还需要知道有少部分搜索在任何时候都能得到很好的结果。针对安全方面的工作而言，我们来看看在安全性评估中相当有效的10个搜索，尤其是当这些搜索和site操作符组合使用时更为有效，所以我们把site操作符放在了搜索列表的第一位。随着对Google地不断了解，你可能会对这个列表进行添加、修改或是能删除其中的几个搜索，但是这里的搜索仅仅是作为你的前10位列表的基础。不再多说，我们一起来研究这些搜索。

7.1.1　site

在一次评估的信息收集阶段，site操作符绝对是最有价值的。至少可以这么说，如果把site操作符和一个主机名或者域名组合在一起查询时，返回的结果是足够让人感到惊奇的。但是，site操作符只是打算作基本搜索之用，而不一定要作为单独的搜索。当然，可以（并不完全鼓励）用这个查询把结果中的每一个页面都扫描一遍，但是在大多数情况下这显然是不切实际的。

即便如此，还是能够直接用site搜索来获取重要的信息。首先，要知道Google是按照页面排名高低的顺序列出结果的。换句话说，最流行的页面放在结果的最上面。这意味着你可以很快地知道网站中哪些是最值得关注的。这种信息的涵义是多种多样的，但是最基本地，通过查看最上面的结果你至少可以知道公众对网上事物的印象或者看法。除了site搜索自身的特殊性之外，它也可以帮助深入了解源自其他网站的链接。如果一个链接的文本说了某些关于“CompanyXYZ sucks!”的影响，那么很有可能在其他地方有人对CompanyXYZ不满。

在第5章中我们也看到，site搜索也可以用来收集关于目标主机或服务器的信息。使用简单的搜索缩简技术，我们能够很快地得到关于目标的网上信息。先来看一下如图7-1所示的简单例子site:nytimes.com-site:www.nytimes.com。

这个查询有效地搜索到了位于nytimes.com域上的页面，而不是www.nytimes.com域上的。可以一眼看出，图7-1给出了三个其他的域：theater.nytimes.com、www2.nytimes.com、salary.nytimes.com和realestate.nytimes.com。它们可能是主机也可能是子域。我们将对它们做进一步的调查以确定它们的身份。同样，要记得在放弃你的庞大的扫描结果之前验证Google结果的合法性。

图7-1 Site缩简可以得到域名

7.1.2 intitle:index.of

intitle:index.of是用于目录列表的通用搜索。目录列表充满了丰富的细节信息，这正如我们在第3章中看到的那样。针对一个目标实施intitle:index.of查询既快速又简单，而且还可能产生直接攻陷服务器的结果。

7.1.3 error | warning

在本书通篇中你都可看到，错误（error）消息能够泄露出目标的大量信息。这些错误消息能够提供目标正在运行的应用程序或者操作系统软件的内部信息，目标所处网络的结构，系统中的用户信息等，但是它们常常被忽略。错误消息不仅能泄露信息，而且它们也能够产生许多结果。单独进行该查询可以得到不错的结果，当与site查询绑定在一起时会得最佳的结果。如图7-2所示，一个（“for more information”|“not found”）（error | warning）查询就返回了很多条有趣的结果。

遗憾的是，有些错误消息实际上并不显示出单词error，如图7-3中所示的是用“accessdenied for user”“using password”查询得到的SQL错误。

这个错误页面泄露了用户名、文件名、路径信息、IP地址以及行号，但是单词error却没有在这个页面中的任何地方出现。几乎与错误消息能产生许多结果一样，应用程序也可以生成许多警告消息。但是，在某些情况下，单词warning被专门写到了页面的文本中，以警告Web用户发生或者即将发生一些很重要的事情。不管它们是如何生成的，包含这些单词的页面在评估中总是很有趣的，除非你没有仔细地筛选结果。

图7-2 在文档标题中单词Error很常见

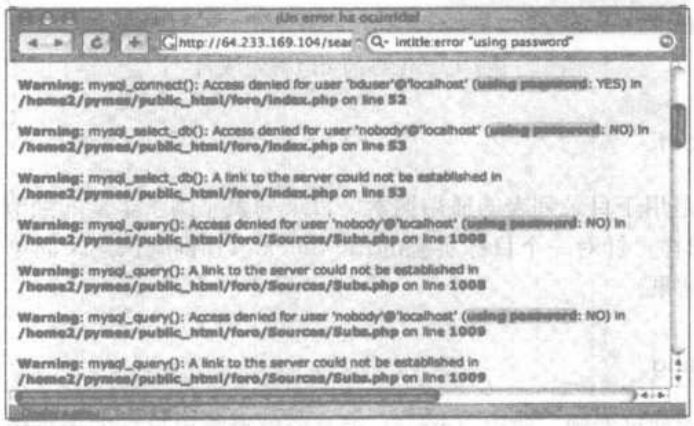


图7-3 错误隐藏，警告潜伏

7.1.4 login | logon

我们将在第8章中看到，登录（login）入口就是一个网站的“前门”。登录入口可以泄露目标的软件及操作系统信息，而且在大多数情况下，登录入口主页面中也有“自我帮助”文档的链接。这些文档是用来帮助那些在登录过程中出现问题的用户的。不管用户是忘记了他或她的密码，甚至是用户名，这些文档都能帮助攻击者，或者是我们这种安全测试员获取该网站的访问权限。

许多时候，登录入口页面上链接的文档会列出E-mail地址、电话号码或者是能够帮助遇到麻烦的用户重新获取访问权限的人工客服的URL。这些客服人员，或者服务台操作员，都是很好的社会工程攻击目标。即便是规模最小的安全测试小组都不能没有一个社会工程专家，他甚至能够说服一个爱斯基摩人脱掉用于保暖的衣服。大部分安全系统都有一个共同的弱点链接：操作键盘的人。单词login和logon广泛用于互联网，如图7-4所示，它们出现在上百万个页面中。

图7-4 login和logon搜索登录入口

同样常见的是页面文本中显示的词组login trouble。诸如此的短语链接给那些忘记了登录证书的任性的用户提供了帮助。当然，这个信息对于那些攻击者和渗透测试者也同样有价值。

7.1.5 username | userid | employee.ID | "your username is"

我们将在第9章中看到，有许多不同的方法用来从一个目标系统中获取用户名(username)。即使在大多数的认证机制中，用户名都是不太重要的一半，但是至少应对它进行最低限度的保护，以防止外人获取。图7-5表明，即便是那些很少泄露信息的站点在面对接二连三的大量刺探型Google查询时，也会返回许多有趣的结果。为了避免对这个例子中的目造成负面影响，我们对图中的某些细节作了编辑。

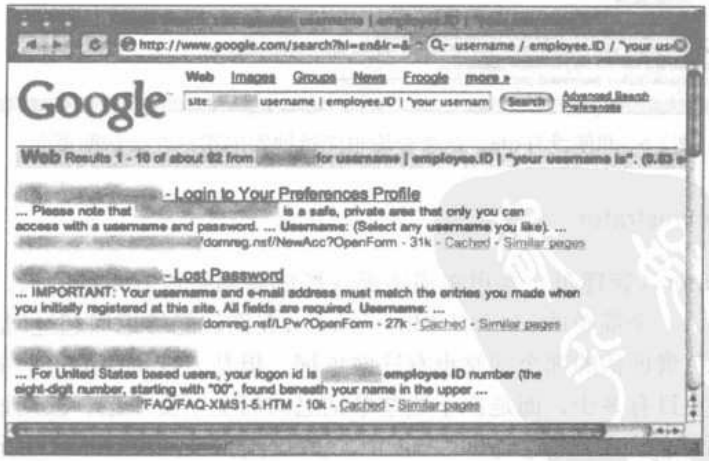


图7-5 即便"口风很紧的"站点也会提供登录入口

在一条结果中只出现单词username并不能说明存在一个漏洞，但是这个查询的结果却给攻击者提供了一个着手点。由于没有什么好的理由能够说明为什么要从你所保护的站点中删除单词username的出处，那么为什么不依靠这个常见的单词集合，以在评估中至少能找到一个立足点呢？

7.1.6 password | passcode | "your password is"

单词password（口令、密码）在互联网上太常见了，仅仅这一个单词的查询就有超过十亿条结果。仅仅单独查找这个单词是没有什么意义的，但是如果和site操作符组合起来查询，那就不一样了。

在一次评估中，很有可能这个和site操作符组合在一起的查询会包含给那些忘记了口令的用户提供帮助的页面。在某些情况下，这个查询也会搜索提供关于如何创建口令的规则信息的页面。这类信息可以用来对口令域实施智能猜解或者暴力破解攻击。

不管这个查询看起来是什么样的，这类查询很少返回真实的口令。网络中确实存在口令，但是这个查询不太适合搜索它们。（我们将在第9章中了解能够搜索口令的查询。）类似于登录入口和用户名查询，这个查询也能够提供进入系统的着手信息。大多数时候，该查询都需要与site操作符一起使用，但是只需要少许变化，该查询就可以在不与site一起使用时来做到这一点，如图7-6所示。诸如此类的"忘记口令"页面就非常有用。

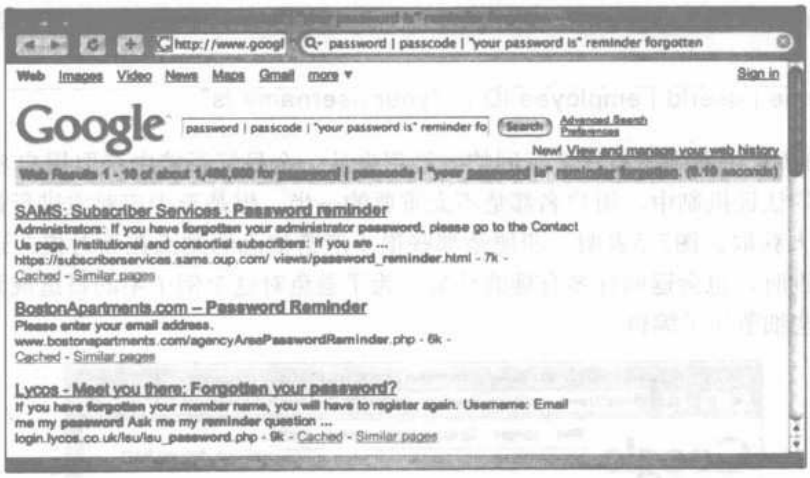


图7-6 即使没有site，这个查询也能够搜索出用户登录帮助页面

7.1.7 admin | administrator

单词administrator（管理员）常用来描述那些控制着网络或者系统的人。网络上许多地方都出现了这个单词，一个简单的admin | administrator查询就能够返回超过5亿条结果。这也说明了很有可能你所负责评估的那个网站也有这些单词。但是，查询中的这些或者其他单词的价值并不在于结果的数目有多少，而是在于这些单词之间的上下文联系。改动该查询，就算在没有site操作符的情况下添加"change your"也可以返回有趣的结果，如图7-7所示。

正如几个基本的词源那样，词组Contact your system administrator（联系系统管理员）在网络中也是十分常见的词组。像"please contact your * administrator"这样的查询所返回的结果引用了当地、公司、网站、部门、服务器、系统、网络、数据库、E-mail，甚至是网球管理员。如果一个网络用户被告知需要联系管理员，那么很有可能存在对安全测试人员而言比较重要的数据。

图7-7 调整过的Admin查询及结果

单词administrator也可以用来搜索管理登录页面，或者是登录入口。（我们会在第8章中更深入地了解登录入口检测。）查询“administrative login”会返回数百万条结果，其中许多条都是管理登录页面。安全测试人员就可以利用这类登录页面上看似不重要的线索来分析网站服务器。大多数登录入口都给攻击者提供了关于服务器所用的软件线索，而且更吸引了那些装备了该类软件漏洞利用的攻击者。如图7-8所示，绑定的admin查询的大部分结果都泄露了管理登录页面。

图7-8 admin login泄露管理登录页面

另外一种有趣的利用administrator词源的用法是使用inurl搜索在页面的URL中对它们进行查找。如果在URL中的主机名、目录名或者文件名中找到了单词admin，那么很有可能该URL具有某种管理功能，从安全性的角度来看，这是很有用的。

7.1.8 -ext:html -ext:htm -ext:shtml -ext:asp -ext:php

-ext:html -ext:htm -ext:shtml -ext:asp -ext:php查询使用了filetype操作符的同义词ext，它是一个可忽略的查询。它单独使用时不会返回任何结果，而且需要和site操作符同时使用才能正常工作。这个查询的意思是在一次查找我们所感兴趣的文件的查询中，排除那些最常见的互联网文件类型。

你将在整本书中看到，的确存在许多泄露了有趣的信息的HTML、PHP和ASP页面，但是这一章却是关于如何快速地查找到有用的信息，而且这也是这个查询所要做的事情。这个搜索返回的文档常常可能用于文档研磨[⊖]（document grinding），我们将在第10章中对此作更为详尽的描述。这个查询中的文件扩展名的选择十分仔细。首先，到www.filext.com（一个获得所有已知文件扩展名的最好的资源）上查询以获得一个常见文件扩展名的列表。把列表中的8000多种文件扩展名一一用filetype操作符转化为一个Google查询。例如，如果我们想要搜索PDF扩展名，就可以使用类似于filetype:PDF PDF这样的查询来得到网上的已知结果数目。这类对filext.com中的每个已知文件扩展名都执行的Google查询，会花费大量时间，同时还要考虑到Google使用条款协议。当结果收集完毕后，结果会按照点击数目从高到低进行排列。查询结果排行前30位的文件扩展名见表7-1。

表7-1 互联网上排名前30位的文件扩展名

扩展名	近似结果数目
HTML	4 960 000 000
HTM	1 730 000 000
PHP	1 050 000 000
ASP	831 000 000
CFM	481 000 000
ASPX	442 000 000
SHTML	310 000 000
PDF	260 000 000
JSP	240 000 000
CGI	83 000 000
DO	63 400 000
PL	54 500 000
XML	53 100 000
DOC	42 000 000
SWF	40 000 000
PHTML	38 800 000
PHP3	38 100 000
FCGI	30 300 000

⊖ 指文档的仔细分析。——译者注

（续）

扩展名	近似结果数目
TXT	30 100 000
STM	29 900 000
FILE	18 400 000
EXE	17 000 000
JHTML	16 300 000
XLS	16 100 000
PPT	13 000 000
DLL	12 900 000
PS	10 400 000
GZ	10 400 000
STORY	9 850 000
X	8 640 000

该表根据Google的结果给出了互联网中最常见的文件类型。结合忽略排行前10位最常见的文件类型的站点搜索可以让你快速直达那些潜在的有趣的文档。有些时候，这个查询需要作一些调整，尤其是当该网站使用一种很少见的服务器生成文件扩展名时。例如，考虑和site操作符同时使用的这种查询，如图7-9所示。（为了保护目标的真实身份，我们对图中的某些特定部分进行了编辑。）

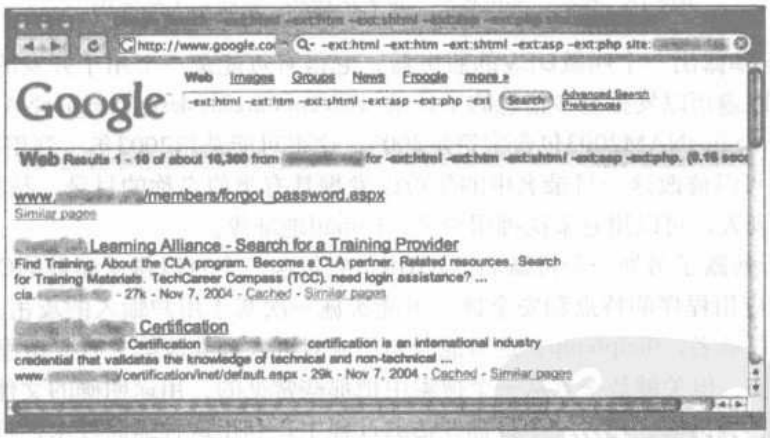


图7-9 一个带有site操作符的基本搜索

正如搜索结果中显示的那样，这个站点使用ASPX扩展名来显示某些网站内容。通过给查询添加-ext:aspx，然后再重新提交，就可以把这类内容从搜索结果中删除了。这个修改后的搜索暴露了一些有趣的信息，如图7-10所示。

通过给查询添加该网站所常用的文件扩展名进行查询，在翻过几页普通的页面之后，我们发现了一页充满了有趣的信息的页面。结果的第1行暴露出该站点支持HTTPS协议，这是HTTP的安全增强版本，用于保护敏感的信息。仅仅存在HTTPS协议就可以说明这个服务器含有值得保护的东西。结果的第1行还暴露出了几个嵌套的子目录（/research/files/summaries），我们可

以对其进行挖掘或者遍历以搜索其他信息。同样这一行还暴露出存在一个创建日期为2003年第一季度的PDF文档。

图7-10 新的、改进后的、更有价值的、更吸引人的查询

结果的第2行暴露出一个叫做DEV的服务器，它很有可能是一个用于开发的服务器。这个服务器也包含可以遍历以发掘更多信息的子目录（/events/archieves/strategiesNAM2003）。其中一个子目录名strategiesNAM2003包含字符串2003，这很可能是指2003年。利用第3章中介绍的递增置换技术，可以修改这一目录名中的年份以发掘具有类似名称的目录。结果的第2行也暴露出一个出席者列表，可以用它来挖掘用户名，E-mail地址等。

结果的第3行暴露了另外一个机器名，JOBS，它包含一个可以接受参数的ColdFusion应用程序。根据这个应用程序的特点和安全性，可能实施一次基于用户输入的攻击。结果的第4行暴露出一个新的目录名，/help/emp，这可能会遍历或者进入到其他第三方应用程序中。

结果还有很多，但关键是一旦从一个搜索中把那些常见的、用途明确的文件删除掉，有趣的信息就会突出显现出来。在对一个目标评估的过程中，利用这种缩简技术，可以节约攻击者或者安全技术专家大量的时间。

7.1.9 inurl:temp | inurl:tmp | inurl:backup | inurl:bak

把site操作符与inurl:temp | inurl:tmp | inurl:backup | inurl:bak组合在一起进行查询，可以用来搜索服务器上临时的或者备份文件或目录。虽然还可能有许多针对临时或者备份文件的命名约定，但是这个搜索关注的是最常见的关键字。由于这个搜索使用了inurl操作符，所以它也会搜索包含这些关键字作为文件扩展名的文件，例如index.html.bak。要把这个搜索改为关注文件扩展名是一种选择，但是如果在URL里找到这些关键字将更有趣。

7.1.10 intranet | help.desk

术语intranet，不管其技术含义有多么特殊，它已经成为一个普通的用来描述局限于一个规模较小的网络的术语。大多数情况下，intranet描述了一个封闭式的或者私有的网络，是不对外公开的。但是，许多站点配置了可以从互联网访问到内部网的入口，让这些应该封闭的网络与潜在的攻击者仅有一步之遥。

在极少数情况下，私有的内部网可以由于网络设备的不恰当配置导致能从公共的互联网中发现。在这些情况中，网络管理员完全没有意识到他们的私有网络已经可以被互联网中的任何一个人访问。一般来说，具有互联网连接的内部网只允许来自外部空间的部分访问。这时，通常会使用过滤器来限定只允许来自特定地址的访问者访问某些页面，这些地址可能来自某个设施或者校园内部。这种配置存在两个主要的问题。首先，监控特定页面的访问权限对于管理来说，是个很繁重的工作。其次，这并不是真正的访问控制。如果攻击者获得了一台本地代理服务器的访问权限，把请求绑定在一台本地配置不恰当的Web服务器上，或者简单地攻陷了同一个网络内作为可信内部网用户的机器时，这种限制可以很容易突破。遗憾的是，我们几乎不可能提供这种技术在实际应用中的一个例子。我们在这一节中给出的每个例子，攻击者都可以很容易地利用一些简单的Google查询来进行重构。

服务台的帮助作用过大以至于给它的名声带来了不好的影响。从服务台开始，黑客就已经开始伪装自己以试图从毫无戒心的技术专家那里获取敏感的信息。近一段时间，服务台已经采取了针对黑客威胁的防卫措施，即在帮助他们之前需要对请求帮助的人进行验证。大多数服务台员工会（或者应该会）询问身份信息，如用户名、社保号码、员工编号，甚至是PIN（个人身份）号码以合理地验证请求者的身份。有些措施还要更好，但是对于大多数情况而言，今天的服务台技术专家至少已经意识到了由冒名顶替者发起的潜在威胁。

在第4章中，我们曾讨论了用Google来获取服务台人员可能需要的身份信息的方法，但是intranet | help.desk这个查询并不是用来突破服务台防卫措施的，而是用来搜索描述服务台帮助策略页面的。当把这个查询和site操作符组合使用时，结果会给出服务台（网页、电话号码及其类似信息）的位置，服务台技术专家可能要求提供的信息（攻击者可以在请求之前进行收集），以及在大多数情况下描述了解决问题步骤的链接。自我帮助文档通常都是相当冗长的，而且狡猾的攻击者能够使用这些文档中的信息来剖析目标网络或者服务器。每个规则都有例外，但不同的是这个查询，当和site操作符组合在一起时，能够挖掘出关于目标的可用于后续攻击的信息。

7.2 总结

这个列表可能并不完美，但是你完全可以用这个列表中的10个搜索来创建你自己的“杀手”搜索列表。意识到能够对一个目标起作用的搜索不一定也能够对其他的目标起作用是相当重要的。记下可用的搜索，并且试着分别找出有效的和不起作用的搜索之间的共同之处。在第11章和第12章中讨论的自动化工具可以用来提供更长的Google查询列表，例如那些在Google Hacking数据库中找到的列表，但是在某些情况下，越简单越好。如果你在总结有效的查询之间的共性时遇到了困难，要毫不犹豫地把这些查询保存在一个列表中，以用于在后续章节中将

要讨论的自动化工具。

7.3 快速查找解决方案

site

- site操作符擅长从Google所收集的内容中提取与目标有关的信息。
- 这个操作符常用于和其他本章中提到的查询组合使用，以对某一目标实现精确搜索。

intitle:index.of

- 针对Apache风格的目录列表的通用搜索。
- 目录列表能够给攻击者提供有价值的信息。

error | warning

- 错误消息在每种环境下都会泄露出很多信息。
- 在某些情况下，警告文本能够提供目标所使用的代码内部的重要信息。

login | logon

- 这个查询可以高效地搜索出登录入口。
- 它也可以用来获取用户名以及解决问题的步骤。

username | userid | employee.ID | "your username is"

- 一种获取用户名的最通用的搜索。
- 当这个查询不能获取用户名时，其周围的上下文也能够泄露攻击者可以用于后续攻击行为的信息。

password | passcode | "your passcode is"

- 这个查询反映了单词password的常见用法。
- 这个查询能够泄露描述登录步骤、口令修改步骤以及目标所用的口令策略的文档。对于搜索有关电话会议的信息而言，特别是在Google日历搜索中，passcode特别有用。

admin | administrator

- 该查询用两个最常用的术语来表示网站的所有者或者维护者，可以用来暴露程序上的信息（例如“contact your administrator”），甚至是管理登录入口。

-ext:html -ext:htm -ext:shtml -ext:asp -ext:php

- 当把这个查询和site操作符组合使用时，能够剔除最常见的文件，而暴露出更有趣的文档。
- 该查询也可以针对不同的目标而作相应的修改以剔除其他的常见文件类型。

inurl:temp | inurl:tmp | inurl:backup | inurl:bak

- 这个查询可以搜索备份或者临时的文件及目录。

intranet | help.desk

- 这个查询用来搜索内部网站点（这些站点应该受到保护以防止外部人员访问）以及服务台联系信息和操作步骤。

7.4 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：如果选择使用自动化的工具，那么这10个微不足道的搜索又有什么巨大的价值呢？

答：自动化工具有其自身的价值，例如那些在第11章和第12章中讨论的。但是，规模较大的查询列表中的大多数搜索都是非常特殊的，它们只针对很少的一部分互联网站点。虽然这些特殊查询的效果极佳，但是一般最好还是保存一个规模较小的强大的搜索列表，以在评估中灵活地创造出丰富的搜索，尤其是当你用常规方法感到无计可施时更为有用。

问：用例如www.sans.org/top20?中的SANS Top 20列表中的搜索来作为基本的搜索列表不是比这一章中提到的那个列表更有意义吗？

答：SANS Top 20列表并没有什么问题，但是那个列表中的大部分关键项所描述的漏洞都不是基于Web的。这意味着，在大多数情况下，那个列表中描述的漏洞都无法通过基于Web的服务来检测或者利用，例如Google这样的基于Web的服务。

第8章 跟踪搜索Web服务器、登录入口和网络硬件

8.1 简介

通常认为，渗透测试人员也是专业的黑客，因为他们本质上是通过攻入客户的网络来查找、记录并且最终帮助客户解决系统或者网络中的安全缺陷的。不过，渗透测试人员和黑客在某些方面的区别还是很大的。

例如，大部分渗透测试人员都具有与他们将要测试的网络和系统相关的特殊指令。渗透测试人员的目标都是特定的，之所以如此的原因有很多（参见附录A以更深入地了解渗透测试方法学），但是在任何情况下，他们的目标都十分明确或者限定在某种形式。另一方面，黑客可以随意地从更为广泛的目标范围内选择目标。根据攻击者自身的动机和技术水平，他或她可以随意地选择针对某种已知漏洞利用的目标。这和渗透测试人员所用的模型恰好相反，而且正因为如此，也影响了我们探索Google Hacking这一主题的结构。我们在接下来的几章中讨论的技术都是黑客——“坏家伙”所常用的。

显然渗透测试人员也接触过我们将在这些章节中讨论的技术，但是在大多数情况下，在漏洞评估过程中时间都是尤为重要的，此时这些技术就很不方便。专业的安全人员通常以类似于流水作业的方式使用一些特殊的工具来执行这些任务，但是这些工具制造了许多的麻烦而且经常忽略了形式最为简单的信息泄露，但Google却能够做到这一点，所用的泄露方式是“雷达”几乎无法做到的。在日常的查找和挖掘互联网上的系统和网络的工作中，都会用到我们在这里讨论的技术，所以讨论如何用这些技术来更好地理解信息泄露的程度以及如何合理地减少这种信息泄露是相当重要的。

本章所讨论的技术常用于查找和分析与互联网连接的网络中的前端系统。我们将了解攻击者是如何利用Google查询所发现的似乎看起来没有任何意义的线索来分析Web服务器的。接着，我们将要看一看查找登录入口的方法，登录入口实际上就是指大多数网站的字面意义上的前门。我们会看到，一些登录入口给系统管理员提供了执行各种管理功能的访问点。大多数登录入口都给攻击者提供了服务器用的是何种软件的线索，同时也吸引了那些拥有相应软件的漏洞利用程序的攻击者。在本章的最后，我们将介绍一些用来查找各种网络设备的技术，这些网络设备有防火墙、路由器、网络打印机甚至是Web摄像机！

8.2 定位并剖析Web服务器

如果一个攻击者还没有定好目标，那么他可能会从Google搜索开始来查找符合他所能使用

的漏洞利用的特定目标。他可能尤其会关注操作系统、Web服务器软件的版本和名称、默认的配置、有漏洞的脚本，或者这些因素的任意组合。

有许多不同的查找服务器的方法。最常用的方法是使用简单的端口扫描技术。使用Nmap这样的工具，只需要对一个C类网络进行一个简单的端口80的扫描就可以暴露出多个潜在的Web服务器。一些集成工具，例如Nessus、H.E.A.T.或者Retina也可以运行某种类型的端口扫描，然后再执行一系列的安全性测试。这些功能也可以用Google查询来代替，虽然在大多数情况下，其结果远不及精心设计的漏洞扫描工具或者Web评估工具得到的结果有效。但是，要知道Google查询更为隐蔽而且也给攻击者和目标之前提供了某种程度的隔离。同时，还要知道黑客能够使用Google Hacking技术查找到你所负责保护的系统。最起码也应该明白Google黑客的能力以及意识到Google在攻击者的操作中扮演什么样的角色。

8.2.1 目录列表

在第3章中，我们讨论了目录列表，但更为重要的是目录列表对于剖析方法的重要性。目录列表底部的server标记能够明显地提供出关于正在运行的Web服务器软件的细节信息。如果某个攻击者有一个针对运行于UNIX服务器上的Apache 2.0.52的漏洞利用程序，那么他能够用一个例如server at “Apache/2.0.52”这样的查询来查找具有Apache 2.0.52 server标记的目录列表的服务器，如图8-1所示。

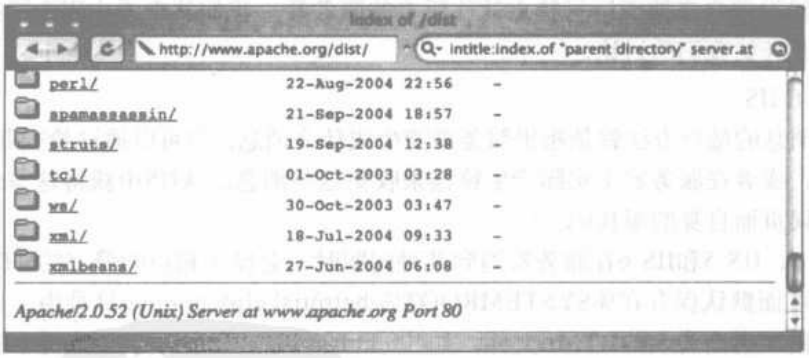


图8-1 标准的Server标记可以用来查找服务器

提示

要时常检查搜索到的结果的真实的页面（相对于搜索引擎所保存的缓存页面而言），因为服务器的版本序号可能会在搜索引擎的捕捉间隙里发生改变。

并不是所有的Web服务器都会在目录列表的底部放置这一标记，但是在默认情况下，大多数Apache衍生的软件都启用了这一特性。其他的平台，例如Microsoft的IIS（Internet Information Server）也会显示server标记，例如，图8-2所示的“Microsoft-IIS/5.0 server at”查询。

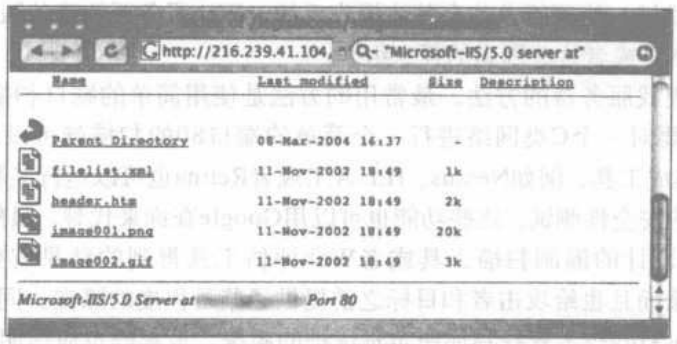


图8-2 查找IIS 5.0服务器

在搜索这些目录标记时，请始终记住，查询语法非常重要。查询“Microsoft-IIS/6.0” “server at”会返回许多不相关的结果，而查询“Microsoft-IIS/6.0 server at”就只返回相当接近的结果。由于我们在前面章节已经讨论过了目录列表，所以这里就不再赘述。你可以回过头来参考第3章来复习关于目录列表的知识。

8.2.2 Web服务器软件的错误消息

错误消息包含了大量有用的信息，但是在查找特定的服务器这一环境中，我们可以使用部分不同的几种错误消息来搜索运行特定软件版本的服务器。我们从查看由Web服务器软件自身产生的错误消息开始我们的讨论。

1. Microsoft IIS

查找错误消息的最好方法就是指出服务器能生成什么消息。你可以通过检查服务器源代码，或者配置文件，或者在服务器上实际产生错误来收集这些消息。从IIS中获得这些信息的最好的办法是检查错误页面自身的源代码。

默认情况下，IIS 5和IIS 6在服务器遇到某种问题时，会显示相应的静态的HTTP/1.1错误消息。这些错误页面默认保存在%SYSTEMROOT%\help\iishelp\common目录中。这些文件实质上是以相应的错误类型为名的HTML文件，如400.htm、401-1.htm、501.htm等。通过分析这些文件，我们能够得到这些页面之间的趋向及共性，而这对有效的Google搜索而言是必需的。例如，给出400错误的400.htm页面的第12行如下：

```
<title>The page cannot be found</title>
```

这就等于告诉我们可以用一个有效的intitle查询来搜索这一页面，例如intitle:“The page cannot be found”。但遗憾的是，这个搜索产生的结果（你也可以猜到）太多了。我们还需要再深入地挖掘400.htm文件以获取更多的线索。400.htm的第65～88行如下：

```
65. <p>Please try the following:</p>
66. <ul>
67. <li>If you typed the page address in the Address bar, make sure that it is
spelled correctly.</li>
68.
```

```
69. <li>Open the
70.
71. <script language="JavaScript">
72. <!--
73. if (!((window.navigator.userAgent.indexOf("MSIE") > 0) &&
(window.navigator.appVersion.charAt(0) == "2")))
74. {
75. Homepage();
76. }
77. -->
78. </script>
79.
80. home page, and then look for links to the information you want.</li>
81.
82. <li>Click the
83. <a href="javascript:history.back(1)">
84. Back</a> button to try another link.</li>
85. </ul>
86.
87. <h2 style="COLOR:000000; FONT: 8pt/11pt verdana">HTTP 400 - Bad Request<br>
88. Internet Information Services</h2>
```

第65行的“Please try the following”存在于这个目录中的每个错误文件中，这使得它可以作为一个较好的基础搜索的完美候选部分。这一行也可以有效地简化为“please * * following”。第88行也给出了另外一个出现在每个错误文档中的词组：“Internet Information Services”。这些都是用来搜索Google已经抓取的IIS HTTP/1.1错误页面的“黄金关键字”。例如查询intitle:“The page cannot be found”“please * * following”“Internet * Services”可以用来搜索显示了400错误页面的IIS服务器，如图8-3所示。

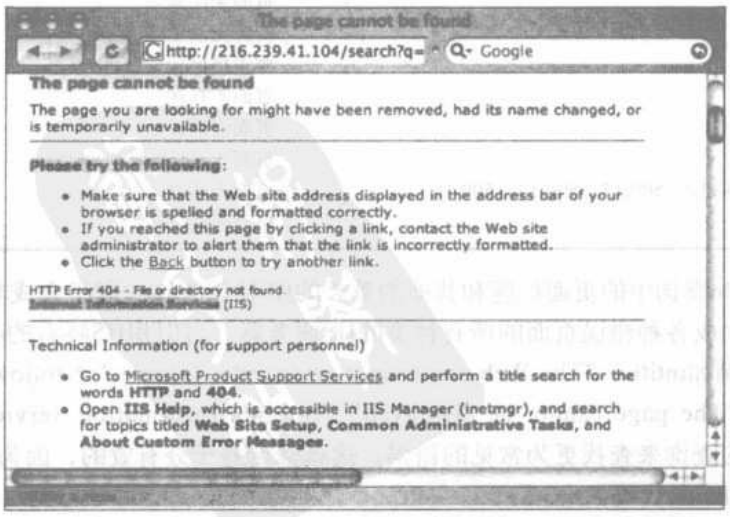


图8-3 灵活搜索IIS服务器

仔细观察这一缓存页面，你会注意到实际的错误代码自身也出现在这一页面上，大约在页面的中下部。这一错误行也出现在了其他每个IIS错误页面中，这又给我们的搜索提供了另外一种不错的限定方法。页面中的这一行是以“HTTP Error 404”开头，这似乎不是我们所关注的，因为考虑到我们是在搜索400错误页面，而不是404错误代码。之所以页面中给出错误代码，是因为有些IIS错误页面会给出相似的页面。虽然共性通常对Google搜索是有好处的，但是它们也会导致一些干扰，而且如果是在搜索某个特定的、比较复杂的错误页面时，会产生许多无效的结果。显然我们需要分清楚这些错误文件中包含的东西究竟是什么。表8-1列出了所有的从默认的IIS 5安装中提取出来的HTML错误页面的标题和错误代码。

表8-1 IIS HTTP/1.1错误页面标题

错误代码	页面标题
400	找不到该页面
401.1、401.2、401.3、401.4、401.5	无权查看该页面
403.1、403.2	页面无法显示
403.3	页面无法保存
403.4	页面必须通过安全渠道查看
403.5	页面必须使用高安全性能的Web浏览器查看
403.6	无权查看该页面
403.7	页面要求一个客户认证
403.8	无权查看该页面
403.9	页面无法显示
403.10、403.11	无权查看该页面
403.12、403.13	页面要求一个合法的客户认证
403.15	页面无法显示
403.16、403.17	页面要求一个合法的客户认证
404.1、404.b	网站无法找到
405	页面无法显示
406	资源无法显示
407	要求提供代理认证
410	页面不存在
412	页面无法显示
414	页面无法显示
500、500.11、500.12、500.13、500.14、500.15	页面无法显示
502	页面无法显示

这些用于intitle查询中的页面标题和其他的重要的IIS错误查询一起组合成非常高效的搜索，它们能够查找到生成各种错误页面的所有种类的IIS服务器。可以用IIS特有的404.1错误页面来搜寻IIS服务器，例如intitle:“The Web site cannot be found”“please * * following”。同时，也可以用像intitle:“The page cannot be displayed”“Internet Information Services”“please * * following”这样的查询来查找更为常见的错误，这类查询是十分有效的，因为搜索到的页面会显示出各种不同的错误代码。

除了显示默认的静态的HTTP/1.1错误页面之外，IIS也能够显示自定义的错误消息。这可

以通过管理控制台来配置。图8-4给出了一个这类自定义错误页面的例子。这种功能让Google黑客的工作变得更为困难了，因为没有一种明显的方法来查找自定义的错误页面。但是，一些错误消息，包括400、403.9、411、414、500、500.11、500.14、500.15、501、503以及505页面，都是不能自定义的。从Google hacking这方面来说，这也就意味着IIS 6服务器不能很容易地就阻止显示我们之前可以有效地搜索到的静态的HTTP/1.1错误页面。这样就打开了利用Google来搜索这些服务器的大门，即便服务器已被配置为显示自定义的错误页面。

除了利用搜索精确的词组来查找IIS的错误页面，我们还能够执行更为通用的查询，例如intitle:“the page cannot be found” inetmgr，这个查询所关注的是相当独特的用于描述IIS管理控制台的术语——inetmgr，如图8-3的底部所示。可以执行同样搜索的其他方法有intitle:“the page cannot be found”“internet information services”或者intitle:“Under construction”“Internet Inforamtion Services”。

还有其他一些更为特别的查询，它们能够揭露出IIS服务器的确切的版本号，如图8-4所示的查询intext:“404 Object Not Found” Microsoft-IIS/5.0。

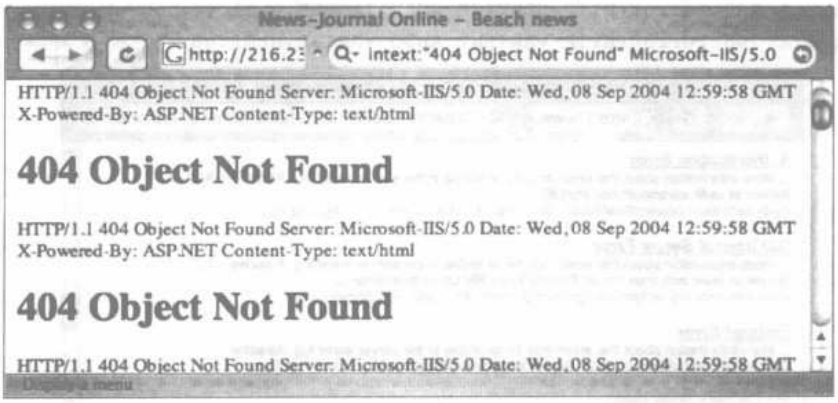


图8-4 用于查找IIS 5.0的错误消息“Object Not Found”

2. Apache Web服务器

也可以利用服务器生成的错误消息来查找Apache Web服务器。一些常见的搜索，例如“Apache/1.3.27 Server at”“-intitle:index.of intitle:inf”或者“Apache/1.3.27 Server at”“-intitle:index.of intitle:error”（如图8-5所示），都可以用来查找由一条信息或者错误消息而暴露服务器版本的Apache Web服务器。

例如“Apache/2.0.40” intitle:“Object not found!”这样的查询就可以搜索出现了这种错误消息的Apache 2.0.40 Web服务器。图8-6就是一个来自运行于Red Hat 9.0之上的Apache 2.0.40服务器的错误页面。

虽然利用通用的并且良好的基本搜索来进行查询也没有什么不好的，但是我们已经在上一节中看到，查阅服务器软件本身来发现搜索线索更为高效。大部分Apache的安装都依赖于一个称作httpd.conf的配置文件。遍历Apache 2.0.40的httpd.conf文件就能够找到错误消息的HTML模板位于何处。这些引用的文件（如下所示）位于网站的根目录，例如/error/http_BAD_REQUEST.html.var，

在文件系统中相应的目录为/var/www/error。

```
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
```

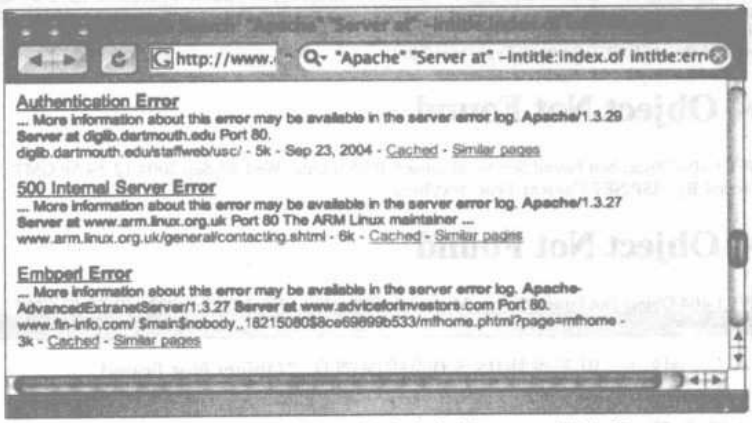


图8-5 用普通的错误搜索来查找Apache服务器

图8-6 一条产生自Apache 2.0.40的常见错误消息

查看这些模板文件中的任意一个，我们都能够看到熟悉的HTML代码以及展示了一个错误页面的结构的变量列表。每个文件自身都会根据语言分成几节。HTTP_NOT_FOUND.html.var文件的英语部分如下：

```
Content-language: en
Content-type: text/html
Body:----------en--
<!—#set var="TITLE" value="Object not found!" —>
<!—#include virtual="include/top.html" —>

   The requested URL was not found on this server.

   <!--#if expr="$HTTP_REFERER" -->

   The link on the
   <a href="<!--#echo encoding="url" var="HTTP_REFERER"-->">referring
   page</a> seems to be wrong or outdated. Please inform the author of
   <a href="<!--#echo encoding="url" var="HTTP_REFERER"-->">that page</a>
   about the error.

<!--#else -->

   If you entered the URL manually please check your
   spelling and try again.

   <!--#endif -->

<!—#include virtual="include/bottom.html"—>
----------en--
```

我们可以注意到错误页面的每一节都标记得很清晰，这给转化为Google查询带来了方便。靠近列表顶部的TITLE变量表明文本“Object not found!”会显示在浏览器的标题栏上。当处理完这个文件并把它显示在一个Web浏览器中时，它的样子如图8-2所示。但是，Google Hacking并不总是这么容易的。intitle:“Object not found!”这一搜索太普通了，其返回的结果如图8-7所示。

这些结果并不是我们要找的。为了让我们的结果更为精确，我们需要一个更好的基本搜索。根据Apache 2.0源代码中包含的模板文件来构造我们的基本搜索不仅能让我们搜索到所有服务器能产生的潜在的错误消息，还能够让我们了解那些消息是如何翻译为其他语言的，这样就形成了非常牢靠的多语言基本搜索。

前面列出的HTTP_NOT_FOUND.html.var文件引用了两行虚拟包含（virtual include），一行靠近源代码的前面（include/top.html），另一行在源代码的后面（include/bottom.html）。这两行将命令Apache读取两个文件（在本例中位于/var/www/error/include目录中）的内容并且把这两个文件的内容插入到当前文件中。如下的代码列出了bottom.html文件的内容，并且还点明了一些微妙之处以帮助构造完美的基本搜索：

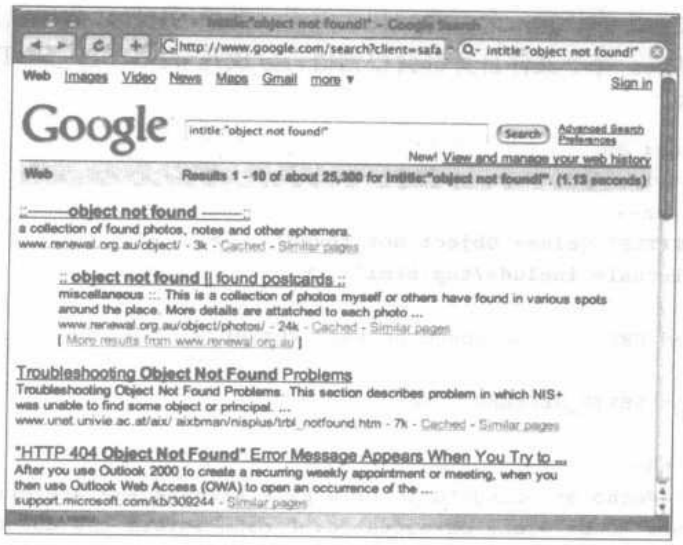


图8-7 错误消息文本并不足以用来剖析服务器

```
</dd></dl><dl><dd>
<!--#include virtual="../contact.html.var" -->
</dd></dl>
<h2>Error <!--#echo encoding="none" var="REDIRECT_STATUS" --></h2>
<dl>
<dd>
<address>
<a href="/"><!--#echo encoding="url" var="SERVER_NAME" --></a>
<br />
<!--#config timefmt="%c" -->
<small><!--#echo encoding="none" var="DATE_LOCAL" --></small>
<br />
<small><!--#echo encoding="none" var="SERVER_SOFTWARE" --></small>
</address>
</dd>
</dl>
</body>
</html>
```

首先，注意第4行，这一行将在页面中显示单词“Error”。尽管这可能看起来非常普通，但它却是让结果与图8-7所显示的完全不同的重要的细微之处。第2行表明这个文件还读取并包含了另外一个文件（/var/www/error/contact.html.var）。这个文件包含了更多我们可以融合到基本搜索中的细节信息，文件内容如下：

```
1. Content-language: en
2. Content-type: text/html
3. Body:----------en--
4. If you think this is a server error, please contact
```

单地用浏览器连接到他自己的Web服务器以检查Web软件是否安装正确。一些操作系统甚至带有已经安装好了的Web服务器软件。在这种情况下，机器的所有者甚至可能都没有意识到在他的机器上正运行着一个Web服务器。机器所有者的这类不小心的行为会导致攻击者理所当然地假设这个Web软件没有很好地进行维护，而且可以扩展为不安全的。进一步扩展，攻击者还可以假设服务器所在的整个操作系统由于维护不好而具有某些漏洞。

在某些情况下，Google会在Web服务器的早期安装阶段，即服务器仍显示一组默认的页面时，抓取服务器页面。在这些情况下，通常在Google抓取站点和在服务器上放置实际的内容之间会有一个短暂的时间间隔。这意味着当前显示的页面和Google缓存显示的页面是不一致的。不过，这对于Google黑客而言是没有什么区别的，因为即便是过去存在的默认页面都足以满足剖析的目的。要知道的是，我们所提交的查询实际上搜索的是页面的Google缓存版本。不管服务器安装默认页面的原因是什么，最终都会有一个攻击者对Google搜索找到的显示默认页面的机器感兴趣。

默认页面的一个典型的例子是Apache Web服务器的默认页面，如图8-13所示。

图8-13 一个典型的Apache默认Web页面

注意，其中管理员的E-mail还是通用的，从这一点可以看出在该服务器安装时，管理员没有在细节上花费太多的精力。这些默认的页面都没有列出服务器的版本号，而这正是成功的攻击所必需的一块消息。但是，攻击者还是可以查找这些默认页面之间的特殊变化以确定服务器版本的特定范围。如图8-13和图8-14所示，运行Apache 1.3.11到1.3.26版本的默认页面有些细微的差别。

为了有效地利用这些差别，我们可以使用特定的Google查询来查找带有这些默认页面的服务器，这些默认页面表明了服务器很有可能运行某一特定版本的Apache服务器。表8-4给出了可以用来查找运行默认页面的特定Apache服务器组的查询。

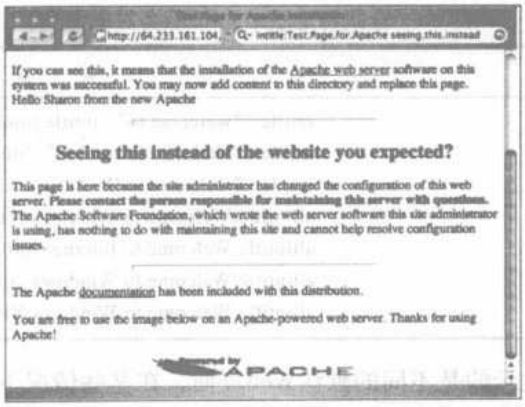


图8-14 Apache默认页面的细微差别

表8-4 搜索默认的Apache安装的查询

Apache服务器版本	查 询
Apache 1.2.6	intitle: “Test Page for Apache Installation” “You are free”
Apache 1.3.0 - 1.3.9	intitle: “Test Page for Apache” “It worked!” “this Web site!”
Apache 1.3.11 - 1.3.31	intitle:Test.Page.for.Apache seeing.this.instead
Apache 2.0	intitle:Simple.page.for.Apache Apache.Hook.Functions
Apache SSL/TLS	intitle:test.page “Hey, it worked !” “SSL/TLS-aware”
安装在Red Hat上的Apache	“Test Page for the Apache Web Server on Red Hat Linux”
安装在Fedora上的Apache	intitle: “test page for the apache http server on fedora core”
安装在Debian上的Apache	intitle: “Welcome to Your New Home Page!” debian
安装在其他Linux上的Apache	intitle: “Test Page * * Apache Web Server on ” - red.hat -fedora

IIS在第一次安装的时候也会显示一个默认的网页。像intitle: “Welcome to IIS 4.0”这样的查询能够搜索到非常特殊版本的IIS，如图8-15所示。关于查找特定版本的ISS服务器的查询语句见表8-5。

图8-15 搜索Windows NT 4.0/OP上的默认的IIS 4.0安装

表8-5 查找特定版本的IIS服务器的查询

IIS服务器版本	查 询
多个	intitle: "welcome to" intitle:internet IIS
未知	intitle: "Under construction" "does not currently have"
IIS 4.0	intitle: "welcome to IIS 4.0"
IIS 4.0	allintitle:Welcome to Windows NT 4.0 Option Pack
IIS 4.0	allintitle:Welcome to Internet Information Server
IIS 5.0	allintitle:Welcome to Windows 2000 Internet Services
IIS 6.0	allintitle:Welcome to Windows XP Server Internet Services

每个不同版本的IIS显示的是不同的默认Web页面，在某些情况下，一些补丁可能会修改默认页面的内容。在这些情况下，如果把这些细微的页面变化和搜索结合起来，则不仅能够找到操作系统的版本及Web服务器的版本，还能找出其服务包补丁级别和安全补丁级别。这些信息对于那些喜欢不只是攻击Web服务器，而且还攻击位于Web服务器背后的操作系统的攻击者而言，是相当重要的。一般情况下，一个控制了操作系统攻击者比仅控制了Web服务器的攻击者能做出更大的破坏。

也可以利用一些简单的查询来搜索到Netscape服务器，例如allintitle:Netscape Enterprise Server Home Page，如图8-16所示。

图8-16 搜索Netscape Web服务器

其他的Netscape服务器也可以用简单的allintitle搜索查找到，这些搜索见表8-6所示。

表8-6 搜索Netscape服务器的查询

Netscape服务器类型	查 询
Enterprise Server	allintitle:Netscape Enterprise Server Home Page
FastTrack Server	allintitle:Netscape FastTrack Server Home Page

许多其他不同类型的Web服务器也可以通过默认页面来搜索到。表8-7给出了可以用这种技术来剖析的更多特别的Web服务器列表。

表8-7 搜索更多特别的服务器的查询

服务器/版本	查 询
Cisco Micro Webserver 200	"micro webserver home page"
Generic Appliance	"default web page" congratulations "hosting appliance"
HP appliance sa1*	intitle: "default domain page" "congratulations" "hp web"
iPlanet/Many	intitle: "web server, enterprise edition"
Intel Netstructure	"congratulations on choosing" intel netstructure
JWS/1.0.3-2.0	allintitle:default home page java web server
J2EE/Many	intitle: "default j2ee home page"
Jigsaw/2.2.3	intitle: "jigsaw overview" "this is your"
Jigsaw/Many	intitle: "jigsaw overview"
KFSensor honeypot	"KF Web Server Home Page"
Kwiki	"Congratulations! You've created a new Kwiki website."
Matrix Appliance	"Welcome to your domain web page" matrix
NetWare 6	intitle: "welcome to netware 6"
Resin/Many	allintitle:Resin Default Home Page
Resin/Enterprise	allintitle:Resin-Enterprise Default Home Page
Sambar Server	intitle: "sambar server" "1997..2004 Sambar"
Sun AnswerBook Server	inurl: "Answerbook2options"
TivoConnect Server	inurl:/TiVoConnect

8.2.5 默认文档

Web服务器软件通常会附带一些手册和文档，它们位于Web目录中。攻击者可以利用这种文档对Web软件进行剖析或者搜索。例如，Apache Web服务器所带的文档是HTML格式的，如图8-17所示。

图8-17 用于剖析服务器的Apache文档

在大多数情况下，默认文档和错误消息或者默认页面一样都不能精确地描述服务器的版本，但是这种信息显然可以用于搜索目标以及用于了解服务器的潜在安全隐患。如果服务器管理员忘记删除默认文档，攻击者就有理由相信服务器的其他细节，如安全性，同样被管理员所忽略。其他的Web服务器，例如IIS，同样带有默认文档，如图8-18所示。

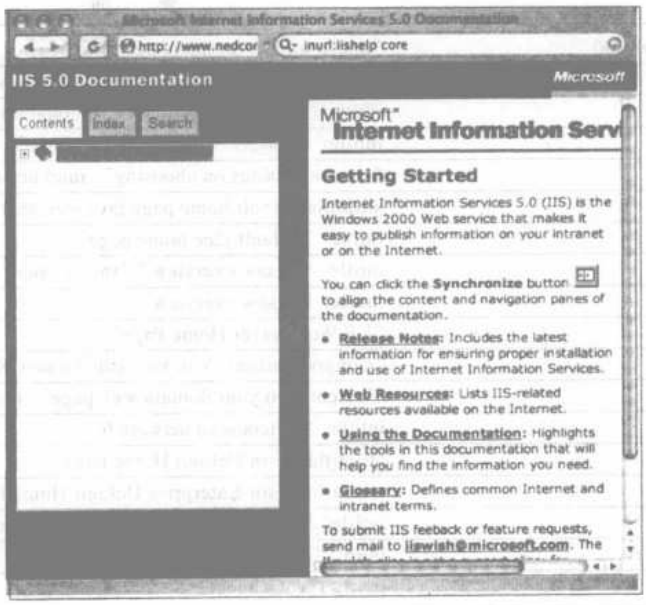


图8-18 通过默认手册来剖析IIS服务器

一般来说，专用的程序，例如CGI扫描器或者Web应用程序评估工具更适合于查找这类默认页面和程序，但是如果Google抓取了这些页面的话（例如，默认主页上的一个链接），你也可以用Google查询来查找这些页面。表8-8列出了一些可以用于查找默认文档的查询。

表8-8 查找默认文档的查询

搜索主题	查 询
Apache 1.3	intitle: "Apache 1.3 documentation"
Apache 2.0	intitle: "Apache 2.0 documentation"
Apache Various	intitle: "Apache HTTP Server" intitle: "documentation" \
ColdFusion	inurl:cfdocs
EAServer	intitle: "Easerver" "Easerver Version * Documents"
iPlanet Server 4.1/Enterprise Server 4.0	inurl: "/manual/servlets/" intitle: "programmer"
IIS/Various	inurl:iishelp core
Lotus Domino 6	intext:/help/help6_client.nsf
Novell Groupwise 6	inurl:/com/novell/gwmonitor
Novell Groupwise WebAccess	inurl: "/com/novell/webaccess"
Novell Groupwise WebPublisher	inurl: "/com/novell/webpublisher"

8.2.6 示例程序

Web软件除了附带文档和手册之外，一个软件包通常还包含一些默认的应用程序。这些默认的应用程序，例如默认的Web页面，可以帮助演示软件的功能，作为开发者的起点，提供可以用作学习工具的示例函数和代码。遗憾的是，这些示例程序不仅可以用来剖析Web服务器，而且通常这些示例程序包含了一些攻击者可以用来攻陷服务器的缺陷或者功能。如图8-19所示的Microsoft Index Server的内容查询页面可以允许Web访问者搜索整个网站的内容。在某些情况下，这个查询页面能够查找到没有被其他任何页面链接的页面或者包含敏感信息的页面。

图8-19 Microsoft Index Server内容查询页面

类似于默认页面，专用的设计用来抓取网站以用于搜索这些示例程序的工具更适合于查找这类页面。但是，如果Web服务器提供了一个包含演示页面和程序链接的默认页面，那么Google就能找到这些演示页面和程序。在某些情况下，即使网站的主页已经更新了而且链接也被删除，这些页面的缓存版本仍然存在。要记得，你可以使用这些缓存页面与&strip=1选项一道来匿名查看页面。这可以让信息收集工作远离服务器管理员的监视。表8-9给出了一些可以用来查找默认安装程序的查询。

表8-9 搜索默认程序的查询

软　件	查　询
Apache Cocoon	inurl:cocoon/samples/welcome
Generic	inurl:demo | inurl:demos
Generic	inurl:sample | inurl:samples
IBM Websphere	inurl:WebSphereSamples
Lotus Domino 4.6	inurl: /sample/framew46

（续）

软　件	查　询
Lotus Domino 4.6	inurl:/sample/faqw46
Lotus Domino 4.6	inurl:/sample/pagesw46
Lotus Domino 4.6	inurl:/sample/siregw46
Lotus Domino 4.6	inurl:/sample/faqw46
Lotus Domino 4.6	inurl:/sample/faqw46
Lotus Domino 4.6	inurl:/sample/faqw46
Lotus Domino 4.6	inurl:/sample/faqw46
Microsoft Index Server	inurl:samples/Search/queryhit
Microsoft Site Server	inurl:siteserver/docs
Novell NetWare 5	inurl:/lcgi/sewse.nlm
Novell GroupWise WebPublisher	inurl:/servlet/webpub groupwise
Netware WebSphere	inurl:/servlet/SessionServlet
OpenVMS!	inurl:sys$common
Oracle Demos	inurl:/demo/sql/index.jsp
Oracle JSP Demos	inurl:demo/basic/info
Oracle JSP Scripts	inurl:ojspdemos
Oracle 9i	inurl:/pls/simpledad/admin_
IIS/Various	inurl:iissamples
IIS/Various	inurl:/scripts/samples/search
Sambar Server	intitle: "Sambar Server Samples"

8.3 定位登录入口

术语“登录入口”描述了一个作为网站“前门”的页面。登录入口用于允许用户在登录之后访问某些特定的功能。而Google黑客搜索登录入口是用来剖析目标所用的软件以及查找提供了对攻击有用的信息的链接和文档。另外，如果攻击者拥有一个针对某个特定软件的漏洞利用程序，同时这个软件也提供了登录入口，那么攻击者就可能使用Google查询来查找潜在的目标。

一些登录入口，类似如图8-20所示的用allinurl:" exchange/logon.asp"捕捉到的登录入口，显然是由软件制造商提供的默认页面，在本例中是Microsoft。就像攻击者只要简单地搜索默认页面就能够了解目标的潜在安全性一样，默认的登录入口不但也表明了服务器管理员的技术水平通常很低，同时也暴露出网站的安全性同样很弱。更糟的是，如图8-20所示的默认登录入口还给出了程序的软件修订版本，在本例中为version 5.5 SP4。攻击者可以利用这种信息来搜索该软件版本的已知漏洞。

通过浏览登录入口页面的链接，攻击者通常还能了解到服务器的其他信息。在这类信息泄露方面，Outlook Web访问入口特别有名，因为它提供了不用登录到邮件系统就可以查看的匿名的公共访问区域。如图8-21所示，这种公共访问区域有时提供了访问公共目录的权限或者提供了查看可以用来收集用户名或信息的群发邮件的权限。

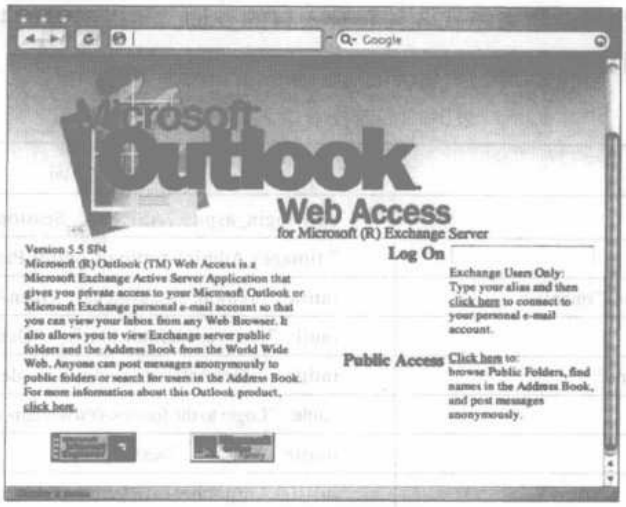


图8-20 Outlook默认的Web访问入口

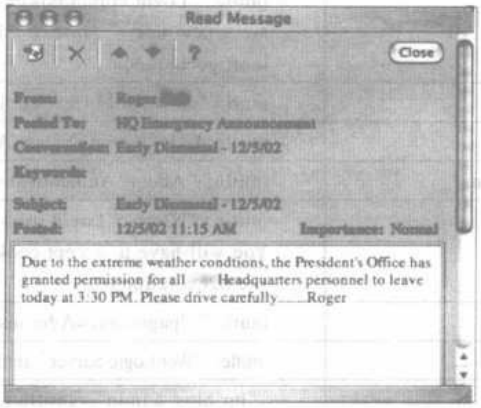


图8-21 能够通过登录入口找到公共访问区域

有些登录入口能提供更为详细的信息。如图8-22所示，Novell管理入口提供了大量关于服务器的信息，包括服务器软件的版本、应用软件的版本、软件更新的日期以及服务器运行时间。这类信息对于一个正在对该服务器进行攻击的攻击者是十分有用的。

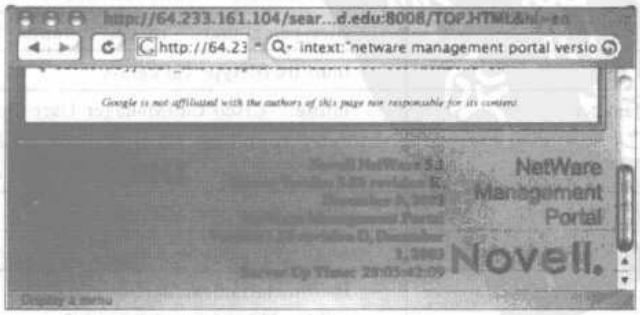


图8-22 Novell管理入口泄露了大量的信息

表8-10给出了一些可以用来查找各种登录入口的查询。可以参考第4章了解更多关于登录入口的信息及它们所泄露的信息。

表8-10 查找登录入口的查询

登录入口	查 询
.NET login pages	ASP.login_aspx "ASP.NET_SessionId"
4images Gallery	"4images Administration Control Panel"
Aanval Intrusion Detection Console	intitle: "remote assessment" OpenAanval Console
ActiveX Login	inurl: "Activex/default.htm" "Demo"
Affiliate Tracking Software	intitle: "iDevAffiliate - admin" -demo
Aimoo	intitle: "Login to the forums-@www.aimoo.com" inurl:login.cfm?id=
AlternC Desktop	intitle: "AlternC Desktop"
Ampache	intitle:Ampache intitle: "love of music" password|login| "Remember Me." -welcome
Anyboard Login Portals	intitle: "Login Forum Powered By AnyBoard" intitle: "If you are a new user:" intext: "Forum Powered By AnyBoard" inurl:gochat -edu
aspWebCalendar	inurl: "calendar.asp?action=login"
Asterisk Recording Interface	intitle:ARI "Phone System Administrator"
Athens Access Management system	intitle: "Athens Authentication Point"
b2evolution	intitle: "b2evo > Login form" "Login form. You must log in! You will have to accept cookies in order to log in " -demo -site:b2evolution.net
Bariatric Advantage	inurl: "/?pagename=AdministratorLogin"
BEA WebLogic Server 8.1	intitle: "WebLogic Server" intitle: "Console Login" inurl:console
betaparticle	"bp blog admin" intitle:login | intitle:admin -site:johnny.ihackstuff.com
bitboard2	intext: ""BiTBOARD v2.0" BiTSHiFTERS Bulletin Board"
Blogware Login Portal	intitle: "Admin Login" "admin login" "blogware"
Cacti	intitle: "Login to Cacti"
Cash Crusader	"site info for" "Enter Admin Password"
CGIIRC	filetype:cgi inurl: "irc.cgi" | intitle "CGI:IRC Login"
CGIIRC	inurl:irc filetype:cgi cgi:irc
Cisco CallManager CallManager	intitle: "Cisco CallManager User Options Log On" "Please enter your User ID and Password in the spaces provided below and click the Log On button to co"
Cisco VPN 3000 concentrators	intitle: "inc. vpn 3000 concentrator"
Cisco WebVPN Services Module	inurl:webvpn.html "login" "Please enter your"
Citrix Metaframe	inurl:metaframexp/default/login.asp | intitle: "Metaframe XP Login"

（续）

登录入口	查　询
Citrix Metaframe	inurl:/Citrix/Nfuse17/
CMS/Blogger	inurl:textpattern/index.php
ColdFusion	intitle: "ColdFusion Administrator Login"
ColdFusion	inurl:login.cfm
Communigate Pro	intitle:communigate pro entrance
Confixx	inurl:confixx inurl:login|anmeldung
Coranto	inurl:coranto.cgi intitle:Login (Authorized Users Only)
CPanel	inurl::2082/frontend -demo
Create Pro.	inurl:csCreatePro.cgi
CUPS	inurl: "631/admin" (inurl: "op=*") | (intitle:CUPS)
CuteNews	"powered by CuteNews" "2003..2005 CutePHP"
Cyclades TS1000 and TS2000 Web Management Service	allintitle: "Welcome to the Cyclades"
Dell OpenManage	inurl: "usysinfo?login=true"
Dell Remote Access Controller	intitle: "Dell Remote Access Controller"
Docutek Eres	intitle: "Docutek ERes - Admin Login" -edu
DWMail	"Powered by DWMail" password intitle:dwmail
Easy File Sharing Web Server	intitle: "Login - powered by Easy File Sharing Web"
EasyAccess Web	inurl:ids5web
EasySite	"You have requested access to a restricted area of our website. Please authenticate yourself to continue."
Ecommerce	inurl: "vsadmin/login" | inurl: "vsadmin/admin" inurl:.php|.asp - "Response.Buffer = True" - javascript
eHealth	inurl:bin.welcome.sh|inurl:bin.welcome.bat | intitle:eHealth.5.0
Emergisoft	"Emergisoft web applications are a part of our"
eMule	intitle: "eMule *" intitle: "- Web Control Panel" intext: "Web Control Panel" "Enter your password here."
Ensim WEBppliance Pro.	intitle: "Welcome Site/User Administrator" "Please select the language" -demos
Enterprise Manager 10g Grid Control	inurl:1810 "Oracle Enterprise Manager"
ePowerSwitch D4 Guard	intitle: "ePowerSwitch Login"
eRecruiter	intitle: "OnLine Recruitment Program - Login" - johnny .ihackstuff
eXist	intitle: "eXist Database Administration" -demo
Extranet login pages	intitle: "EXTRANET login" -.edu -.mil -.gov - johnny.ihackstuff
eZ publish	Admin intitle: "eZ publish administration"
EZPartner	intitle: "EZPartner" -netpond

（续）

登录入口	查　询
Fiber Logic Management	"Web-Based Management" "Please input password to login" -inurl:johnny.ihackstuff.
Flash Operator Panel	intitle: "Flash Operator Panel" -ext:php -wiki - cms - inurl:asternic -inurl:sip -intitle:ANNOUNCE -inurl:lists
FlashChat	FlashChat v4.5.7
Free Perl Guestbook (FPG)	ext:cgi intitle: "control panel" "enter your owner password to continue!"
Generic	inurl:login.asp
Generic	inurl:/admin/login.asp
Generic	"please log in"
Generic	"This section is for Administrators only. If you are an administrator then please"
Generic	intitle: "Member Login" "NOTE: Your browser must have cookies enabled in order to log into the site." ext:php OR ext:cgi
Generic (with password)	intitle: "please login" "your password is *"
GNU GNATS	inurl:gnatsweb.pl
GradeSpeed	inurl: "gs/adminlogin.aspx"
GreyMatter	"login prompt" inurl:GM.cgi
Group-Office	intitle:Group-Office "Enter your username and password to login"
HostingAccelerator ControlPanel	"HostingAccelerator" intitle: "login" + "Username" - "news" -demo
HP WBEM Clients	intitle: "*- HP WBEM Login" | "You are being prompted to provide login account information for *" | "Please provide the information requested and press
H-SPHERE	intext: "Welcome to" inurl: "cp" intitle: "HSPHERE" inurl: "begin.html" -Fee
IBM TotalStorage Open Software	intext: "Storage Management Server for" intitle: "Server Administration"
IBM WebSphere	allinurl:wps/portal/ login
Icecast	intext: "Icecast Administration Admin Page" intitle: "Icecast Administration Admin Page"
iCMS	intitle: "Content Management System" "user name" | "password" | "admin" "Microsoft IE 5.5" -mambo -johnny.ihackstuff
iCMS	intitle: "Content Management System" "user name" | "password" | "admin" "Microsoft IE 5.5" -mambo-johnny.ihackstuff
iCONECTnxt	"iCONECT 4.1 :: Login"
IlohaMail	intitle:ilohamail intext: "Version 0.8.10" "Powered by IlohaMail"

（续）

登录入口	查　询
IlohaMail	intitle:ilohamail "Powered by IlohaMail"
IMail Server	"IMail Server Web Messaging" intitle:login
INDEXU	+ "Powered by INDEXU" inurl:(browse|top_rated|power)
Inspanel	"inspanel" intitle: "login" - "cannot" "Login ID" - site:inspediumsoft.com
Intranet login pages	intitle: "Employee Intranet Login"
iPlanet Messenger Express	"This is a restricted Access Server" "Javascript Not Enabled!" | "Messenger Express" -edu -ac
I-Secure	intitle: "i-secure v1.1" -edu
ISPMan	intitle: "ISPMan : Unauthorized Access prohibited"
Jetbox	Login ("Powered by Jetbox One CMS" | "Powered by Jetstream *")
Kerio Mail server	inurl: "default/login.php" intitle: "kerio"
Kurant StoreSense admin logon	intitle: "Kurant Corporation StoreSense" filetype:bok
Lights Out	"Establishing a secure Integrated Lights Out session with" OR intitle: " Data Frame - Browser not HTTP 1.1 compatible" OR intitle: "HP Integrated Lights-
Linux Openexchange Server	filetype:pl "Download: SuSE Linux Openexchange Server CA"
Listmail	intitle: "ListMail Login" admin -demo
Lotus Domino	inurl:names.nsf?opendatabase
Lotus Domino Web Administration.	inurl: "webadmin" filetype:nsf
MailEnable Standard Editions	inurl:mewebmail
MailMan	intitle: "MailMan Login"
Mailtraq WebMail	intitle: "Welcome to Mailtraq WebMail"
Mambo	inurl:administrator "welcome to mambo"
MDaemon	intitle: "WorldClient" intext: "(2003|2004) Alt-N Technologies."
Merak Email Server	"Powered by Merak Mail Server Software" - .gov -.mil -.edu -site:merakmailserver.com - johnny.ihackstuff
Merak Email Server	intitle: "Merak Mail Server Web Administration" -ihackstuff.com
MetaFrame Presentation Server	inurl:Citrix/MetaFrame/default/default.aspx
Microsoft Certificate Services	intitle: "microsoft certificate services"
Authority (CA)	inurl:certsrv
Microsoft CRM Login portal.	"Microsoft CRM : Unsupported Browser Version"
Microsoft Outlook or Microsoft Exchange	allinurl: "exchange/logon.asp"
Microsoft Outlook or Microsoft Exchange	inurl: "exchange/logon.asp" OR intitle: "Microsoft Outlook Web Access - Logon"
Microsoft Software Update Services	inurl:/SUSAdmin intitle: "Microsoft Software Update Services"

（续）

登录入口	查　询
Microsoft's Remote Desktop Web Connection	intitle:Remote.Desktop.Web.Connection inurl:tsweb
Midmart Messageboard	"Powered by Midmart Messageboard" "Administrator Login"
Mikro Tik Router	intitle: "MikroTik RouterOS Managing Webpage"
Mitel 3300 Integrated Communications Platform (ICP)	"intitle:3300 Integrated Communications Platform" inurl:main.htm
Miva Merchant	inurl:/Merchant2/admin.mv | inurl:/Merchant2/admin.mvc | intitle: " Miva Merchant Administration Login" - inurl:cheapmalboro.net
Monster Top List	"Powered by Monster Top List" MTL numrange:200-
MX Logic	intitle: "MX Control Console" "If you can't remember"
Neoteris Instant Virtual Extranet (IVE)	inurl:/dana-na/auth/welcome.html
Netware servers (v5 and up)	Novell NetWare intext: "netware management portal version"
Novell Groupwise	intitle:Novell intitle:WebAccess "Copyright *-* Novell, Inc"
Novell GroupWise	intitle: "Novell Web Services" intext: "Select a service and a language."
Novell GroupWise	intitle: "Novell Web Services" "GroupWise" - inurl: "doc/11924" -.mil -.edu -.gov -filetype:pdf
Novell login portals	intitle: "welcome to netware *" - site:novell.com
oMail-webmail	intitle: "oMail-admin Administration-Login" - inurl:omnis.ch
Open groupware	intitle:opengroupware.org "resistance is obsolete" "Report Bugs" "Username" "password"
Openexchange Server	intitle: "SuSE Linux Openexchange Server" "Please activate JavaScript!"
Openexchange Server	inurl: "suse/login.pl"
OpenSRS Domain Management System	"OPENSRS Domain Management" inurl:manage.cgi
Open-Xchange 5	intitle:open-xchange inurl:login.pl
Oracle Single Sign-On solution	inurl:orasso.wwsso_app_admin.ls_login
Oscommerce Admin	inurl: "/admin/configuration. php?" Mystore
Outlook Web Access Login Portal	inurl:exchweb/bin/auth/owalogon.asp
Ovislink	intitle:Ovislink inurl:private/login
pcANYWHERE EXPRESS Java Client	"pcANYWHERE EXPRESS Java Client"
Philex	intitle: "Philex 0.2*" -script -site:freelists.org
Photo Gallery Managment Systems	"Please authenticate yourself to get access to the management interface"
PhotoPost	-Login inurl:photopost/uploadphoto.php
PHP Advacaned TRansfer	intitle: "PHP Advanced Transfer" inurl: "login.php"
PHP iCalendar	intitle: "php icalendar administration" - site:sourceforge.net
PHP iCalendar	intitle: "php icalendar administration" - site:sourceforge.net

（续）

登录入口	查　询
PHP Poll Wizard 2	Please enter a valid password! inurl:polladmin
PHP121	inurl: "php121login.php"
PHPhotoalbum	intitle: "PHPhotoalbum-Upload" inurl: "PHPhotoalbum/upload"
PHPhotoalbum	inurl:PHPhotoalbum/statistics intitle: "PHPhotoalbum - Statistics"
phpMySearch	inurl:search/admin.php
PhpNews	intitle:phpnews.login
phpPgAdmin	intitle: "phpPgAdmin - Login" Language
PHProjekt	intitle: "PHProjekt - login" login password
PHPsFTPd	"Please login with admin pass" - "leak" -sourceforge
PhpWebMail	filetype:php login (intitle:phpWebMail|WebMail)
Plesk	intitle:plesk inurl:login.php3
Plesk	inurl:+:8443/login.php3
Polycom WebCommander	inurl:default.asp intitle: "WebCommander"
Postfix	intext: "Mail admins login here to administrate your domain."
Postfix Admin login pages	inurl:postfixadmin intitle: "postfix admin" ext:php
Qmail	intext: "Master Account" "Domain Name" "Password" inurl:/cgi-bin/qmailadmin
Qmail	intext: "Master Account" "Domain Name" "Password" inurl:/cgi-bin/qmailadmin
Quicktime streaming server	inurl: "1220/parse_xml.cgi?"
Real Estate	intitle: "site administration: please log in" "sitedesigned by emarketsouth"
RemotelyAnywhere	inurl:2000 intitle:RemotelyAnywhere - site:realvnc.comg
Request System	(inurl: "ars/cgi-bin/arweb?O=0" | inurl:arweb.jsp)
RT	intitle:Login intext: "RT is * Copyright"
rymo	(intitle: "rymo Login")|(intext: "Welcome to rymo")-family
Sak Mail	intitle:endymion.sak.mail.login.page | inurl:sake.servlet
SalesLogix	inurl: "/slxweb.dll/external?name= (custportal|webticketcust)"
SAP Internet Transaction Servers	intitle: "ITS System Information" "Please log on to the SAP System"
ServiceDesk	intitle: "AdventNet ManageEngine ServiceDesk Plus" intext: "Remember Me"
SFXAdmin	intitle: "SFXAdmin - sfx_global" | intitle: "SFXAdmin - sfx_local" | intitle: "SFXAdmin - sfx_test"
Shockwave (Flash) login	inurl:login filetype:swf swf
SHOUTcast	intitle: "SHOUTcast Administrator" inurl:admin.cgi

（续）

登录入口	查　询
Sift Group	intitle: “Admin login” “Web Site Administration” “Copyright”
SilkRoad Eprise	inurl:/eprise/
SilkyMail	(intitle: “SilkyMail by Cyrusoft International, Inc
SquirrelMail	inurl:login.php “SquirrelMail version”
SquirrelMail	“SquirrelMail version” “By the SquirrelMail Development Team”
SQWebmail.	inurl:/cgi-bin/sqwebmail?noframes=1
Sun Cobalt RaQ	“Login - Sun Cobalt RaQ”
Supero Doctor III Remote Management	intitle: “Supero Doctor III” -inurl:supermicro
Surgemail	“SurgeMAIL” inurl:/cgi/user.cgi ext:cgi
Synchronet Bulletin Board System	intitle:Node.List Win32.Version.3.11
SysCP	“SysCP - login”
Tarantella	“ttawlogin.cgi/?action=”
TeamSpeak	intitle: “teamspeak server-administration
Terracotta web manager	“You have requested to access the management functions” -.edu
This finds login portals for Apache Tomcat, an open source Java servlet container which can run as a standalone server or with an Apache web server.	intitle: “Tomcat Server Administration”
Topdesk	intitle: “TOPdesk ApplicationServer”
TrackerCamÃ	intitle:(“TrackerCam Live Video”)|(“TrackerCam Application Login”)|(“Trackercam Remote”)- trackercam.com
TUTOS	intitle: “TUTOS Login”
TWIG	intitle: “TWIG Login”
TYPO3	inurl: “typo3/index.php?u=” -demo
UBB.classic	inurl:cgi-bin/ultimatebb.cgi?ubb=login
UBB.threads	(intitle: “Please login - Forums powered by UBB.threads”) |(inurl:login.php “ubb”)
UebiMiau	“Powered by UebiMiau” -site:sourceforge.net
Ultima Online game.	filetype:cfg login “LoginServer=”
UltiPro Workforce Management	inurl: “utilities/TreeView.asp”
Usermin	“Login to Usermin” inurl:20000
vBulletin	inurl:/modcp/ intext:Moderator+vBulletin
vBulletin Admin Control Panel	intext: “vbulletin” inurl:admincp
VHCS	“VHCS Pro ver” -demo
vHost	intitle: “vhost” intext: “vHost . 2000-2004”
VISAS	intitle: “Virtual Server Administration System”

（续）

登录入口	查　询
VisNetic WebMail	intitle: “VisNetic WebMail” inurl: “/mail/”
VitalQIP Web Client	intitle: “VitalQIP IP Management System”
VMware GSX Server	intitle: “ VMware Management Interface:” inurl: “vmware/en/”
VNC	“VNC Desktop” inurl:5800
VNC	intitle: “VNC viewer for Java”
VOXBOX	intitle:asterisk.management.portal web-access
webadmin.	filetype:php inurl: “webeditor.php”
WebConnect	inurl:WCP_USER
Web-cyradm	intitle: “web-cyradm”	“by Luc de Louw” “This is only for authorized users” -tar.gz -site:webcyradm.org -johnny.ihackstuff
WebEdit	inurl:/webedit.* intext:WebEdit Professional - html
WebExplorer Server	“WebExplorer Server - Login” “Welcome to WebExplorer Server”
Webmail	intitle:Login * Webmailer
Webmail	inurl:webmail./index.pl “Interface”
Webmail	intitle: “Login to @Mail” (ext:pl|inurl: “index”)-dwaffleman
Webmail	intitle:IMP inurl:imp/index.php3
Webmail	intitle: “Login to @Mail” (ext:pl|inurl: “index”)-dwaffleman
Webmin	inurl: “:10000” intext:webmin
WebMyStyle	(intitle: “ WmSC e-Cart Administration”)|(intitle: “WebMyStyle e-Cart Administration”)
WEBppliance	inurl:ocw_login_username
WebSTAR	“WebSTAR Mail - Please Log In”
W-Nailer	uploadpics.php?did= -forum
WorkZone Extranet Solution	intitle: “EXTRANET * - Identification”
WRQ Reflection	filetype:r2w r2w
WWWthreads	(intitle: “Please login - Forums powered by WWWThreads”)| (inurl: “wwwthreads/login.php ”)| (inurl: “wwwthreads/login.pl?Cat=”)
xams	intitle: “xams 0.0.0..15 - Login”
XcAuction	intitle: “XcAuctionLite”	“DRIVEN BY XCENT” Lite inurl:admin
XMail	intitle: “XMail Web Administration Interface” intext:Login intext:password
Zope Help System	intitle: “Zope Help System” inurl:HelpSys
ZyXEL Prestige Router	intitle: “ZyXEL Prestige Router” “Enter password”

登录入口提供了大量可用于漏洞评估过程的信息。第4章讲解了更多关于如何从这些页面中获取尽可能多的信息的细节。

使用并且查找各种Web工具

虽然Google强大且灵活，但它并不是无所不能的。有时候，如果不用Google反而更简单。当不使用Google来执行类似于WHOIS查询，“pings”，traceroute以及端口扫描等任务时，更为简单方便。有很多具有这些功能的工具，但是通过一些灵活的Google搜索就可以执行这些费力的功能，同时也保证了Google黑客所希望的匿名级别。考虑一个叫做NQT的工具，即网络查询工具（Network Query Tool），如图8-23所示。

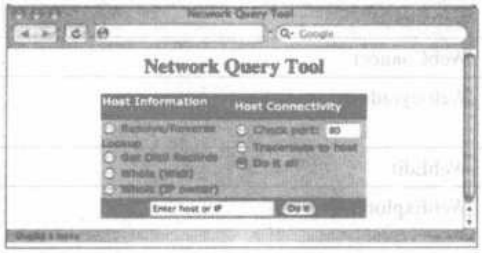


图8-23 网络查询工具提供了多个有趣的选项

默认的NQT安装允许任何网络用户执行IP主机名和地址查询，DNS查询，WHOIS查询，端口测试与traceroute。NQT是一个基于Web的应用程序，意思是说任何能够访问该页面的用户都可以对任意目标执行这些功能。对安全研究人员而言，它无疑是一个十分方便的工具。NQT的功能看起来是源于运行NQT应用程序的网站。其网站服务器会掩盖用户的真实地址。如果使用一个匿名代理服务器，则可以进一步的隐藏用户的真实身份。

我们可以利用一个非常简单的查询来使用Google查找运行NQT程序的服务器。NQT程序通常都被称作nqt.php，而且在它的默认配置还显示了标题“Network Query Tool”。诸如inurl:nqt.php intitle:“Network Query Tool”的简单查询会返回很多结果，如图8-24所示。

图8-24 使用Google来查找安装NQT的服务器

在提交了这个查询之后，就可以很容易地通过在结果页面上点击结果链接来查找到一个可用的NQT程序。但是，NQT程序也打接受远程的POST请求，这意味着可以从你的Web服务器发送一条NQT“命令”给foo.com服务器，而foo.com服务器会为你执行命令。如果你认为这似乎没有什么意义，那么不妨来考虑这样一个事实，即这可以简单地扩展NQT的布局与能力。例

如，我们可以很容易地把一个NQT制作为“回转装置”以对一个目标执行NQT命令，要实现这一点，需要考虑一个互联网NQT服务器。我们来看看这是如何处理的。

首先，我们从图8-24中所示的结果页面中提取运行NQT程序的网站列表。考虑如下的Linux/Mac OS X命令：

```
lynx -dump "
http://www.google.com/search?q=inurl:nqt.php+%22Network+\
Query+Tool%22&num=100" | grep "nqt.php$" | grep -v google |
awk '{print $2}' | sort -u
```

这一命令会取出Google查询inurl:nqt.php intitle:“Network Query Tool”的100条结果，在每一行的结尾查找单词nqt.php，删除任何包含单词google的行，打印出列表中的第二个域（即NQT网站的URL），并且对列表进行分类。这个命令不会捕捉到包含参数的NQT URL（因为nqt.php不是链接中的最后一个单词），但它能产生如下的整洁的输出：

```
http://bevmo.dynsample.org/uptime/nqt.php
http://biohazard.sifsample7.com/nqt.php
http://cahasample.com/nqt.php
http://samplehost.net/resources/nqt.php
http://linux.sample.nu/phpwebsite_v1/nqt.php
http://noc.bogor.indo.samplenet.id/nqt.php
http://noc.cbn.samplenet.id/nqt.php
http://noc.neksample.org/nqt.php
http://portal.trgsample.de/network/nqt.php
```

我们可以通过在前面的sort命令后加上>> nqtfile.txt来把这个输出保存到一个文件中。既然我们已经得到了一个有效的NQT服务器列表，那么我们还需要一份NQT代码，来生成如图8-23所示的界面。这个带有按钮和“enter host or IP”域的界面就是我们的“回转装置”的界面。得到这个界面的一份副本和查看一个现有的nqt.php网页（nqtfile.txt文件中的网站列表）源代码一样容易，并且把其HTML内容保存到我们自己的Web服务器上的一个叫做rotator.php的文件中。此时，在Web服务器的同一目录下存在两个文件一个是包含了NQT服务器列表的nqtfile.txt文件，一个是包含了NQT的HTML源代码的rotator.php文件。我们通过修改rotator.php文件中的某一行来创建我们的“回转装置”程序。这一行是NQT输入表单的第一行：

```
<form method="post" action="/nqt.php">
```

这一行的意思是，当按下“Do it”按钮时，表单数据将会被发送给一个叫做nqt.php的脚本。如果我们把这个表单域改为<form method=“post” action=“http://foo.com/nqt.php”>，那么我们的回转装置程序就会把NQT命令发送到foo.com上的NQT程序，并按照我们的要求执行命令。我们打算再做进一步的研究，即插入一段可以从nqtfile.txt文件中读出一个随机站点的PHP代码，这段代码看起来如下（为了更清楚而列出行号）：

```
1.  <?php
2.  $array = file("./nqtsites.txt");
3.  $site = substr($array[rand(0,count($array)-1)],0,-1);
4.  print "<form method=\"post\" action=$site<br>";
```

```
5.    print "Using NQT Site: $site for this session.<br>";
6.    print "Reload this page for a new NQT site.<br><br>";
7.    ?>
```

这段PHP代码段是用来替换原始NQT HTML代码中的<form method="post" action="/nqt.php">这一行的。第1行表示即将开始一段PHP代码。由于rotator.php文件的其余部分都是HTML代码，所以这一行和第7行结束这段PHP的代码是必需的。第2行读取文件nqtsites.txt，把文件中的每一行的内容（一个NQT网站的URL）赋给一个数组元素。第3行是为了更容易读懂而单独提出来的，它从nqtsites.txt中随机任取一行赋给变量$site。第4行输出原始form行的修改版本，并把action目标改为指向一个随机的远程NQT站点。第5行和第6行只是简单地输出一些有用的消息，比如所选择的NQT站点，加载一个新的NQT站点的操作指令。rotator.php脚本文件中的下一行应该是画出主要的NQT表格的table行。把rotator.php保存并在浏览器中查看以后，可以看到类似于图8-25的界面。

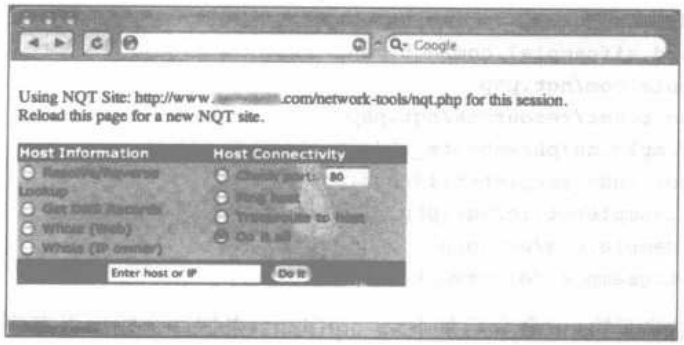


图8-25 实现NQT回转程序

我们的回转程序除了开始两行的文本之外，其他看起来也很像标准的NQT程序界面，当选择了“check port”，主机域便会填上www.microsoft.com，然后再点击Do It按钮，就可以转向远程NQT服务器所显示的结果页面，即端口80确实是打开的，而且也接受连接，如图8-26所示。

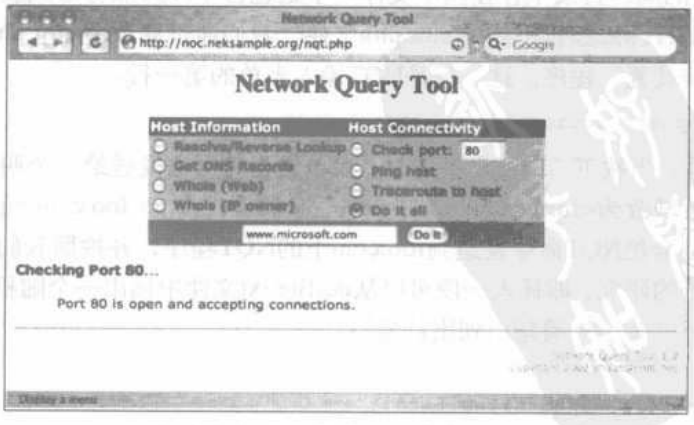


图8-26 NQT“回转程序”的输出

举这个例子是为了说明Google可以用来辅助许多基于Web的应用程序的使用。而实现这一点时所需要的仅仅是知道如何使用Google以及一定的创造力。

8.4 瞄准使用Web的网络设备

Google也可以用来检测许多使用Web（Web-enabled）网络设备。许多网络设备都预安装有一个Web接口以允许管理员查询设备的状态或者通过Web浏览器来改变设备的设置。虽然这样很方便，而且也可以通过使用SSL的连接来保证安全，但是如果某个设备的Web接口被Google抓取到——即使这种设备可能很少，也可以添加到一个网络映射的创建中。例如，查询intitle:“BorderManager information alert”就能够找到存在Novell BorderManager代理或者防火墙的服务器，如图8-27所示。

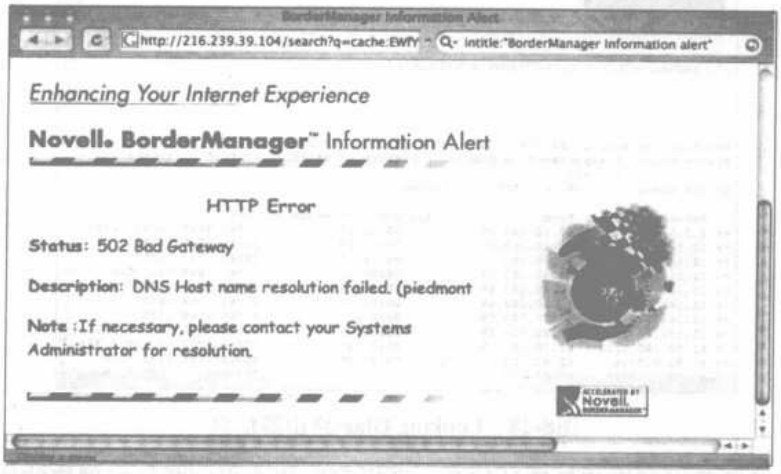


图8-27 Google泄露了Novell BorderManager代理或者防火墙

一个狡猾的攻击者就可以利用这种微不足道的设备的存在来实施针对目标网络的攻击。例如，如果这个设备是一个代理服务器，那么攻击者就可以通过和该服务器的连接来访问一个可信网络内部的机器。除此之外，攻击者还可能会搜索这种产品的任何公开的漏洞以试图直接利用这一设备。虽然可以用这种方法来查找到许多不同的设备，但是通常使用网络统计程序的输出来获取IP和网络数据更为容易，我们将在下一节中看到这一点。为了了解利用这一技术可以得到的设备类型，不妨考虑类似于“Version Info”“Boot Version”“Internet Settings”之类可以查找Belkin Cable/DSL路由器的查询；intitle:“wbem” compaq login可以用来查找HP Insight Management Agents；intitle:“lantronix web-manager”可以用来查找Lantronix web-manager；inurl:tech-support inurl:show Cisco或者intitle:“switch home page”“cisco systems”“Telnet - to”可以查找各种Cisco的产品；或者intitle:“axis storpoint CD” intitle:“ip address”可以用来查找Axis StorPoint服务器。这里的每个查询都会泄露出它们所安装的网络的各种信息。

8.5 查找各种网络报告

除了直接把网络设备作为目标之外，也可以用Google来查找各种网络文档和状态报告，它们能够让外部人员访问所有的信息，从网络上的IP地址到完整的、可用的网络图表。例如，查询“Looking Glass”(inurl:“lg/” | inurl:lookingglass)能够查找到显示路由统计信息的looking glass服务器，如图8-28所示。

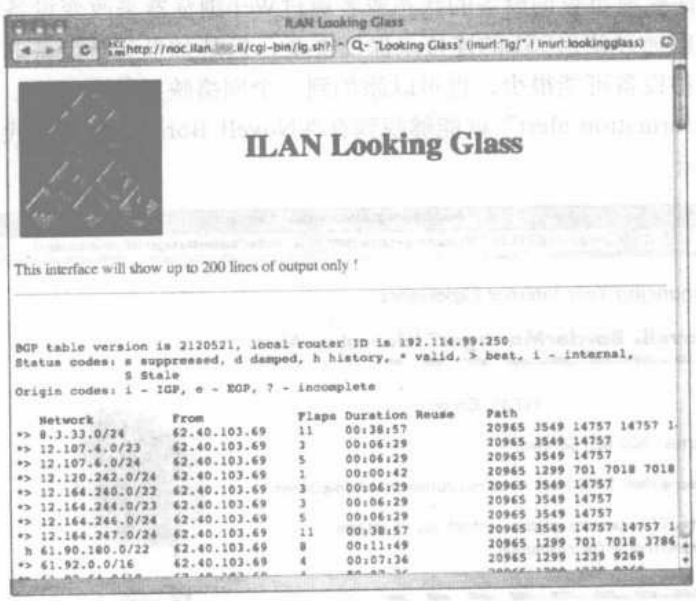


图8-28 Looking Glass路由器信息

ntop程序可以显示出网络流量统计信息，所以可以用它来判断目标网络的体系结构。查询intitle:“Welcome to ntop!”可以查找到公开ntop程序的服务器，它产生的输出，如图8-29所示。

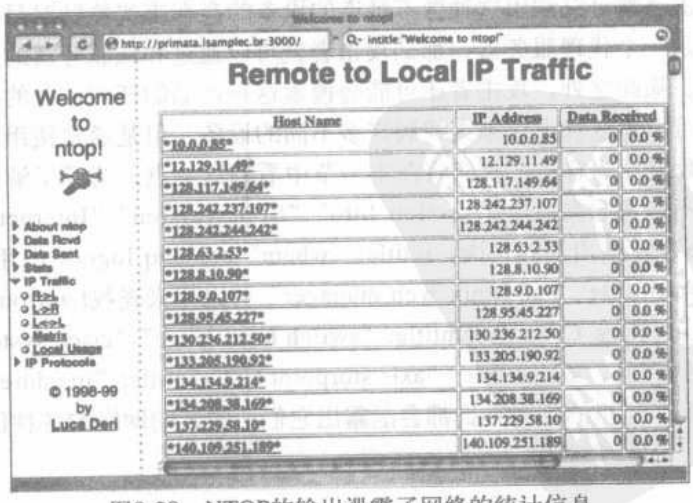


图8-29 NTOP的输出泄露了网络的统计信息

实际上，任何基于Web的网络统计数据包信息都可以用Google查找到。表8-11中给出几个取自Google Hacking数据库（GHDB）中的例子，它们可以用来查找各种网络文档。

表8-11 取自GHDB的网络文档示例

查 询	设备/报告
intitle: "statistics of" "advanced web statistics"	awstats显示Web服务器的统计信息
intitle: "Big Sister" + "OK Attention Trouble"	Big Sister程序会泄露网络信息
inurl: "cacti" +inurl: "graph_view.php" + "Settings TreeView" -cvs -RPM	cacti泄露了内部网络的信息，包括体系结构、主机和服务
inurl:fcgi-bin/echo	fastcgi echo程序泄露详细的服务器信息
"These statistics were produced by getstats"	Getstats 程序泄露服务器的统计信息
inurl: "/cricket/grapher.cgi"	grapher.cgi显示出了如配置、服务和带宽等网络信息
intitle: "Object not found" netware "apache 1.."	HP交换机Web接口
((inurl:ifgraph "Page generated at") OR ("This page was built using ifgraph"))	ifGraph SNMP数据收集器
"Looking Glass" (inurl: "lg/" | inurl:lookingglass)	Looking Glass网络统计信息输出
filetype:reg "Terminal Server Client"	Microsoft终端服务连接设置注册表文件泄露证书和配置数据
intext: "Tobias Oetiker" "traffic analysis"	MRTG分析页面泄露各种网络统计信息
intitle: "Welcome to ntop!"	ntop程序显示当前的网络用法
inurl: "smb.conf" intext: "workgroup" filetype:conf	Samba配置文件泄露了服务和网络数据
intitle: "Ganglia" "Cluster Report for"	服务器集群报告
intitle: "System Statistics" "System and Network Information Certer"	SNIC泄露了内部网络信息，包括网络配置，ping时间，服务和主机信息
intitle: "ADSL Configuration page"	SolWise ADSL Modem网络统计信息
"cacheserverreport for" "This analysis was produced by calamaris"	Squid缓存服务器报告
inurl:vbstats.php "page generated"	vbstats report泄露了服务器统计信息
filetype:vsd vsd network -samples -examples	Visio网络图表

这类信息在安全审计中是十分有用的信息，能够节约许多时间，但是要意识到任何以这种方式查找到的信息在用于各类完整的报告之前都必须进行验证。

8.6 查找网络硬件

一个连接到网络的设备通常都具有某种Web页面。如果该设备连接到了互联网上，并且曾经存在一个到该设备的Web页面的链接，那么很有可能该页面存在于Google的数据库之中，等待某个灵巧的查询找到它。正如我们在第5章中讨论的那样，这些页面能够泄露关于目标网络的信息，如图8-30所示。这类信息在映射目标网络过程中会发挥相当重要的作用。

所有类型的设备都可以连接到网络。这些设备，从交换机和路由器到打印机，甚至是防火墙，对于任何对网络侦察感兴趣的攻击者来说都是不小的发现，而一些如Webcam（网络摄像

头）之类的设备对攻击者来说也是很有趣的发现。

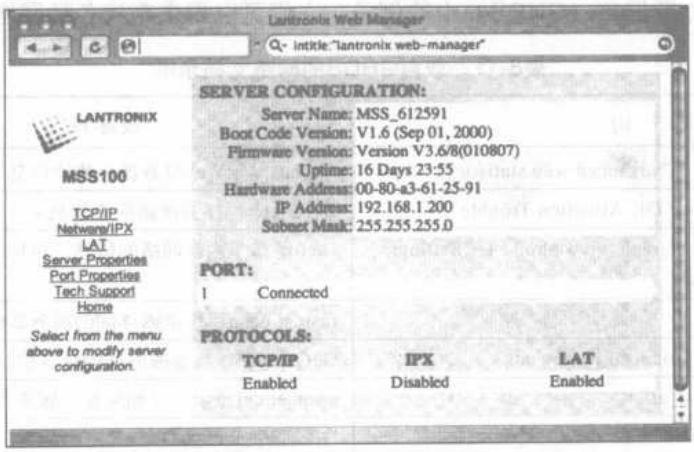


图8-30 网络设备的Web页面泄露了网络数据

通常来讲，连接到网络的Webcam并不能当作是一种安全威胁，而更多是作为娱乐之用。但是，还要知道一些事情。首先，一些公司认为让客户看看他们的工作场所是很时髦而且很酷的。Netscape在它的全盛时期就以此而著名。位于这些公司工作场所的Webcam显然是经过上级管理人员允许了的。如果你的工作是进行一次物理评估，那么查看工作场所的内部情形是十分有益的。其次，有许多Webcam通常是放在工作场所外面的，如图8-31所示。这类摄像头对物理评估而言是十分有用的。同样，不要忘记雇员在工作时所作的事情并不一定能反映他在个人时间所作的事情。如果你搜索到一名雇员的个人Web空间，那么很有可能空间上存在这类设备。

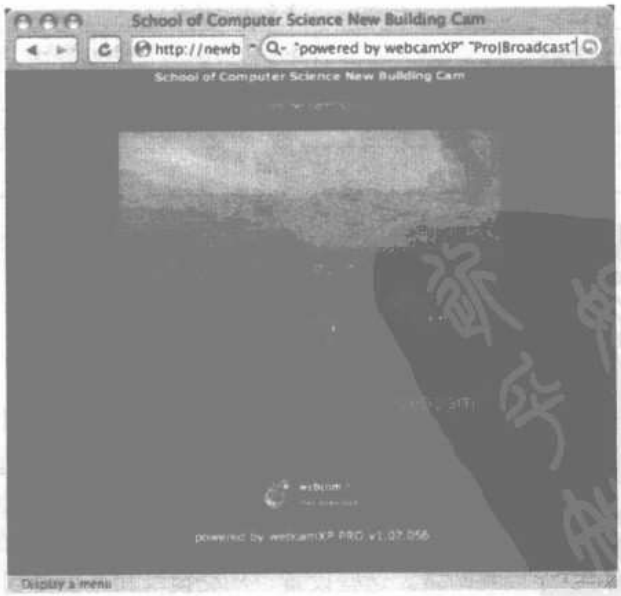


图8-31 将Webcam放在工作场所之外

许多近些年生产的网络打印机都安装了某种基于Web的接口。如果这些设备（或者甚至是和这些设备一同提供的文档或驱动程序）由某个Web页面来链接的，那么就可以用各种Google查询来搜索它们。

一旦搜索到网络打印机，那么它就能够给攻击者提供大量的信息。如图8-32所示，网络打印机通常都会列出周围网络的详细信息、命名约定等更多的信息。许多通过Google搜索查找到的设备运行的仍然是默认的、不安全的配置，不需要用户名或者口令就可以控制这些设备。在最坏的情况下，攻击者能够浏览打印作业，甚至强令这些打印机把文件保存起来或者发送网络命令。

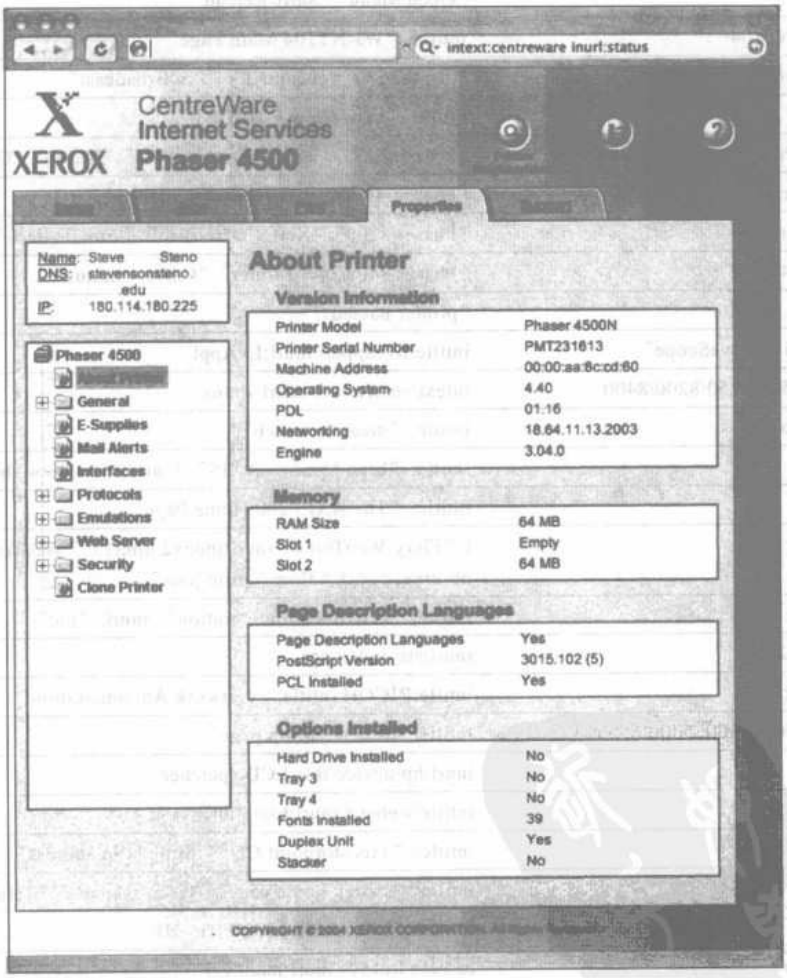


图8-32 网络打印机提供了大量的详细信息

表8-12给出了一些可以用来搜索各种网络设备的查询。

表8-12 搜索各种网络设备的查询

网络设备	查 询
AXIS 2400	inurl:indexFrame.shtml Axis
PhaserLink Printers	intitle: "View and Configure PhaserLink"
Panasonic Network Cameras	inurl: "ViewerFrame?Mode="
Sony NC RZ30 Camers	SNC-RZ30 HOME
Sony NC RZ20 Cameras	intitle:snc-z20 inurl:home/
Mobotix netcams	(intext: "MOBOTIX M1" | intext: "MOBOTIX M10") intext: "Open Menu" Shift-Reload
Panasonic WJ-NT104	intitle: "WJ-NT104 Main Page"
XP PRO Webcams	"powered by webcamXP" "Pro|Broadcast"
AXIS Cameras	intitle: "Live View / - AXIS"
Phaser 6250N Printer	"Phaser 6250" "Printer Neighborhood" "XEROX CORPORATION"
Xerox Phaser Printer	"Phaser740 Color Printer" "printer named: "
Phaser 8200 Printer	"Phaser 8200" "Xerox" "refresh" " Email Alerts"
Xerox Phaser 840	"Phaser 840 Color Printer" "Current Status"
Color Printer	"printer named:"
Canon "WebView LiveScope"	intitle:liveapplet inurl:LvAppl
Xerox Phaser 4500/6250/8200/8400	intext:centreware inurl:status
Linux Dreamboxes	intitle: "dreambox web"
Axis Netcams	intitle: "Live View / - AXIS" | inurl:view/view.sht
Axis 200	intitle: "The AXIS 200 Home Page"
Fiery WebTools	("Fiery WebTools" inurl:index2.html) | "WebTools enable * * observe, *, * * * flow * print jobs"
Konica Network Printer	intitle: "network administration" inurl: "nic"
Ricoh Aficio 1022	inurl:sts_index.cgi
Ricoh Afficio Printer	intitle:RICOH intitle: "Network Administration"
Canon ImageReady 3300, 5000 & 60000.	intitle: "remote ui:top page"
HP Printers.	inurl:hp/device/this.LCDispatcher
Webeye webcams.	intitle:webeye inurl:login.ml
AXIS StorPoint CD+.	intitle: "axis storpoint CD" intitle: "ip address"
Cisco Switches	intitle: "switch home page" "cisco systems" "Telnet - to"
HP switches	intitle: "DEFAULT_CONFIG - HP"
Linksys webcam	camera linksys inurl:main.cgi
My webcamXP server	intitle: "my webcamXP server!" inurl: ":8080"
Ricoh Aficio 2035(fax/scanner)	(inurl:webArch/mainFrame.cgi) | (intitle: "web image monitor" -htm -solutions)
Axis Network Camera	inurl:netw_tcp.shtml

（续）

网络设备	查　询
Tivo Devices	inurl:TiVoConnect?Command=QueryServer
Embedded DVR	intitle: "DVR Web client"
Panasonic Network Camera	site:.viewnetcam.com -www.viewnetcam.com
Toshiba netcams	intitle: "toshiba network camera - User Login"
CCTV webcams	"please visit" intitle: "i-Catcher Console" Copyright "iCode Systems"
AMX Netlink	WebControl intitle: "AMX NetLinx"
XeroxDocuPrint printer.	intitle: "Home" "Xerox Corporation" "Refresh Status"
Xerox 860 and 8200 Printers.	intext: "Ready with 10/100T Ethernet"
Lexmark printers	intext: "UAA (MSB)" Lexmark -ext:pdf
Axis Netcams	inurl:axis-cgi
SiteZap webcam	"Starting SiteZAP 6.0"
EvoCam	intitle: "EvoCam" inurl: "webcam.html"
Tandberg video conferencing appliances	intext: "Videoconference Management System" ext:htm
Novell Iprint	inurl: "ipp/pdisplay.htm"
Phaser printers	"Copyright (c) Tektronix, Inc." "printer status"
Xerox DocuPrint printer	intext: "MaiLinX Alert (Notify)" -site:networkprinters.com
Brother HL Printers	inurl: "printer/main.html" intext: "settings"
Axis Storpoint	axis storpoint "file view" inurl:/volumes/
Netsnap Online Cameras	intitle: "Live NetSnap Cam-Server feed"
V-Gear Bee Web Cameras	intitle: "V-Gear BEE"
Audio ReQuest home CD/MP3 player	intitle: "AudioReQuest.web.server"
CUPS Printers	inurl: ":631/printers" -php -demo
iVista Camera	intitle: "iVISTA.Main.Page"
Axis Video Cameras	
Linksys Wireless-G web cams.	inurl: "next_file=main_fs.htm" inurl:img inurl:image.cgi
SnapStream Digital Video Recorder	filetype:cgi transcoder.cgi
Axis Network Print Server	intitle: "Network Print Server" filetype:shtm (inurl:u_printjobs | inurl:u_server | inurl:a_server | inurl:u_generalhelp | u_printjobs)
Axis Network Print Server	intitle: "Network Print Server" intext: "http://www.axis.com" filetype:shtm
ActiveX webcam	intitle: "Browser Launch Page"
Sweex, Orite Web Cameras	allinurl:index.htm?cus?audio
EDSR video cameras	intitle: "EverFocus.EDSR.applet"
Epson Web Assist	intitle: "EpsonNet WebAssist Rev"
Brother printers	intitle: "Brother" intext: "View Configuration" intext: "Brother Industries, Ltd."

（续）

网络设备	查　询
Linksys webcams	intitle:Linksys site:ourlinksys.com
SupervisionCam	intitle: "supervisioncam protocol"
Vivotec webcams	inurl:camctrl.cgi
mmEye webcam	allintitle:Brains, Corp. camera
Dell ESW Printers	intitle: "Dell Laser Printer" ews
HomeSeer home automation server	intitle:HomeSeer.Web.Control | Home.Status.Events.Log
Samsung webthru cameras	"Webthru User Login"
Lexmark printers (4 models)	intitle: "Lexmark *" inurl:port_0
Aficio printers	inurl:/en/help.cgi "ID=*"
HP Officejet help page.	intitle:jdewshlp "Welcome to the Embedded Web Server!"
Xerox Phaser printers.	"display printer status" intitle: "Home"
GeoHttpServer	inurl:JPGLogin.htm
Winamp Servers	"About Winamp Web Interface" intitle: "Winamp Web Interface"
NeroNet Servers	intitle: "NeroNET - burning online"
Xerox (*Centre) Printers	ext:dhtml intitle: "document centre|(home)" OR intitle: "xerox"
Lexmark and Dell Printers	inurl: "port_255" -htm
Adobe's PrintGear	intext: "Powered by: Adobe PrintGear" inurl:admin
AVTech Video Web Server	intitle: "—- VIDEO WEB SERVER —-" intext: "Video Web Server" "Any time & Any where" username password
VPON (Video Picture On Net) video surveillance system	inurl:start.htm?scrw=
Dell Printers	intitle: "Dell *" inurl:port_0
Kpix Java Based Traffic Cameras	(cam1java)|(cam2java)|(cam3java)| (cam4java)| (cam5java)| (cam6java) -navy.mil -backflip -power.ne.jp
Mobile Cameras	inurl: "S=320x240" | inurl: "S=160x120" inurl: "Q=Mob"
Panasonic IP cameras	inurl: "CgiStart?page="
Dell and Lexmark Printers	intitle: "configuration" inurl:port_0
Dell Laser Printer M5200	intitle: "Dell Laser Printer M5200" port_0
AXIS 240 Camera Servers	intitle: "AXIS 240 Camera Server" intext: "server push" -help
Veo Observer Web Client	intitle: "Veo Observer Web Client"
Standalone Network Camera	intitle: "Java Applet Page" inurl:ml
DVR Systems	intitle: "WEBDVR" -inurl:product -inurl:demo
sensorProbe Environmental Monitoring Device	"Summary View of Sensors" | "sensorProbe8 v *" |"
iDVR Camera	intitle:iDVR -intitle: "com | net | shop" -inurl: "asp | htm | pdf | html | php | shtml | com | at | cgi | tv"
INTELLINET IP camera	intitle: "INTELLINET" intitle: "IP Camera Homepage"
StarDot netcam	intitle: "NetCam Live Image" -.edu -.gov - johnny. ihackstuff. com

（续）

网络设备	查　询
Netbotz devices	intitle: "netbotz appliance" -inurl:.php -inurl:.asp - inurl:.pdf - inurl:securitypipeline -announces
Phaser Network Printers	Phaser numrange:100-100000 Name DNS IP "More Printers" index help filetype:html | filetype:shtml
Orite 301 Netcams	intitle: "Orite IC301" |intitle: "ORITE Audio IPCamera IC-301" - the -a
Brimsoft webcam	intitle: "Biromsoft WebCam" -4.0 -serial -ask -crack -software - a -the -build -download -v4 -3.01 -numrange:1-10000
VisionGS Webcam	(intitle: "VisionGS Webcam Software")|(intext: "Powered by VisionGS Webcam") -showthread.php -showpost.php - "Search Engine" -computersglobal.com -site:g
IQeye netcam	intitle: "IQeye302 | IQeye303 | IQeye601 | IQeye602 | IQeye603" intitle: "Live Images"
Samsung printers	"This page is for configuring Samsung Network Printer" | printerDetails.htm
Intel Netport Express Print Server.	intitle: "SNOIE Intel Web Netport Manager" OR intitle: "Intel Web Netport Manager Setup/Status"
Express6 live video controller	Display Cameras intitle: "Express6 Live Image"
Sony SNT-V304 Video Network Station	intitle: "Sony SNT-V304 Video Network Station" inurl:hsrindex.shtml
Windows 2003 Remote Printing	inurl:Printers/ipp_0001.asp
Linksys wireless G Camera	inurl:/img/vr.htm
Sony DCS-950 Web Camera	DCS inurl: "/web/login.asp"
Dell laser printers	intitle: "Dell Laser Printer *" port_0 -johnny.ihackstuff
INTELLINET IP Camera	intitle: "::::: INTELLINET IP Camera Homepage :::::
Celestix Taurus Server	intext: "Welcome to Taurus" "The Taurus Server Appliance" intitle: "The Taurus Server Appliance"
Sharp printers	intitle: "AR-*" "browser of frame dealing is necessary"
Watchdogs WxGoos Camera	intitle: "WxGoos-" ("Camera image" | "60 seconds")
Nuvico DVR	intitle: "DVR Client" -the -free -pdf -downloads -blog - download -dvrtop
Hunt Electronics web cams	"OK logout" inurl:vb.htm?logout=1
EverFocus DVR	intitle: "Edr1680 remote viewer"
IVC Security Cameras	intitle: "IVC Control Panel"
MOBOTIX Cameras	(intitle:MOBOTIX intitle:PDAS) | (intitle:MOBOTIX intitle:Seiten) | (inurl:/pda/index.html +camera)
Netbotz devices	intitle: "Device Status Summary Page" -demo
iGuard Fingerprint Security System	intitle: "iGuard Fingerprint Security System"
Veo Observer XT	intitle: "Veo Observer XT" -inurl:shtml|pl|php| htm|asp| aspx| pdf|cfm -intext:observer

（续）

网络设备	查　询
EyeSpyFX or OptiCamFX Camera	(intitle:(EyeSpyFX|OptiCamFX) "go to camera")| (inurl:servlet/DetectBrowser)
MOBOTIX cameras	inurl:cgi-bin/guestimage.html
Sony SNC-RZ30 IP camera	intitle: "SNC-RZ30" -demo
Everfocus EDSR400	allintitle: EverFocus | EDSR | EDSR400 Applet
Everfocus EDR1680	allintitle:Edr1680 remote viewer
Everfocus EDR1600	allintitle: EDR1600 login | Welcome
Everfocus EDR400	allintitle: EDR400 login | Welcome
Boshe/Divar Net Cameras	intitle: "Divar Web Client"
Axis Cameras	intitle: "Live View / - AXIS" | inurl:view/view.shtml OR inurl:view/indexFrame.shtml | intitle: "MJPG Live Demo" | "intext:Select preset position"
Axis Cameras 2XXX Series	allintitle: Axis 2.10 OR 2.12 OR 2.30 OR 2.31 OR 2.32 OR 2.33 OR 2.34 OR 2.40 OR 2.42 OR 2.43 "Network Camera"
BlueNet Video Viewer	intitle: "BlueNet Video Viewer"
Stingray File Transfer Server	intitle: "stingray fts login" | (login.jsp intitle:StingRay)
Softwell Technology "Wit-Eye" DVR	allintitle: "DVR login"
WR Control Lite Multi-Camera View	inurl:wrcontrollite
Device	Query
Axis Video Server (CAM)	inurl:indexFrame.shtml Axis
AXIS Video Live Camera	intitle: "Live View / - AXIS"
AXIS Video Live View	intitle: "Live View / - AXIS" | inurl:view/view.sht
AXIS 200 Network Camera	intitle: "The AXIS 200 Home Page"
Canon Network Camera	intitle:liveapplet inurl:LvAppl
Mobotix Network Camera	intext: "MOBOTIX M1" intext: "Open Menu"
Panasonic Network Camera	intitle: "WJ-NT104 Main Page"
Panasonic Network Camera	inurl: "ViewerFrame?Mode="
Sony Network Camera	SNC-RZ30 HOME
Seyeon FlexWATCH Camera	intitle:flexwatch intext: "Home page ver"
Sony Network Camera	intitle:snc-z20 inurl:home/
webcamXP	"powered by webcamXP" "Pro|Broadcast"
Canon ImageReady	intitle: "remote ui:top page"
Fiery Printer Interface	("Fiery WebTools" inurl:index2.html) | "WebTools enable * * observe, *, * * * flow * print jobs"
Konica Printers	intitle: "network administration" inurl: "nic"
RICOH Copier	inurl:sts_index.cgi

（续）

网络设备	查　询
RICOH Printers	intitle:RICOH intitle: "Network Administration"
Tektronix Phaser Printer	intitle: "View and Configure PhaserLink"
Xerox Phaser (generic)	inurl:live_status.html
Xerox Phaser 6250 Printer	"Phaser 6250" "Printer Neighborhood" "XEROX CORPORATION"
Xerox Phaser 740 Printer phaserlink	"Phaser® 740 Color Printer" "printer named:"
Xerox Phaser 8200 Printer	"Phaser 8200" "© Xerox" "refresh" " Email Alerts"
Xerox Phaser 840 Printer	Phaser "®" p840 Color Printer
Xerox Centreware Printers	intext:centreware inurl:status
XEROX WorkCentre	intitle: "XEROX WorkCentre PRO - Index"

8.7 总结

攻击者使用Google有多种目的。攻击者可能拥有一个针对某个Web软件特定版本的漏洞利用程序，并且正在寻觅有漏洞的目标。有时攻击者可能已经确定了一个目标，并正在使用Google来搜索网络中其他设备的信息。在某些时候，攻击者只是在简单地搜索带有默认页面和程序的配置不当的Web设备，这种设备的安全性是很弱的。

目录列表可以提供设备上所用的软件版本信息。服务器和应用程序错误消息能够给攻击者提供大量的信息，而且可能是所有信息收集技术中最被低估的技术。默认页面、程序和文档不仅能够用来剖析目标，而且它们也表明服务器在某种程序上被忽略了，可能具有可以利用的漏洞。登录入口对正常用户而言是Web服务器的"前门"，然而也可以利用它来剖析目标，利用它来查找更多的关于目标所用的服务和功能的信息，而且还会吸引拥有相应的漏洞利用程序的攻击者。在某些情况下，管理员设置登录入口是用来允许远程访问服务器或者网络的。一旦这种类型的登录入口被攻陷，那么就会给入侵者提供一个入口。

Google可以用来查找或者扩大诸如NQT之类的基于Web的网络工具，NQT工具支持各种网络查询应用程序的远程执行。使用灵活的查询，Google甚至可以通过来自网络统计数据包的目标或者输出来查找正在使用的使用Web的网络设置。在基于网络的评估中，不论你的目标是什么，都可以使用Google来扩大现有的工具和技术的能力。

8.8 快速查找解决方案

查找并剖析Web服务器

- 目录列表和默认的服务器生成的错误消息能够提供关于服务器的详细信息。即使这些信息可以通过直接连接到服务器来得到，但是一名拥有针对特定版本的软件的漏洞利用程序的攻击者会使用Google查询来搜索相应的目标，而这些Google查询都是针对这类信息设计的。

- 服务器和应用程序错误消息暴露了大量的信息，从软件的版本和补丁级别到源代码片段以及关于系统进程和程序的信息。错误消息是最被低估的信息泄露途径之一。
- 默认页面、文档和程序能够泄露相应服务器的许多信息。它们能够说明服务器没有被很好地维护，拓展开来说，会由于维护不当造成一些漏洞。

查找登录入口

- 登录入口会吸引那些正搜索特定类型软件的攻击者。另外，登录入口也可以作为信息收集攻击的起点，因为大多数登录入口都是设计成对用户友好的，给新用户提供帮助文档和操作步骤的链接。管理登录入口和远程管理工具有时更加危险，尤其是当它们没有被正确的配置时。

查找网络硬件

- 各种网络设备都可以用Google查询搜索到。对某些攻击者而言，这些设备不仅只是满足他们技术上的好奇，许多由Web链接了的设备配置不当，而且安全审计人员通常会忽视那些可信的设备。网络摄像头就是一种被忽视的设备，它能够给攻击者提供目标的内部情况，即使只有相当少的一部分目标安装了网络摄像头。当网络打印机被攻陷时，会泄露大量的敏感信息，尤其是对攻击者而言，他们可以查看打印作业和网络信息。

使用和查找各种Web设备

- 使用Web的网络设备可以使用简单的Google查询来查找。
- 来自这些设备的信息可以用来帮助创建网络映射。

查找各种网络报告

- 网络统计报告可以使用简单的Google查询来查找。
- 来自这些报告的信息可以用来帮助创建网络映射。

8.9 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：我用的是IIS 6.0服务器，而且我不喜欢那些挂在自己站点的静态的HTTP 1.1错误页面，这会吸引恶意的攻击者对我的服务器产生兴趣。我怎样才能使用自定义的错误消息？

答：如果你现在还没有询问Google的习惯，那么就应该从现在开始养成这种习惯。认真地讲，你可以用这样的Google搜索，即site:microsoft.com “Configuring Custom Error Message” IIS 6.0来解决这一问题。在本书写作之时，第一条结果就是描述相应步骤的文章。这一步骤大概是打开IIS管理器，双击我的电脑（My Computer），在Web Sites文件夹上右击，然后选择属性（Properties），再参考自定义错误（Custom Errors）选项卡。

问：我用的是Apache服务器，而且我不喜欢那些在错误页面和目录列表上的服务器标记。我该怎样把这种功能取消掉？

答：要删除这种标记，只需在你的httpd.conf（通常位于/etc/httpd/conf/httpd.conf）文件中

找到包含下面一段代码的部分：

```
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#
ServerSignature On
```

其中的ServerSignature可以设置为Off，这样就可以删除服务器标记；或者设置为Email，它代表一个链接到ServerAdmin E-mail地址的E-mail链接。

问：我想到了一个搜索，但是这里却没有列出来。如果你擅长于使用Google，为什么我的搜索没有在这本书中出现呢？

答：本书是一本入门书籍，而不是一本参考书。除了本书中提到的搜索之外，还有许多的搜索，以至于无法在一本书中都囊括进去。这本书中列出的大部分的搜索都是在社区里许多人的共同努力之下创建的尽可能有效的搜索。幸运的是，这种个人之间的讨论社区创建了一个独特而且广泛的数据库，这个数据库是对外公开的，可以用于对抗这类威胁。搜索引擎Hacking（Search Engine Hacking）论坛和Google Hacking数据库（GHDB，Google Hacking Database）都位于http://johnny.ihackstuff.com。如果你想到了一个新的搜索，首先应该搜索一下这个数据库以确定它是唯一的。如果你认为它是唯一的，那么把它提交到论坛上，这样你的搜索就会被添加到数据库中。但是，Google搜索者，请您注意。Google Hacking是有趣而且令人着迷的。如果你提交了一个搜索，那么我觉得你会发现很难停下来。你只要问一问Google大师列表上的任何一个人，他们都会这么说。他们其中的一些人甚至在提交了10个或者20个唯一的搜索之后，都很难再停下来。你可以看看致谢那一页以了解对Google Hacking社区做出过重要贡献的用户列表。

问：NQT工具一次只能扫描一个端口，这个操作可以修改吗？

答：如果不修改远程NQT服务器的代码，那么这个任务就需要编写一个PHP循环程序，每次向NQT服务器请求一个端口。但是，要注意的是当执行一次实际的网络端口扫描任务时，甚至一个单独的端口都有可能起到关键的作用。针对许多不同的扫描任务，准备一个已知的常处于打开状态的端口列表是很有用的。

问：除了NQT之外，就没有一种能够扫描更大范围端口的基于Web的工具了吗？

答：如果你想要扫描大量的端口，那么可能最好使用一种标准的扫描器，例如namp。但是，为了再次使用Google，可以试试Jimmy Neutron在Google Hacking论坛上建议的查询inurl:portscan.php ("from Port" | "Port Range")。虽然现在这个查询得到的结果不多，但是谁知道将来这个查询会不会得到更多的结果！

问：打开网络设备的Web接口不好吗？

答：没有必要打开Web接口，总得说来有以下几点理由。首先，如果考虑可以用串行端口连接或者专用于管理的网络连接来更为安全地完成同样的任务时，打开Web接口就显示有些过

了。其次，小的设备只需要小的服务器，所以有些比较少用的Web服务器并没有像Apache一样得到很好的测试，例如，从安全的观点来看Axis cams的漏洞。第三，正如我们在这一章中看到的那样，如果管理员不小心，那么就可以用Google来找到（或者提交给Google）这些Web接口页面。这会给那些所有羽翼并未丰满的Google黑客打开一扇“水闸”。

问：我们的网络设备（路由器）不能从外部访问，这是否意味着是安全的？

答：即使它不能从WAN访问，它也可能由你的LAN中一个被攻陷的主机来访问。在用户新闻组或者技术论坛上提交网络设备的信息是有很大风险的。例如，试着搜索Google Hacking论坛上的hevnsnt所建议的intext:“enable secret 5 $”。然后在Google Group中搜索同样的内容。Cisco对那些口令实现了相当强度的加密，这是件好事，因为这些搜索经常泄露关于设备的敏感信息。

第9章 用户名、口令和其他秘密信息

9.1 简介

这一章的内容不是关于在评估过程中查找敏感数据的，而是关于为了搜索这些数据，那些“坏人”可能会做什么。这一章所提到的例子通常都代表了最常见的安全问题。黑客在每次攻击时都要瞄准这些信息。为了对抗这类攻击者，我们必须十分明确这种攻击会造成的最坏的可能。但是，我们也不用过于了解这些最坏的可能。请不要告诉那些有坏心思的人一些他们尚不知道的观念。

我们从学习一些可以用来挖掘用户名的查询开始，而用户名是大多数认证系统中不太重要的一部分。用户名的值经常被忽视，但是正如我们曾经讨论过的，攻击者可以通过灵活地运用哪怕是最少的、最没有危害的一点信息来攻破一个完整的造价数百万美元的安全系统。

然后，我们再来看看可以用来挖掘口令的查询。这些查询中有一些会泄露加密了的或者经过编码了的口令，攻击者要利用这些口令还需要再多做些工作。我们也会了解一些能够挖掘出明文口令的查询。即使在最初级的攻击者手中，这些查询也是最危险的几个。有什么比把一组用户名和口令交给攻击者能让攻击更为容易的?

在本章的最后，我们将讨论发掘高度敏感数据，例如信用卡信息以及用于实施身份盗取的信息，例如社保号码的现实可能性。在这里，我们的目标是探索抵抗这种非常现实的威胁的方法。因此，我们并不深入讨论和发掘财产信息等诸如此类相关的细节。如果你是一名“黑暗的”黑客，那么你可能需要自己挖掘这些东西。

9.2 搜索用户名

大多数认证系统都是使用用户名和口令来保护信息。为了达到突破这类保护的“前门”的目的，你需要判断用户名以及口令。如前所述，用户名同样也可用于社会工程学攻击中。

有许多方法可以用来判断用户名。在第4章中，我们会研究利用数据库错误消息来收集用户名的方法。在第8章中，我们研究了能够泄露各种信息，包括用户名的Web服务器和应用程序的错误消息。这些间接查找用户名的方法虽然很有帮助，但是攻击者能够直接用像“your username is”这样的简单查询来直接搜索用户名。这个查询词组能够搜索到描述用户名创建过程的帮助页面，如图9-1所示。

攻击者可以利用这种信息，再根据从其他信息源，如Google Group的帖子或者电话列表中收集到的信息，就可以假设出用户名。用户名也可以用于攻击的各种其他阶段，例如一次蠕虫垃圾邮件大战或者一次社会工程学攻击。攻击者可以从各种信息源收集用户名，表9-1列出了一些示例查询。

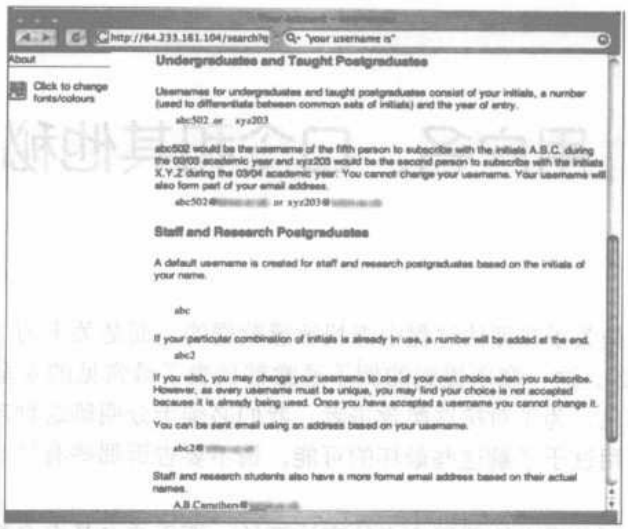


图9-1 帮助文档能够泄露用户名的创建过程

表9-1 搜索用户名的示例查询

查 询	描 述
inurl:admin inurl:userlist	普通的用户列表文件
inurl:admin filetype:asp inurl:userlist	普通的用户列表文件
inurl:php inurl:hlstats intext: Server Username	Half-life（游戏《半条命》）的统计文件，列出了用户名和其他信息
filetype:ctl inurl:haccess.ctl Basic	使用htaccess的Microsoft FrontPage会显示出Web用户证书
filetype:reg reg intext: “internet account manager”	Microsoft Internet Account Manager能够泄露用户名以及更多的信息
filetype:wab wab	Microsoft Outlook Express邮件地址薄
filetype:mdb inurl:profiles	Microsoft Access数据库包含（用户）的配置文件
index.of perform.ini	MIRC IRC ini文件能够列出IRC用户名和其他信息
inurl:root.asp?acs=anon	Outlook Mail Web 访问目录可以用来发掘服务器信息
filetype:conf inurl:proftpd.conf -sample	PROFTP FTP服务器配置文件泄露用户名和服务器信息
filetype:log username putty	PUTTY SSH客户端日志能泄露用户名和服务器信息
filetype:rdp rdp	Remote Desktop Connection文件泄露用户证书
intitle:index.of .bash_history	UNIX bash shell历史会泄露在bash命令提示符下输入的命令；而用户名通常作为参数字符串
intitle:index.of .sh_history	UNIX shell历史会泄露在shell命令提示符下输入的命令；而用户名通常作为参数字符串
“index of” lck	各种lock文件列出了用户当前正在使用的文件
+intext:webalizer +intext:Total Usernames +intext: “Usage Statistics for”	Webalizer Web统计页面列出了Web用户名和统计信息
filetype:reg reg HKEY_CURRENT_ USER username	导出的Windows注册表会泄露用户名和其他信息

Google搜索背景知识

搜索某种已知的文件名

要知道的是，有几种搜索某种已知文件名的方法。一种方法是基于查找目录列表中的文件，类似于intitle:index.of install.log。另外一种通常更好的方法是基于filetype操作符，例如filetype:log inurl:install.log。目录列表并不常见。Google会抓取目录列表中到一个文件的链接，这意味着filetype这种方法既可以找到目录列表的条目，也能找到Google以其他方法抓取的文件。

在某些情况下，可以从基于Web并检查Web活动的统计程序来收集用户名。Webalizer能够列出和一个Web服务器的用法相关的各种信息。可以用如intext:webalizer intext:“Total Usernames” intext:“Usage Statistics for”这样的查询来搜索Webalizer程序的输出文件。如图9-2所示，在显示出的信息中有用来连接到该Web服务器的用户名。但是，有些时候这些显示的用户名可能是无效的，“Visits”这一列列出了在捕捉期间一个用户账号被使用的次数。这使得攻击者能够容易地判断出哪些账号更可能是有效的。

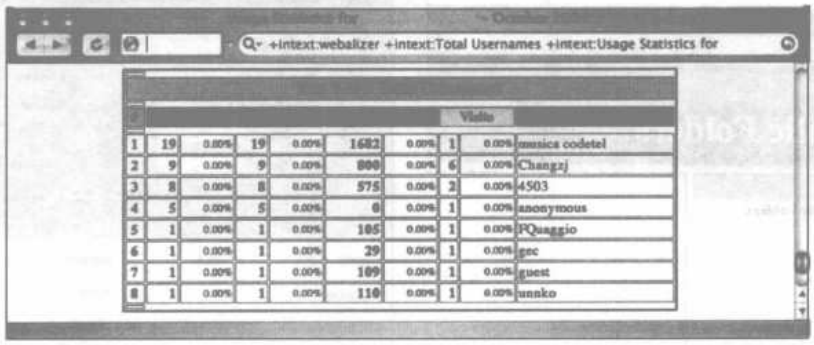


图9-2 Webalizer输出页面列出了Web用户名

Windows注册表保存了各种认证信息，包括用户名和口令。虽然不可能（而且相当少见）在Web中搜索到可用的，导出了的Windows注册表文件，但是在本书写作之时，查询filetype:reg HKEY_CURRENT_USER username还是找到了将近200条结果，这些结果都包含了单词username，甚至有些还包含口令的Windows注册表文件，如图9-3所示。

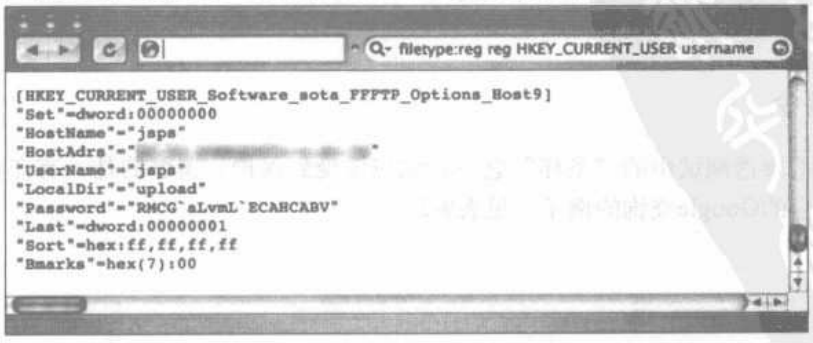


图9-3 普通的Windows注册表文件能够泄露用户名和口令

任何一个聪明的攻击者或者安全人员都会告诉你，很少能够恰好获得你所需要的信息。大多数有价值的发现都需要一点恒心、创造力、智慧和好运气。例如，Microsoft Outlook Web访问入口，就可以使用诸如inurl:root.asp?acs=anon之类的查询来搜索到。这个查询返回的网站不多，但是却有很多运行Microsoft基于Web的邮件入口。不管你是怎样找到运行这种E-mail入口的站点的，但是通常这些站点都会有一个公共的目录（默认表示为“Find Names”），如图9-4所示。

公共目录允许访问一个搜索页面，这个搜索页面可以根据名字查找用户。一般情况下，是不允许用通配符进行搜索的，这意味着用*去搜索并不会像预料中那样返回一个所有用户的列表。输入一个空格进行搜索是一个有趣的主意，因为大部分用户的描述都包含一个空格，但是许多大型目录都会返回“这个查询可能会返回过多的地址！（This query would return too many addresses!）”错误消息。再用上一点创造力，攻击者可能开始搜索一些单独的常见的字母，例如“幸运之轮字母（Wheel of Fortune letters）”，R，S，T，L，N和E。实际上，这些搜索中的任何一个都能够泄露一张类似于图9-5所示那样的用户信息列表。

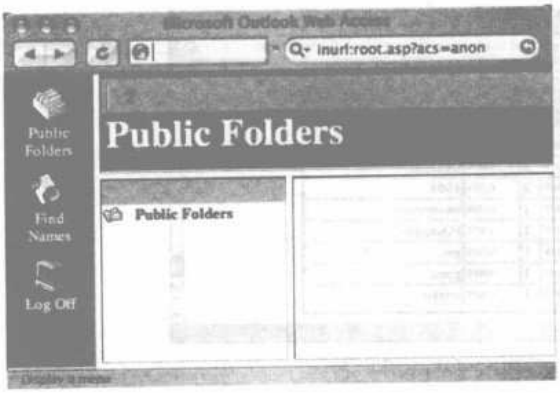


图9-4 Microsoft Outlook Web访问入口具有一个公共目录

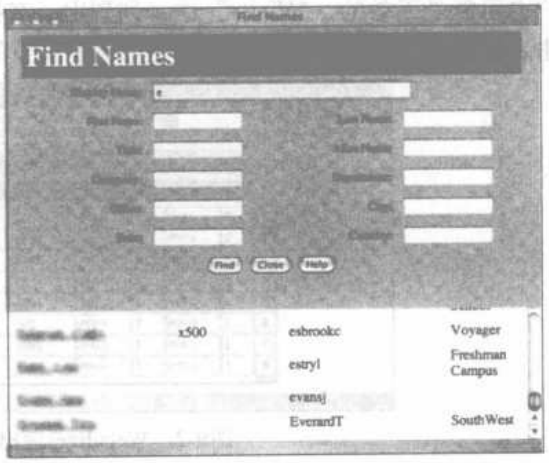


图9-5 用于搜索用户名的公共Outlook目录

一旦返回了一个用户信息列表，攻击者就能够循环利用包含在用户列表中的单词进行搜索，例如搜索单词Vayager、Freshmen或者Campus。然后其结果又可以循环利用，最终会得到一个接近于包含所有用户信息的列表。

9.3 搜索口令

口令数据是渗透测试中的“圣杯”之一，它应该受到保护。遗憾的是，有许多可以用来在网络上搜索口令的Google查询的例子，见表9-2。

表9-2 搜索口令信息的查询

查 询	说 明
filetype:config config intext: appSettings "User ID"	.Net Web Application配置可能包含认证信息
filetype:netrc password	netrc文件可能包含明文口令
intitle: "Index of" passwords modified	"Password" directories
inurl:/db/main.mdb	ASP-Nuke数据文件通常包含口令
filetype:bak inurl: "htaccess|passwd|shadow|htusers"	BAK文件会提及口令和用户名
filetype:log "See `ipsec —copyright"	BARF日志文件会泄露ipsec数据
inurl: "calendarscript/users.txt"	CalenderScript口令
inurl:ccbill filetype:log data	CCBill日志文件可能包含认证
inurl:cgi-bin inurl:calendar.cfg	CGI Calendar（Perl）配置文件会泄露包含程序口令的信息
inurl:chap-secrets -cvs	chap-secrets文件可能会列出用户名和口令
enable password | secret "current configuration" -intext:the	Cisco "secret 5" and "password 7" 口令
intext: "enable secret 5 $"	Cisco可用机密
intext: "enable password 7"	Cisco路由器配置文件
[WFClient] Password= filetype:ica	Citrix WinFrame-Client可能包含登录信息
inurl:passlist.txt	明文口令。不需解密
filetype:cfm "cfapplication name" password	ColdFusion源代码提及了口令（password）
intitle:index.of config.php	Config.php files
inurl:config.php dbuname dbpass	Config.php files
inurl:server.cfg rcon password	计数器强行破解rcon密码
ext:inc "pwd=" "UID="	数据库连接字符串
ext:asa | ext:bak intext:uid intext:pwd - "uid..pwd" database | server | dsn	位于ASA和BAK文件中的数据库证书
filetype:ldb admin	数据库加密文件可能包含证书信息
filetype:properties inurl:db intext: password	db.properties文件包含用户名、加密的口令
filetype:inc dbconn	Dbconn.inc包含站点用来连接到数据库的用户名和口令
filetype:pass pass intext:userid	dbman口令文件
allinurl:auth_user_file.txt	DCForum口令文件
"powered by ducalendar" -site:duware.com	ducalendar数据库可能泄露口令数据
"Powered by Duclassified" -site:duware.com	Duclassified数据库可能泄露口令数据
"powered by duclassmate" -site:duware.com	duclassmate数据库可能泄露口令数据
"Powered by Dudirectory" -site:duware.com	dudirectory数据库可能泄露口令数据
"powered by dudownload" -site:duware.com	dudownload数据库可能泄露口令数据
"Powered by DUpaypal" -site:duware.com	Dupaypal数据库可能泄露口令数据
intitle:dupics inurl:(add.asp | default.asp | view.asp | voting.asp) -site:duware.com	dupics数据库可能泄露口令数据

（续）

查　询	说　明
eggdrop filetype:user user	Eggdrop配置文件
“Powered By Elite Forum Version *.*”	Elite论坛数据库包含证书信息
intitle: “Index of” pwd.db	加密的pwd.db口令
ext:ini eudora.ini	Eudora INI文件可能包含用户名以及加密的口令
inurl:filezilla.xml -cvs	filezilla.xml包含口令数据
filetype:ini inurl:flashFXP.ini	FlashFXP配置文件可能包含FTP口令
filetype:dat inurl:Sites.dat	FlashFXP FTP口令
inurl: “Sites.dat” + “PASS=”	FlashFXP Sites.dat服务器配置文件
ext:pwd inurl:(service | authors | administrators | users) “# -FrontPage-”	与Frontpage敏感证书相关文件
filetype:url +inurl: “ftp://” +inurl: “@”	某些FTP书签包含明文登录名和口令
intitle:index.of passwd passwd.bak	通用PASSWD文件
inurl:zebra.conf intext:password -sample -test -tutorial -download	GNU Zebra可用口令（明文或者加密的口令）
intext: “powered by EZGuestbook”	HTMLJunction EZGuestbook数据库泄露了证书数据
intitle: “Index of” “.htpasswd” htpasswd.bak	htpasswd口令文件
intitle: “Index of” “.htpasswd” “htgroup” -intitle: “dist” -apache -htpasswd.c	htpasswd口令文件
filetype:htpasswd htpasswd	htpasswd口令文件
“http://*:*@www” bob:bob	HTTP Web认证信息
“liveice configuration file” ext:cfg -site:sourceforge.net	Icecast liveice.cfg文件可能包含口令
“sets mode: +k”	IRC信道密钥
signin filetype:url	Javascript用户有效机制可能包含明文用户名和口令
LeapFTP intitle: “index.of./” sites.ini modified	LeapFTP客户配置文件可能泄露认证信息
inurl:lilo.conf filetype:conf password -tatercounter2000 -bootpwd -man	LILO根口令
“Powered by Link Department”	链接管理脚本包含加密的管理口令和会话数据
“your password is” filetype:log	包含词组（你的口令）的日志文件
“admin account info” filetype:log	包含管理服务器账号信息的日志
intitle:index.of master.passwd	master.passwd文件
allinurl: admin mdb	Microsoft Access “管理员（admin）”数据库
filetype:mdb inurl:users.mdb	Microsoft Access “用户数据库”
filetype:xls username password email	Microsoft Excel电子数据表包含用户名、口令和E-mail的单词
intitle:index.of administrators.pwd	Microsoft Frontpage管理用户名和口令
filetype:pwd service	Microsoft Frontpage服务信息
inurl:perform.ini filetype:ini	mIRC IRC口令
inurl:perform filetype:ini	mIRC潜在的连接数据

（续）

查 询	说 明
filetype:cfg mrtg “target[*]” -sample -cvs -example	Mrtg.cfg SNMP配置文件可能会泄露公共和私有的社区字符串
intitle: “index of” intext:connect.inc	MySQL数据库连接信息
intitle: “Index of” .mysql_history	MySQL历史文件
intitle: “index of” intext:globals.inc	MySQL用户/口令信息
“Your password is * Remember this for later use”	NickServ注册口令
filetype:conf oekakibbs	Oekakibss配置文件可能会泄露口令
filetype:conf slapd.conf	OpenLDAP slapd.conf文件包含包括根口令的配置数据
inurl: “slapd.conf” intext: “credentials” -manpage -“Manual Page” -man: -sample	OpenLDAP slapd.conf文件包含包括根口令的配置数据
filetype:dat wand.dat	Opera Web浏览器“魔法棒”存储了证书
inurl:pap-secrets -cvs	pap-secrets文件可能会列出用户名和口令
filetype:dat inurl:pass.dat	Pass.dat文件可能泄露口令
index.of passlist	Passlist口令文件
filetype:dat “password.dat”	Password.dat文件可以包含明文用户名和口令
filetype:log inurl: “password.log”	Password.log文件可能包含明文用户名和口令
filetype:pem intext:private	PEM私人密钥文件
intitle:index.of people.lst	people.lst文件
intitle:index.of intext: “secring.skr” | “secring.pgp” | “secring.bak”	PGP加密密钥环
inurl:secring ext:skr | ext:pgp | ext:bak	PGP加密密钥环
filetype:inc mysql_connect OR mysql_pconnect	PHP .inc文件包含认证信息
filetype:inc intext:mysql_connect	PHP .inc文件包含用户名和口令
ext:php intext: “$dbms” “$dbhost” “$dbuser” “$dbpasswd” “$table_prefix” “phpbb_installed”	phpBB mySQL连接信息
intitle: “phpinfo()” + “mysql. default_password” + “Zend Scripting Language Engine”	phpinfo文件可能包含默认的mysql密码
inurl:nuke filetype:sql	PHP-Nuke或Postnuke数据库转储可能包含认证数据
“parent directory” +proftpdpasswd	ProFTPd用户名以及口令会受Web服务器信息备份的挠乱
filetype:conf inurl:psybnc.conf “USER.PASS=”	psyBNC配置文件可能包含认证信息
intitle:rapidshare intext:login	Rapidshare登录口令
inurl: “editor/list.asp” | inurl: “database_editor.asp” | inurl: “login.asa” “are set”	Results Database Editor用户名/口令
ext:yml database inurl:config	Ruby on Rails数据链接文件
ext:ini Version=4.0.0.4 password	servU FTP Daemon INI可能包含用户名和口令
filetype:ini ServUDaemon	servU FTP Daemon INI文件可能包含设置、会话以及认证数据

（续）

查　询	说　明
filetype:ini inurl:“serv-u.ini”	Serv-U INI文件可能包含用户名和口令数据
intitle:“Index of” sc_serv.conf sc_serv content	Shoutcast sc_serv.conf文件通常包含明文口令
intitle:“Index of” spwd.db passwd -pam.conf	spwd.db口令文件
filetype:sql “insert into” (pass|passwd|password)	SQL转储包含明文口令或加密口令
filetype:sql (“passwd values” | “password values” | “pass values”)	SQL文件口令引用
filetype:sql (“values * MD5” | “values * password” | “values * encrypt”)	SQL文件可能包含加密的口令
filetype:sql +“IDENTIFIED BY” -cvs	提及认证信息的SQL文件
filetype:sql password	提及认证信息的SQL文件
filetype:reg reg HKEY_CURRENT_USER SSHHOSTKEYS	存储在Windows注册表的SSH主机密钥
inurl:“GRC.DAT” intext:“password”	Symantec Norton Anti-Virus Corporate Edition数据文件包含加密口令的数据文件
filetype:inf sysprep	Sysprep.inf文件包含包括管理口令、IP地址和产品ID在内的Windows信息的所有信息
server-dbs “intitle:index of”	teamspeak服务器管理员文件
filetype:ini wcx_ftp	Total commander FTP口令
intitle:index.of trillian.ini	Trillian INI文件包含口令
ext:txt inurl:unattend.txt	unattend.txt文件包含包括管理口令、IP地址和产品ID在内的Windows信息的所有信息
index.of.etc	Unix /etc目录
intitle:“Index of..etc” passwd	Unix /etc/passwd文件
intitle:Index.of etc shadow	UNIX /etc/shadow口令文件
ext:passwd -intext:the -sample -example	多个口令
filetype:bak createobject sa	VBScript数据库连接备份
inurl:ventrilo_srv.ini adminpassword	多个服务器的ventrilo口令
filetype:reg reg +intext: WINVNC3	vnc口令
!Host=*.* intext:enc_UserPassword=* ext:pcf	VPN配置文件通常包含认证数据
inurl:vtund.conf intext:pass -cvs	vtund配置文件可以包含用户名和口令
filetype:mdb wwforum	Web Wiz论坛数据库包含认证信息
intext:“powered by Web Wiz Journal”	Web Wiz Journal ASP Blog数据库包含管理信息
“AutoCreate=TRUE password=*”	Website Access Analyzer口令
filetype:pwl pwl	Windows口令列表文件
filetype:reg reg +intext: “defaultusername” +intext: “defaultpassword”	Windows注册密钥会泄露口令
filetype:ini ws_ftp pwd	WS_FTP.ini文件包含弱加密口令
“index of/” “ws_ftp.ini” “parent directory”	WS_FTP.ini文件包含弱加密口令

（续）

查　询	说　明
inurl: “wvdial.conf” intext: “password”	wvdial.conf可能包含电话号码、用户名和口令
inurl:/wwwboard	WWWBoard “passwd.txt” 认证配置文件
wwwboard WebAdmin inurl: passwd.txt wwwboard|webadmin	WWWBoard口令文件
“login: *” “password= *” filetype:xls	xls文件包含登录名和口令
inurl:/yabb/Members/Admin.dat	YaBB论坛管理员口令

大多数情况下，在Web上发现的口令大都是以某种方式加密了的或者是编码了的。这些密码大都可以用一种口令破译器来生成明文口令以用于攻击，例如John the Ripper，它可以从www.openwall.com/john获得。图9-6是搜索ext:pwd inurl:_vit_pvt inurl:(Service | authors | administrators)的结果，这个查询是对一些常见的Microsoft FrontPage支持的文件进行组合搜索。

图9-6　加密了的或者编码了的口令

导出的Windows注册文件通常也包含加密的或者编码的口令。如果一个用户把Windows注册表导出到一个文件，并且Google后来也抓取到了这个文件，那么一个类似于filetype:reg intext: “internet account manager” 这样的查询就能够得到有趣的包含口令数据的键，如图9-7所示。

要注意的是，可用的、导出的注册表文件并不常见，然而攻击者仅仅因为一个网站具有异常的不安全的文件就攻击它却是很常见的。也可以利用Google查询来挖掘明文口令。不需要使用口令破译工具就可以利用这些口令。在这些极端的情况下，唯一的挑战就是判定相应的用户名以及可以使用这些口令的主机。如图9-8所示，某些查询能够搜索到下面所有信息：用户名、明文口令和使用它们认证的主机！

没有任何神奇的用来查找口令的查询，但是要知道，在一次评估过程中，最简单的直接针对某个站点的查询往往会产生让人惊奇的结果，这正如我们在“排行前10位的搜索”的章节中

讨论的那样。例如，查询“Your password” forgot能够搜索到提供恢复忘记口令机制的页面。这类查询所得到的信息可以用来设计任意数量的针对口令的攻击。同时，有效的社会工程也是一种针对“忘记”口令的极好的非技术类解决方案。

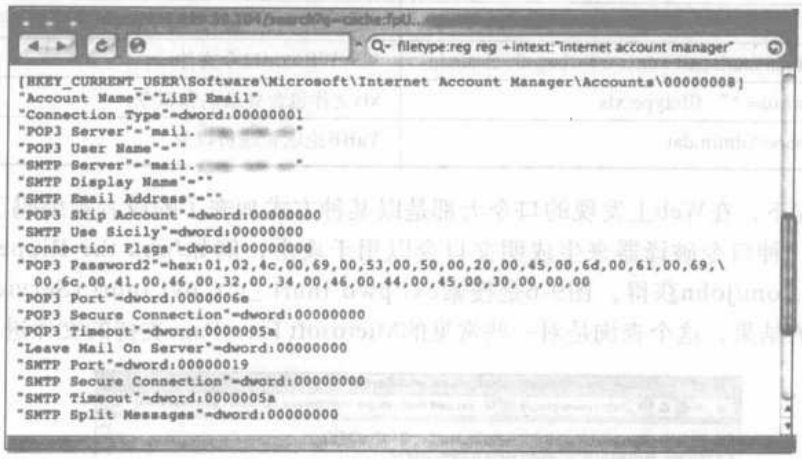


图9-7 特殊的Windows注册表条目会泄露口令

图9-8 圣杯：用户名，明文口令和主机名！

另外一种通用的针对口令信息的搜索是intext:(password | passcode | pass) intext:(username | userid | user)，它是把常见的描述口令和用户ID的单词组合在一起进行搜索。这个查询返回大量的结果，但是大部分靠前的结果引用的都是列出了忘记口令信息的页面，包括链接或者联系信息。使用http://translate.google.com/translate_t提供的Google翻译功能，我们也可以创建多语言的口令搜索。表9-3给出了单词password常见的一些翻译。注意，术语username和userid在大多数语言中都还是会分别翻译为username和userid。

表9-3 英语单词password的翻译

语言	单词	译文
德语	password	Kennwort
西班牙语	password	contraseña
法语	password	mot de passe
意大利语	password	parola d'accesso
葡萄牙语	password	senha
荷兰语	password	Paswoord

9.4 搜索信用卡账号和社保号码等

许多人都听过关于Web黑客盗取客户信用卡信息的新闻故事。随着互联网上不可靠的零售商越来越多，出现如此多的信用卡欺骗现象也就没有什么可奇怪的了。并不是只有这些小规模的零售商会受到黑客的攻击。这些年，成百上千个大型企业的财政数据库也受到了攻击，有些受害公司还是技术比较好并且也非常注意防止攻击的公司。可能让你感到更为惊讶的是要在互联网上发现可用的信用卡账号并不困难，这要多感谢像Google这样的搜索引擎。从信用卡信息到银行数据或者甚为敏感的政府机密文档都可以在网络中找到。先来看一下如图9-9所示的网页（做了大幅的编辑）。

图9-9 Google保存了一堆堆之前窃取的个人数据

这个用Google找到的文档不仅列出了大量的信用卡账号（包括过期日期和信用卡验证号），还列出了所有者的姓名、地址和电话号码。这个特殊的文档还包含电话卡账号。注意图9-9中右手边的滚动栏，它表明图中所显示的只不过是这个庞大的文档的一小部分，而许多这类文档都是如此庞大的。大多数情况下，包含这些账号的页面都不是从在线零售商或者电子商务网站“泄露”出去的，而很有可能是网络钓鱼(phishing)欺骗的结果。网络钓鱼欺骗是通过电话或者E-mail诱骗用户的个人信息。一些网站，包括MillerSmiles.co.uk揭露了这些欺诈手段。图9-10给出了一种流行的利用eBay进行钓鱼欺骗的屏幕快照，这种欺骗手段鼓动用户更新他们的eBay配置信息。

一旦用户填了这个表格，那么其中所有的信息都会通过E-mail发送给攻击者，他们能够使用这些信息做任何事情。有时，这些数据存储在攻击者使用的Web服务器中。我曾看到过联机“钓鱼调查员”张贴出的链接到钓鱼者被盗个人数据的缓存的报告。当搜索引擎抓取到这些链接时，即便是业余的Google黑客也可以使用所有的这些个人数据了。

工具和陷阱

捕捉网络在线诈骗者

某些时候，可以利用Google来帮助捉到那些“坏家伙”。钓鱼欺骗之所以有效是因为那

些伪造的页面看起来很像官方的页面。想要创建一个像官方界面那样的页面，他们必须有一些可以参考的样例，这意味着他们必定浏览过一些真实的公司网站。如果钓鱼欺骗是用一些公司的现有页面中的文字来创建的，那么你就可以重点关注伪造页面中的一些特定的词组，并创建一些设计用来搜索包含原始内容的服务器。一旦查找到包含窃取的文字的服务器，那么就可以从这些服务器的日志文件中提取相应的连接数据。如果欺骗者浏览了每个公司的网页，只要他收集了一点真实的文字，他的IP都会出现于每个日志文件中。SensePost（www.sensepost.com）的审计人员就成功地使用这种技术捉到了网络欺骗“艺术家”。遗憾的是，如果欺骗者只使用了一个公司的页面副本，那么要完成捕捉这个任务就相当困难了。

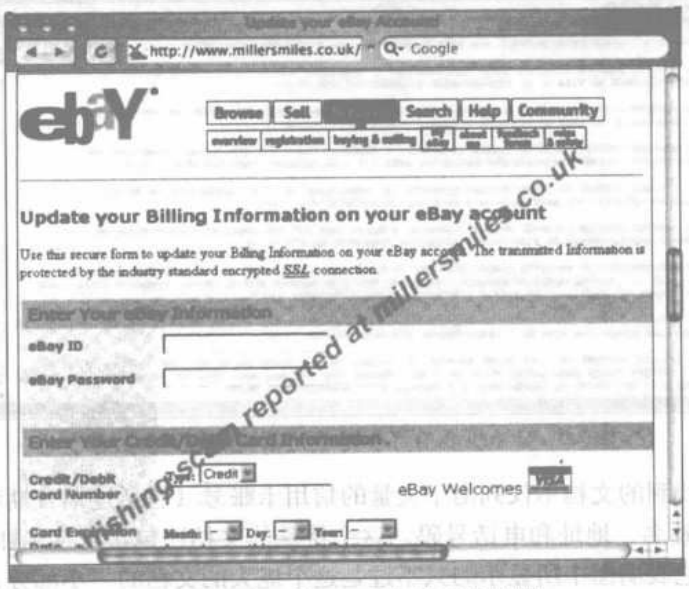


图9-10 一种eBay钓鱼欺骗的快照

9.4.1 社保号码

社保号码（Social Security Number，SSN）和其他的敏感数据也可以很容易地通过Google查找到，所用的技术和搜索信用卡账号的技术相同。SSN出现在网络上的原因有很多，比如，我们都知道教育机构会把SSN用作学生的身份标识（ID），然后会把“学生ID”和年级数据一起发到某个公开的网站上。一个聪明的攻击者只用一个SSN就能够做许多事情，但是，在许多情况下，这也有助于获得与该SSN相关的姓名。除此之外，这些教育机构还会通过列出了学生的姓名、年级和SSN的Excel电子表格来暴露这种信息，尽管事实上，学生的ID号码通常是用来帮助保护学生的隐私的！虽然我们从来都没有揭露如何查找SSN的信息，但是一些媒体渠道还是不负责任地在网上发布了其中的细节。虽然这要怪那些泄露了这种信息的网站，但我们的意见还是不应该把注意力放在如何精确地查找这种信息上。

9.4.2 个人财务数据

在某些情况下，公布个人消息的责任在于钓鱼欺骗；在其他一些情况下，侵犯个人隐私就要归咎于那些攻击网上零售商的黑客了。可悲的是，有许多的例子可以说明缺少保护自己隐私的责任则要归咎于他自己了。本小节提到的个人财务信息就是这样一个例子。随着当今社会个人电脑数量的快速增长，差不多有上百种个人财务程序供用户选择使用。许多这种程序都会创建带有特定文件扩展名的数据文件，而这些文件是能够用Google搜索到的。很难想象为什么会有人把个人财务信息放到公开的网站上（然后被Google抓取），但是这种情况却发生过很多，例如可以通过搜索Quicken和Microsoft Money生成的程序文件得到的结果数来判断这一点。虽然在这里提供能够找出个人财务数据的查询有些不负责任，但重要的是要知道攻击者是可以发现这类数据的。最后，表9-4列出了各种财务、会计和纳税软件生成的文件扩展名。

表9-4 各种财务程序的文件扩展名

文件扩展名	说 明
afm	Abassis Finance Manager
ab4	会计与理财（Accounting and Business）文件
mmw	AceMoney文件
lqd	AmeriCalc Mutual Fund税务报表（Tax Report）
et2	电子纳税申报单安全文件（Electronic Tax Return Security File）（澳大利亚）
tax	Intuit TurboTax纳税申报单
t98-t04	Kiplinger减税（Tax Cut）文件（扩展名是基于用两位数字表示的年份）
mny	Microsoft Money 2004 Money数据文件
mbf	Microsoft Money备份文件
inv	MSN Money Investor文件
ptdb	Peachtree 会计数据库（Accounting Database）
qbb	QuickBooks备份文件会泄露财务数据
qdf	Quicken个人财务数据
soa	Sage MAS 90会计软件
sdb	Simply Accounting会计软件
stx	Simply Tax Form
tmd	时间与计费跟踪（Time and Expense Tracking）
tls	Timeless Time & Expense时间与计费软件
fec	美国联邦竞选费用申请书（U.S. Federal Campaign Expense Submission）
wow	Wings会计文件

9.5 搜索其他有利可图的信息

如你所见，Google可以用来搜索各种敏感信息。在这一节中，我们来看一看一些Google能找到的但是很难归类的数据。这些数据包括地址簿、聊天日志文件以及网络漏洞报告等，这些网上的敏感数据都是很有价值的。表9-5列出了一些可以用来发掘各种敏感数据的查询。

表9-5 搜索各种敏感信息的查询

查 询	说 明
intext:"Session Start * * * *:*:* *" filetype:log	AIM和IRC日志文件
filetype:blt blt +intext:screenname	AIM好友列表
buddylist.blt	AIM好友列表
intitle:index.of cgiirc.config	CGIIRC（一种基于Web的IRC客户端）的配置文件，包含IRC服务器和用户证书
inurl:cgiirc.config	CGIIRC（一种基于Web的IRC客户端）的配置文件，包含IRC服务器和用户证书
"Index of" / "chat/logs"	聊天日志
intitle:"Index Of" cookies.txt "size"	cookies.txt泄露用户信息
"phone * * *" "address *" "e-mail" intitle:"curriculum vitae"	Curriculum vitae（简历）泄露姓名和地址信息
ext:ini intext:env.ini	环境数据
intitle:index.of inbox	邮件箱文件
"Running in Child mode"	Gnutella客户端数据和统计信息
":8080" ":3128" ":80" filetype:txt	HTTP代理列表
intitle:"Index of" dbconvert.exe chats	ICQ聊天日志
"sets mode: +p"	IRC专用频道信息
"sets mode: +s"	IRC秘密频道信息
"Host Vulnerability Summary Report"	ISS漏洞扫描器报告，泄露主机和网络的可能漏洞
"Network Vulnerability Assessment Report"	ISS漏洞扫描器报告，泄露主机和网络的可能漏洞
filetype:pot inurl:john.pot	John the Ripper口令破译器结果
intitle:"Index Of" -inurl:maillog maillog size	Maillog文件泄露e-mail流量信息
ext:mdb inurl:*.mdb inurl:fpdb shop.mdb	Microsoft FrontPage数据库文件夹
filetype:xls inurl:contact	包括联系信息的Microsoft Excel表格
intitle:index.of haccess.ctl	使用htaccess的Microsoft FrontPage会显示出Web认证信息
ext:log "Software: Microsoft Internet Information Services *.*"	Microsoft Internet Information Servic-es（IIS）日志文件
filetype:pst inurl:"outlook.pst"	Microsoft Outlook e-mail和日历备份文件
intitle:index.of mt-db-pass.cgi	Movable Type默认文件
filetype:ctt ctt messenger	MSN Messenger联系人列表
"This file was generated by Nessus"	Nessus漏洞扫描器报告，泄露主机和网络的可能漏洞
inurl:"newsletter/admin/"	Newsletter管理信息
inurl:"newsletter/admin/" intitle:"newsletter admin"	Newsletter管理信息
filetype:eml eml intext:"Subject" +Froms	Outlook Express E-mail文件
intitle:index.of inbox dbx	Outlook Express Mailbox文件
intitle:index.of inbox dbx	Outlook Express Mailbox文件
filetype:mbx mbx intext:Subject	Outlook v1-v4或者Eudora邮件箱文件
inurl:/public/?Cmd=contents	Outlook Web Access公共文件夹
filetype:pdb pdb backup (Pilot | Pluckerdb)	Palm Pilot Hotsync数据库文件
"This is a Shareaza Node"	Shareaza客户端数据和统计信息
inurl:/_layouts/settings	Sharepoint配置信息
inurl:ssl.conf filetype:conf	SSL配置文件，泄露各种配置信息
site:edu admin grades	学生年级信息
intitle:index.of mystuff.xml	Trillian用户Web链接
inurl:forward filetype:forward -cvs	UNIX mail forward文件泄露e-mail地址
intitle:index.of dead.letter	UNIX未完成的e-mail
filetype:conf inurl:unrealircd.conf -cvs -gentoo	UnrealIRCd配置文件泄露配置信息
filetype:bkf bkf	Windows XP/2000备份文件

这些信息中有些是没有太大危险的，比如，用查询filetype:ctt messenger找到的MSN Messenger联系人列表文件，或者用查询filetype:blt blt +intext:screenname，如图9-11所示。

图9-11 AIM好友列表泄露个人联系列表

这个屏幕快照给出了“好友”列表，或者是加入到了他或她的AIM客户端中的熟人。攻击者经常在社会工程攻击中使用像这样的个人信息，试图让目标相信他们是朋友或者认识的人。这种行为就等同于从目标手中骗取一份Rolodex（一种名片簿）或者地址簿。对于一个老练的攻击者而言，像这样的信息就足以实施一次成功的攻击。但是，在一些情况下，用Google查询找到的数据会泄露与安全相关的敏感信息，即便是最初级的攻击者都能够使用这些信息来攻陷一个系统。

例如，来看一下Nessus安全扫描工具的输出。Nessus可以在www.nessus.org上获得。这个很棒的开源工具能够对目标执行一系列的安全测试，并报告各种可能的漏洞。由Nessus生成的报告可以用来帮助引导系统管理员加强受影响系统的安全性。攻击者也可以利用这种报告搜索目标可能具有的漏洞。如图9-12所示，使用Google查询“This file was generated by Nessus”，攻击者就能够找到由Nessus生成的报告。这个报告列出了每个被测试的机器的IP地址、开放的端口和所有检测到的漏洞。

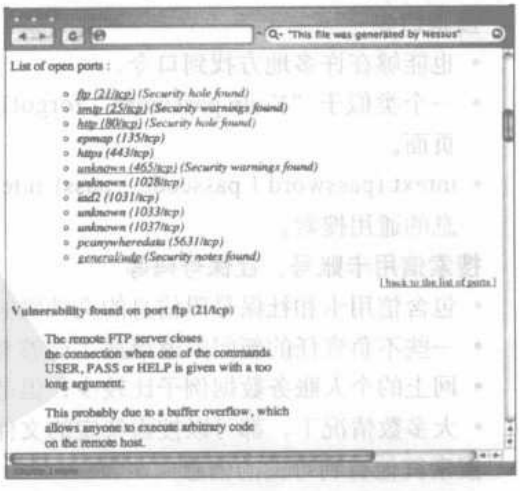


图9-12 在网上找到的Nessus漏洞报告

大多数情况下，用这种方法找到的报告都是一些示例或者测试报告，但是有些报告是可用的，而且被测试的系统确实可以像报告中列出的那样进行漏洞利用。只能希望报告的系统是一些蜜罐，即主要用来引诱黑客并且追踪黑客行为的机器。在下一章中，我们将更为详细地讨论“文档研磨（document-

grinding)”技术，这种技术也有利于挖掘这类信息。这一章关注的是基于文件名来搜索信息，而下一章关注的是文档的实际内容，而不是其名字。

9.6 总结

不要犯任何错误，因为在网上存在各种敏感的数据，而且Google能够找到这些数据。除非你能给出恰当的查询，否则可搜索的信息几乎没有任何限制。用户名、口令、信用卡账号、社保号码和个人财务数据这些信息都在可搜索的范围内。如果作为一名“黑暗艺术”的爱好者，那么你能够看到其他人的愚蠢，而如果作为一名针对这种危险的信息泄露形式而加强客户的网站安全性的专业人员，你就可能被如此大范围的防卫任务压垮。

听起来好像很可笑，一种牢固的，强制实施的安全策略是不错的防止敏感数据泄露的方法。但是如果用户明白信息泄露带来的风险，清楚违反策略会受到的惩罚，那么他们就更愿意为此合作了。

同时，这当然也不妨碍搞清楚敌人在攻击Web服务器时采用的策略。在阅读本书时，有一件事你应该能够明白，即任何一名攻击者都有大量的文件需要分析。一种防止危险的Web信息泄露的方法是通过禁止对未知文件类型的请求。不管你的Web服务器在正常情况下是否只解析CFM、ASP、PHP或HTML文件，管理Web服务器应该解析的比关注不应该解析的要容易得多。调整你的服务器或者边界保护设备只允许特定的内容或文件类型。

9.7 快速查找解决方案

搜索用户名

- 能够在许多地方找到用户名。
- 某些时候，可能需要挖掘文档或E-mail目录。
- 一个像“your username is”这样简单的查询都能够有效地搜索用户名。

搜索口令

- 也能够在许多地方找到口令。
- 一个类似于“Your password”forgot这样简单的查询能够找到提供忘记口令恢复机制的页面。
- intext:(password | passcode | pass) intext:(username | userid | user)是另外一种针对口令信息的通用搜索。

搜索信用卡账号、社保号码等

- 包含信用卡和社保号码信息的文档确实存在而且也是比较多的。
- 一些不负责任的新闻渠道泄露了能够查找这些信息的功能查询。
- 网上的个人账务数据例子比较少，但是表现形式却大不相同。
- 大多数情况下，都可以搜索特定的文件扩展名。

搜索其他有利可图的信息

- 从地址簿和聊天日志文件到网络漏洞报告，网上的敏感数据都是有用的。

9.8 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所涉及概念的理解程度，也可以帮助你在现实生活中运用这些概念解决实现问题。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：我比较关心钓鱼欺骗的方法。有资源能帮助我认清相应的风险和了解一些保护措施吗？

答：有一个相当不错的专门针对钓鱼欺骗的网站，www.antiphishing.org。你也可以阅读由Next Generation Security Software Ltd.编写的文章，The Phishing Guide: Understanding and Preventing Phishing Attacks，www.ngssoftware.com/papers/NISR-WP-Phishing.pdf。

问：为什么你不给出更多关于查找像信用卡账号和社保号码这样的信息的细节？

答：说真的，不管是作者还是出版商都不愿意负这种个人责任，即鼓励可能的非法行为。大部分对这类信息感兴趣的人都是用作非法用途的。如果你想在网上搜索自己的个人信息，那么只要简单地在Google中输入你的信息即可。如果能得到一些结果，那么你就要注意了。当然，键入你所有的个人信息，例如信用卡账号以及社保号码并不是一个好主意，因为一个非法闯入者可以很轻易地捕获这一信息。最好是只输入其中的一部分信息。要想做得更好的话，就是在尝试保护自己的时候不要将自己全方位开放。

问：许多口令只是授予意义不大的服务的访问权限。我为什么要担心没有用的服务的口令泄露到网上呢？

答：研究表明大多数人通常会选择最容易的途径来完成任务。从安全的角度来看，这意味着许多人会在不同服务器上的不同应用程序之间共用同一个口令（或者口令线索）。一个泄露的口令就能够提供关于用于其他系统上的口令的线索。大多数策略都禁止这种口令共享，但是这种限制通常很难执行。

第10章 Hacking Google服务

10.1 AJAX Search API

AJAX Search API是AJAX前端的主要的Google服务。该项服务有意取代旧的SOAP搜索服务，而SOAP搜索服务的支持在不久前已经被废止了。AJAX Search API被认为是比SOAP服务更强大的一个服务，并且更容易使用。该服务的主要目标是使外部网能够寄存Google供给的技术，这些技术提供了位于寄存在Web站点内、外的搜索工具，并且可以是视频剪辑、地图、博客、自定义搜索引擎等等中的一个。

该服务的默认界面如图10-1所示。

图10-1 Google AJAX Search API对话框

图10-1中的搜索对话框被分为几个部分，每个部分都代表一个搜索类别：Local、Web、Video、Blog、News和Book。因为我们可以执行非常有趣的查询并且可以通过整个Google平台得到即时的反馈，所以尤其是将所有的结果放在同一个地方非常有用。这正是Search API最耀眼的功能。让我们来尝试进行一个查询：firefox，如图10-2所示。

简单地访问http://www.google.com/uds/samples/apidocs/helloworld.html演示应用程序并且键入该查询。

注意，AJAS API结果设置也包含带有最多相关结果的图像（Image）搜索区域。在下面的部分我们将详细地介绍AJAX API Search服务。

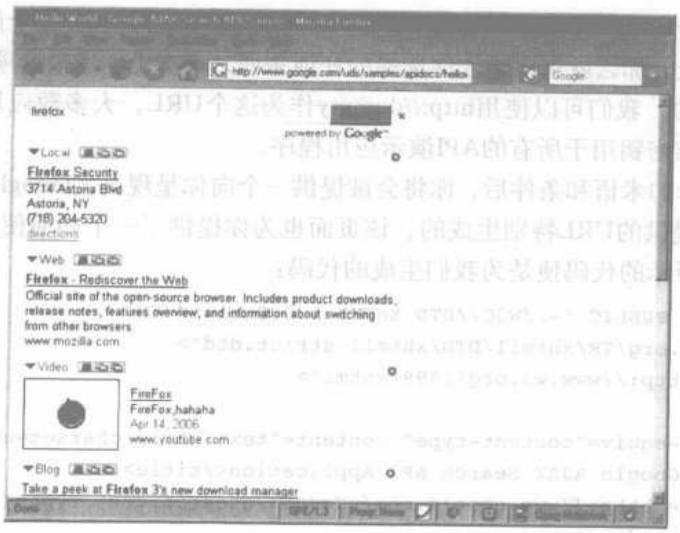


图10-2 搜索“firefox”的AJAX搜索

10.1.1 嵌入式Google AJAX Search API

Google AJAX Search API设计为嵌入到外部页内。这使得服务相当有用，因为我们可以指导自定义界面来获得更好的对Google底层结构的访问。为了着手启用Google AJAX Search API，你需要理解JavaScript和AJAX编程以及你要自行生成的API密钥。假定基本上了解了AJAX，我们将集中关注服务自身的有趣的信息。

为了生成API密钥，可以通过访问http://code.google.com/apis/ajaxsearch网址来访问AJAX Search API主面。在单击Start using the Google AJAX Search API之后，你可以看到与图10-3所示的页面相似的页面。

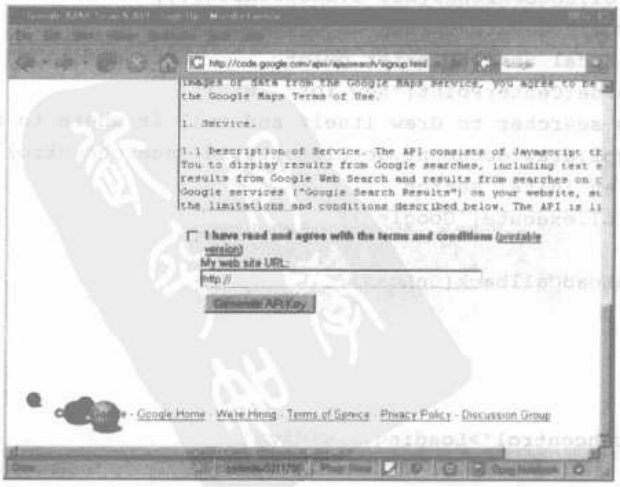


图10-3 AJAX Search API密钥生成机

你需要提供一个可以从中获取服务的URL。如果你打算把这个取自某个简单的页面的应用程序放到你的桌面上，那么就可以输入你选择的任何东西。事实上，这个选项看起来很不相关。为了演示程序的目的，我们可以使用http://dummy作为这个URL。大多数应用程序都会使用互联网Google密钥，该密钥用于所有的API演示应用程序。

在接受了Google的术语和条件后，你将会被提供一个向你呈现真实Google API密钥的页面，而该密钥是为事先提供的URL特别生成的。该页面也为你提供了一个如何使用AJAX Search输入框的示例。下面所示的代码便是为我们生成的代码：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>My Google AJAX Search API Application</title>
    <link href="http://www.google.com/uds/css/gsearch.css" type="text/css"
 rel="stylesheet"/>
    <script
src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0&amp;key=ABQIAAAAsFym1Ew5o48
zXESOPhV4ExSFOvRczLyAyj57qAvViVrKq19E6hSZhJSVQBi2HRSzsW1XyZzxdffdfQ"
type="text/javascript"></script>
    <script language="Javascript" type="text/javascript">
    //<![CDATA[
    function OnLoad() {
      // Create a search control
      var searchControl = new GSearchControl();
      // Add in a full set of searchers
      var localSearch = new GlocalSearch();
      searchControl.addSearcher(localSearch);
      searchControl.addSearcher(new GwebSearch());
      searchControl.addSearcher(new GvideoSearch());
      searchControl.addSearcher(new GblogSearch());
      // Set the Local Search center point
      localSearch.setCenterPoint("New York, NY");
      // Tell the searcher to draw itself and tell it where to attach
      searchControl.draw(document.getElementById("searchcontrol"));
      // Execute an inital search
      searchControl.execute("Google");
    }
    GSearch.setOnLoadCallback(OnLoad);
    //]]>
    </script>
  </head>
  <body>
    <div id="searchcontrol">Loading...</div>
  </body>
</html>
```

复制该代码并将它粘贴到一个新文件中，例如名为test.html的文件。接着，在浏览器中打开该文件。你将看到一个类似于如图10-4所示的页面。

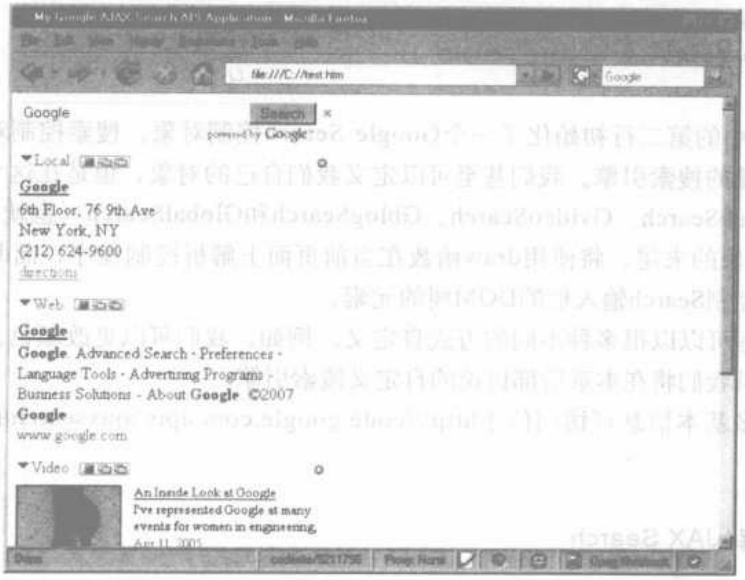


图10-4 测试AJAX Search页面

让我们来回顾一下到目前为止我们所做的工作。生成的HTML代码泄露了API的一些基本特征。首先，代码会加载AJAX Search API默认样式表（CSS），接着会加载JavaScript脚本引用：

```
<script
src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0&amp;key=ABQIAAAAsFym1Ew5o48
zXESOPhV4ExSFOvRczLyAyj57qAvViVrKq19E6hSZhJSVQBi2HRSzsW1XyZzxdffdfQ"
type="text/javascript"></script>
```

该脚本会加载几个JavaScript类，这些类被用作访问API的更便捷的方法。随后你将了解由于我们可以直接访问API，所以我们并不真正需要它们（即，未加处理的访问）。

接下来，开始定义另一个脚本块，该脚本块会初始化环境并配置AJAX Search控制文本框。这可以在OnLoad函数中执行，该函数会在Google加载完所有被要求用来解析图形环境的函数时调用。

```
function OnLoad() {
    // Create a search control
    var searchControl = new GSearchControl();
    // Add in a full set of searchers
    var localSearch = new GlocalSearch();
    searchControl.addSearcher(localSearch);
    searchControl.addSearcher(new GwebSearch());
    searchControl.addSearcher(new GvideoSearch());
    searchControl.addSearcher(new GblogSearch());
    // Set the Local Search center point
```

```
        localSearch.setCenterPoint("New York, NY");
        // Tell the searcher to draw itself and tell it where to attach
        searchControl.draw(document.getElementById("searchcontrol"));
        // Execute an inital search
        searchControl.execute("Google");
}
```

OnLoad函数中的第二行初始化了一个Google Search控制对象。搜索控制对象可以引用我们所需的任意数量的搜索引擎。我们甚至可以定义我们自己的对象，但是在这个实例中，我们将设置默认的GwebSearch、GvideoSearch、GblogSearch和GlobalSearch（也就是Google Local Search）。在程序块的末尾，将使用draw函数在当前页面上解析控制程序，而draw函数则会把参数看作是来自控制Search输入框的DOM树的元素。

该输入文本框可以以很多种不同的方式自定义。例如，我们可以更改颜色、重新排列搜索区域，甚至是提供我们将在本章后部讨论的自定义搜索引擎。

要想得到更多基本信息可访问位于http://code.google.com/apis/ajaxsearch的Google AJAX Search API文档。

10.1.2 深入了解AJAX Search

现在，我们已经了解了如何嵌入AJAX Search输入框，已经到了提出一些更有趣的事的时候了。你可能已经注意到，AJAX Search表格是一个很好的试验该服务的地方，但是它还没有到达那些熟练的黑客通常的工作水平。正是因为这个原因，我们还需要更深入地了解AJAX Search API，并且发掘该服务的更多有趣的特征。下一步，我们将要使用HTTP请求监听程序。我们将要使用Firefox作为开发的基本工具。

这里有几个先决条件。所有条件都需要你先具备Firefox，Firefox可以从www.mozilla.com/firefox上下载。我们还要使用人们所熟知的称为“LiveHTTP Headers”的Firefox扩展程序，它们可以从https://addons.mozilla.org/enUS/firefox/addon/3829上下载。安装完该扩展程序后，重新启动Firefox。

LiveHTTPHeaders扩展程序允许我们分析并且回复HTTP请求。如果你需要监视流量，你可以通过选择View | Sidebar | LiveHTTPHeaders菜单命令在你的浏览器工具栏中简单地打开扩展程序的窗口。另外，如果需要使用请求回复功能，你可能需要选择Tools | LiveHTTPHeaders菜单命令在一个单独的窗口中打开它，如图10-5所示。

诸如LiveHTTPHeaders扩展程序之类的流量监视工具对于Web应用程序安全测试人员而言是必不可少的。这些工具泄露了后台发生了什么，并且为反汇编和重汇编生成的请求提供了多种功能，很容易便揭露了基本应用程序的漏洞，并且洞察到测试过的应用程序内部的工作。

一旦所有环境准备妥当，我们就可以攻入到AJAX搜索逻辑中。这样做是为了在我们对该服务进行连续的查询时，设置LiveHTTP Headers扩展程序以监听所有的流量。接着，我们将要查看生成的输出，并且判定为了模仿AJAX形态的行为而需要作出什么样的请求。我们打算在本章的下一节中使用这一做法，下一节将谈及因为善意或者恶毒的目的而编写自定义搜索引擎

的内容。但是，首先我们要来讨论一下。

图10-5 LiveHttpHeaders主窗口

从Firefox内部启用LiveHTTPHeaders扩展程序并访问一个包含AJAX Search对话框的页面，例如，www.google.com/uds/samples/apidocs/helloworld.html。在提交了一个查询后，LiveHTTPHeaders将揭示幕后发生了什么事。从结果页面内部确保已经启用了位于如图10-6所示的界面所有右部区域的“show all results（显示所有结果）”按钮。在Web部分启用该按钮十分有必要，因为只有这样才能得到完整的查询。注意，很多结果都定位到.jpg，.gif或png图像。其中的不少结果都会定位到Google提供的Ad Indicator服务，但是最有趣的是那些定位到GwebSearch服务。图10-7显示的是那些实况捕获结果的外观。

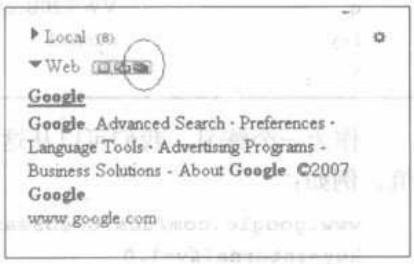


图10-6 显示所有的结果按钮

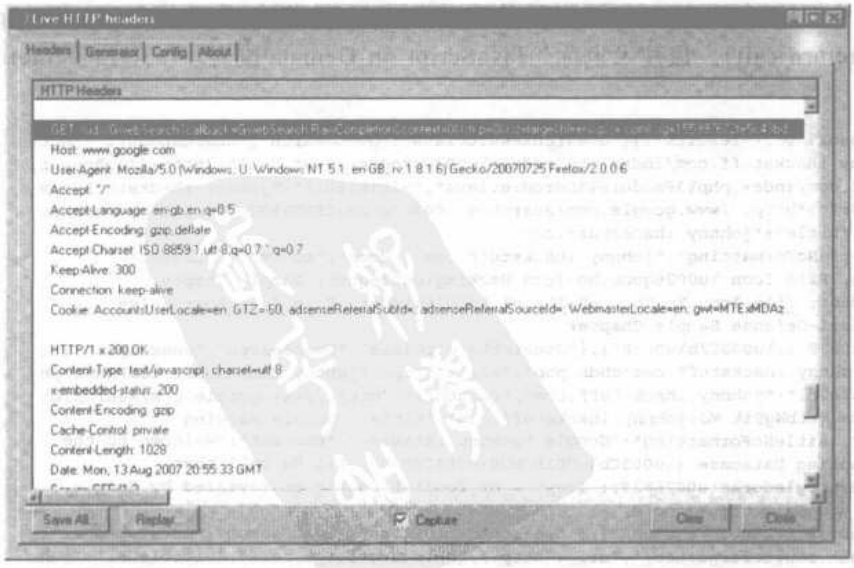


图10-7 LiveHTTP Headers捕获结果

图10-7显示了被用来检索查询的URL格式。以下是一个实例：

```
http://www.google.com/uds/GwebSearch?callback=GwebSearch.RawCompletion&context=0&lstkp=0&rsz=large&hl=en&gss=.com&sig=51248261809d756101be2fa94e0ce277&q=VW%20Beetle&key=internal&v=1.0
```

表10-1列出了每个GET参数，并说明了它们能做什么。

表10-1 GET参数

参数	值	说 明
callback	GwebSearch.RawCompletion	回叫JavaScript函数结果
context	0	-
lstkp	0	-
rsz	large	查询的规模
hl	en	语言参数选择
gss	.com	-
sig	51248261809d756101be2fa94e0ce277	-
q	VW%20Beetle	真实的查询/搜索
key	internal	密钥（使用互联网密钥）
v	1.0	API的版本

作为一个练习，我们可以从这些参数创建一个URL，提供不同的我们认为适合这个任务的值。例如：

```
www.google.com/uds/GwebSearch?callback=our_callback&context=0&rsz=large&q=GHDB&key=internal&v=1.0
```

注意，我们可以把回叫参数从“GwebSearch.RawCompletion”更改为“our_callback”，我们将执行一个GHDB搜索。在浏览器的内部执行URL将会得到一个JavaScript返回调用(JavaScript return call)。该技术也称为JavaScript on Demand或者JavaScript remoting，结果如下所示：

```
our_callback('0',{"results":[{"GsearchResultClass":"GwebSearch","unescapedUrl":"http://johnny.ihackstuff.com/index.php?module\u003Dprodreviews","url":"http://johnny.ihackstuff.com/index.php%3Fmodule%3Dprodreviews","visibleUrl":"johnny.ihackstuff.com","cacheUrl":"http://www.google.com/search?q\u003Dcache:IS5G5YGJmHIJ:johnny.ihackstuff.com","title":"johnny.ihackstuff.com -
Home","titleNoFormatting":"johnny.ihackstuff.com - Home","content":"Latest
Downloads. File Icon \u0026quot;No-Tech Hacking\u0026quot; Sample Chapter
\u0026middot; File Icon Yo Yo  SKillz #1 \u0026middot; File Icon Aggressive
Network Self-Defense Sample Chapter
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://johnny.ihackstuff.com/ghdb.php","url":"http://johnny.ihackstuff.com/ghdb.php","visibleUrl":"johnny.ihackstuff.com","cacheUrl":"http://www.google.com/search?q\u003Dcache:MxfbWg9ik-MJ:johnny.ihackstuff.com","title":"Google Hacking
Database","titleNoFormatting":"Google Hacking Database","content":"Welcome to the
Google Hacking Database (\u003Cb\u003EGHDB\u003C/b\u003E)! We call them
\u0026#39;googledorks\u0026#39;: Inept  or foolish people as revealed by Google.
Whatever you call these fools,
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://ghh.sourceforge.net/","url":"http://ghh.sourceforge.net/","visibleUrl":"ghh.sourceforge.net","cacheUrl":"http://www.google.com/search?q\u003Dcache:WbkSIUl0UtMJ:ghh.sourceforge.net","title":"GHH - The \u0026quot;Google Hack\u0026quot;
```

```
Honeypot","titleNoFormatting":"GHH - The \u0026quot;Google Hack\u0026quot;
Honeypot","content":"\u003Cb\u003EGHDB\u003C/b\u003E Signature #734
(\u0026quot;File Upload Manager v1.3\u0026quot; \u0026quot;rename to\u0026quot;)
\u003Cb\u003E...\u003C/b\u003E \u003Cb\u003EGHDB\u003C/b\u003E Signatures  are
maintained by the johnny.ihackstuff.com community.
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://thebillygoatcurse.com/11/","url":"http://thebillygoatcurse.com/11/","visibl
eUrl":"thebillygoatcurse.com","cacheUrl":"http://www.google.com/search?q\u003Dcache
:O30uZ81QVCcJ:thebillygoatcurse.com","title":"TheBillyGoatCurse.com \u00BB Blog
Archive \u00BB Convert
\u003Cb\u003EGHDB\u003C/b\u003E","titleNoFormatting":"TheBillyGoatCurse.com \u00BB
Blog Archive \u00BB Convert GHDB","content":"The Google Hacking Database
(\u003Cb\u003EGHDB\u003C/b\u003E) has one problem\u2026 it only uses the Google
search  index. The trouble is that advanced search syntax can differ between
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://www.ethicalhacker.net/index.php?option\u003Dcom_smf\u0026Itemid\u003D35\u00
26topic\u003D184.msg328;topicseen","url":"http://www.ethicalhacker.net/index.php%3F
option%3Dcom_smf%26Itemid%3D35%26topic%3D184.msg328%3Btopicseen","visibleUrl":"www.
ethicalhacker.net","cacheUrl":"http://www.google.com/search?q\u003Dcache:EsO7aMyCR6
wJ:www.ethicalhacker.net","title":"The Ethical Hacker Network - Google Hacking
Database (\u003Cb\u003EGHDB\u003C/b\u003E)","titleNoFormatting":"The Ethical Hacker
Network - Google Hacking Database (GHDB)","content":"The Ethical Hacker Network -
Your educational authority on penetration testing  and incident response., Google
Hacking Database
(\u003Cb\u003EGHDB\u003C/b\u003E)"},{"GsearchResultClass":"GwebSearch","unescapedUr
l":"http://snakeoillabs.com/downloads/GHDB.xml","url":"http://snakeoillabs.com/down
loads/GHDB.xml","visibleUrl":"snakeoillabs.com","cacheUrl":"http://www.google.com/s
earch?q\u003Dcache:5nsf_DfjX4YJ:snakeoillabs.com","title":"\u003Cb\u003Eghdb\u003C/
b\u003E xml","titleNoFormatting":"ghdb xml","content":"PS: this vulnerability was
found early this year (search google for the full  report), but was never added to
the \u003Cb\u003EGHDB\u003C/b\u003E for some reason.
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://www.gnucitizen.org/projects/ghdb","url":"http://www.gnucitizen.org/projects
/ghdb","visibleUrl":"www.gnucitizen.org","cacheUrl":"http://www.google.com/search?q
\u003Dcache:dPVtU_3tmnMJ:www.gnucitizen.org","title":"\u003Cb\u003EGHDB\u003C/b\u00
3E | GNUCITIZEN","titleNoFormatting":"GHDB |
GNUCITIZEN","content":"\u003Cb\u003EGHDB\u003C/b\u003E (aka Google Hacking
Database) is HTML/JavaScript wrapper application that  uses advance JavaScript
techniques to scrape information from Johnny\u0026#39;s Google
\u003Cb\u003E...\u003C/b\u003E"},{"GsearchResultClass":"GwebSearch","unescapedUrl":
"http://www.ghdb.org/","url":"http://www.ghdb.org/","visibleUrl":"www.ghdb.org","ca
cheUrl":"http://www.google.com/search?q\u003Dcache:Y6lwVyfCQw8J:www.ghdb.org","titl
e":"Menu","titleNoFormatting":"Menu","content":"\u003Cb\u003E...\u003C/b\u003E to
contact us for any reason, or maybe just leave a comment (good, bad or  ugly, but
not offensive) in our guestbook. Best regards The team at
\u0026#39;\u003Cb\u003EGHDB\u003C/b\u003E\u0026#39;
\u003Cb\u003E...\u003C/b\u003E"}],"adResults":[]}, 200, null, 200)
```


10.1.3 攻击AJAX Search Engine

现在，我们已经知道了如何通过Google的AJAX界面来查询，让我们一起来看一下访问数据的方法。我们将从以下HTML开始——将以下HTML粘贴到一个空白的html文件中，并且在浏览器中打开：

```
<html>
  <head>
    <title>Hacking AJAX API</title>
  </head>
  <body>
    <script>
```

```
            function our_callback(a, b, c, d, e) {
               for (var i = 0; i < b.results.length; i++) {
                  var link = document.createElement('a');
                  link.href = b.results[i].url;
                  link.innerHTML = b.results[i].url;

                  document.body.appendChild(link);

                  var br = document.createElement('br');

                  document.body.appendChild(br);
               }
            }
         </script>
         <script type="text/javascript"
src="http://www.google.com/uds/GwebSearch?callback=our_callback&context=0&rsz=large
&q=GHDB&key=internal&v=1.0"></script>
      </body>

</html>
```

该代码将会向Google的GwebSearch服务提交一个GHDB的请求。注意，回叫参数会定位回our_callback（已在代码的前部定义）。该函数会简单地抓取数据，并且在链接格式的页面DOM内呈现数据。

尽管这看起来很有趣，但是要做的事情还是很多的。让我们来看一下下面的这个实例，这个实例可以动态地抓取所有来自Google Hacking Database的特定类别的条目，执行测试查询并且在一个单页中列出结果：

```
<html>
   <head>
      <title>GHDB Lister</title>
   </head>
   <body>
      <script>
         function get_json(url, callback) {
            var name = '__json_' + (new Date).getTime();

            var s = document.createElement('script');
            s.src = url.replace('{callback}', name);

            window[name] = callback;

            document.body.appendChild(s);
         }
get_json('http://www.dapper.net/transform.php?dappName=GoogleHackingDatabaseReader&
transformer=JSON&extraArg_callbackFunctionWrapper={callback}&applyToUrl=http%3A//jo
hnny.ihackstuff.com/ghdb.php%3Ffunction%3Dsummary%26cat%3D19',
            function (data) {
```

```
                console.log(data);
                for (var i = 0; i < data.groups.entry.length; i++) {
                  var query = data.groups.entry[i].query[0].value;
                  var description =
data.groups.entry[i].description[0].value;

get_json('http://www.google.com/uds/GwebSearch?callback={callback}&context=0&rsz=la
rge&q=' + escape(query) + '&key=internal&v=1.0',
                         function (a, b, c, d, e) {
                           if (!b) {
                             return;
                           }
   for (var i = 0; i < b.results.length; i++)
{
                             var link = document.createElement('a');
                             link.href = b.results[i].url;
                             link.innerHTML = b.results[i].url;

                             document.body.appendChild(link);

                             var br = document.createElement('br');

                             document.body.appendChild(br);
                           }
                         });
                }
              });
    </script>
  </body>
</html>
```

运行这个实例时，你可以得到一个与图10-8所示的页面相似的页面。

图10-8 结果页面

让我们先来分析一下文件。如你所见，页面仅有一个脚本块。该脚本块可以通过Dapper（http://dapper.net）抓屏服务从GHDB获取一个查询列表。我们可以搜索与GHDB第19条（叫做“Advisories and Vulnerabilities”）相符的URL http://johnny.ihackstuff.com/ghdb.php?function=summary&cat=19。该Scraper捕获了几个其他的有趣的事（目前我们还不是太感兴趣）。

背景知识

使用Dapper抓屏

使用Dapper来抓取各种与安全相关的数据库，并且将该信息用作设计良好的面向客户端的攻击向量的一部分，这些内容在OWASP中首次讨论过，那是在2007年的意大利，由程序设计者Petko D. Petkov提出——同时他也是一名pdp（结构设计师）。要想了解该主题的更多详情，可以访问http://www.gnucitizen.org和http://www.gnucitizen.org/projects/6th-owasp-conference。

一旦检索到这些列表，我们可以枚举出各个条目并且创建自定义Google AJAX API查询：

```
get_json('http://www.google.com/uds/GwebSearch?callback={callback}&context=0&rsz=
large&q=' + escape(query) + '&key=internal&v=1.0',
```

如你所见，代替静态字符串的是，我们实际上可以提交一个来自GHDB中获取的信息的查询。接下来的对Google AJAX Search API的查询将会检索到示例结果函数，而且回叫函数将会在页面DOM内部解析它们。

理解函数geg_json的目的很重要。该函数只是一个帮助程序，它可以为我们节约重复编写相同程序的大部分时间。get_json函数可以为回叫程序简单地生成一个独特的名称，并为它赋予一个全局的作用范围。接着，它会把该名称提交给标注有占位符{callback}的callback域，并且调用外部脚本。

该技术可以作为http://www.gnucitizen.org/ghdb上的GHDB Proof of Concept应用程序的一部分执行，如图10-9所示。

图10-9 GNUCITIZEN GHDB

该应用程序可以动态地抓取到所有来自Johnny Long的Google Hacking Database（http://johnny.ihackstuff.com）的信息，并且将它们以一个很漂亮的图形格式呈现给用户。你可以通过选择一个类别来浏览每个向量，并接着选择一个你感兴趣的查询。注意，该应用程序在我们每次选择查询时都提供了一个动态反馈。窗口的底部包含了排行前几位的查询，可以通过Google的AJAX Search API界面获得。

背景知识

XSS和AJAX蠕虫

该技术可以通过XSS/AJAX蠕虫来实现目标搜索并且利用目标来获取有用的信息，因此要确保将来的繁殖。XSS/AJAX蠕虫通常都在源域中繁殖和传播。这是因为JavaScript没有执行跨站点请求的能力。本章提及的该技术允许蠕虫绕过JavaScript的限制，并且可以访问在线的其他资源。要想了解该主题的更多详情，可以访问以下资源：http://www.gnucitizen.org/blog/googlesearch-api-worms、http://www.gnucitizen.org/projects/ghdb和http://www.gnucitizen.org/blog/the-web-has-betrayed-us。

10.2 Calendar

Google Calendar是一个功能强大的日历管理应用程序，它支持类似日历共享、邀请创建、搜索以及日历发布。该服务也可以与Google Mail（Gmail）结合起来，并且可以通过移动设备来访问。总而言之，Google Calendar对于我们每天的工作非常有帮助。

因为单个用户便可以维护其他用户可能也很感兴趣的事件列表以及日历，所以Calendar共享是一个非常有用的功能。通常，为了共享日历，你必须从如图10-10所示的日历管理界面明确地做好这些工作。

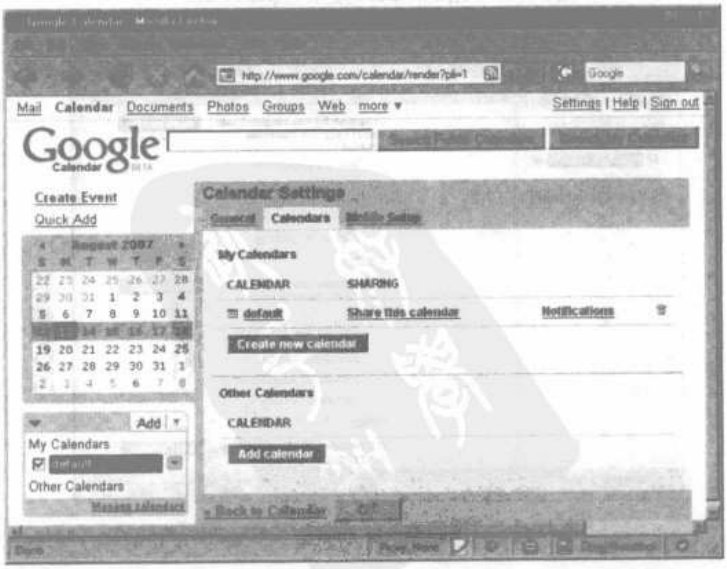


图10-10 Calendar管理界面

一旦日历共享了，那么每个人都可以查阅日历或者甚至是订阅内部的事件。这可以通过Calendar应用程序或者任何RSS反馈阅读程序做到。

作为一名安全专家，这些共享的日历尤其有趣。通常，甚至是当执行那些最基本的搜索时，都完全有可能无意间发现那些可以执行恶意目的的敏感信息。例如，登录到Calendar内部，并且查找术语“password”会返回如图10-11所示的很多结果。

如你所见，这里有几个可以满足我们搜索标准的日历条目。在这些条目中，有少许十分有趣的并且值得我们关注的几项。另一个有趣的可以带回大量丰富信息的查询是“passcode”，如图10-12所示。

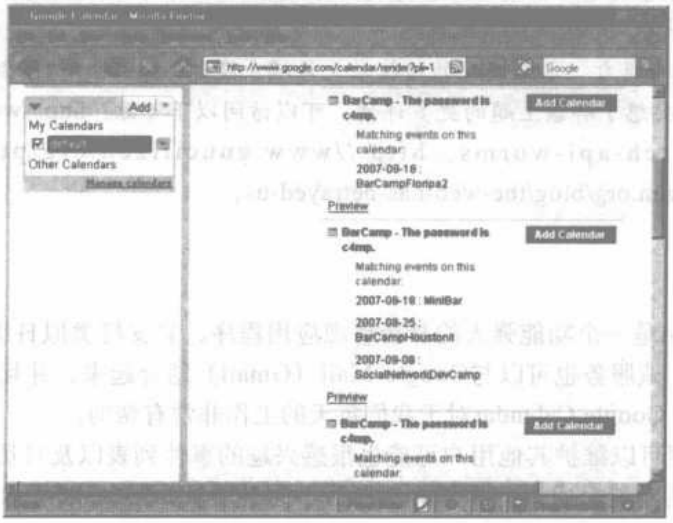


图10-11 用Calendar搜索“password”

图10-12 用Calendar搜索“passcode”

图10-12泄露了几个预定的电话会议。注意，会议电话号码和访问代码也列了出来。攻击

者可以在预定的时间很容易地加入到电话会议中，并且安静地窃听整个会议。任务完成。从该会话中可以学到大量的有用的东西，例如公司机密、操作系统的技术细节等。

当然，我们也可以尝试以上查询的各种变体，甚至是在查询中间放置更多的关键字，以得到更好的查询结果。例如，查询“username password”会返回有关谁会在他们的日历中存储敏感登录信息的结果，如图10-13所示。

这仅仅是开始，你还可以查找生日、宠物名等。正如你了解的，大量的密码提示工具都有一个机密的问题。这些机密问题的答案通常取自我们的日常生活，因此我们才会牢记。不过，Calendar应用程序也可能包含我们日常的活动。当我们摸清了所有的这些问题，我们便可以通过简单地查看他们的日历来窃取到目标用户账户。

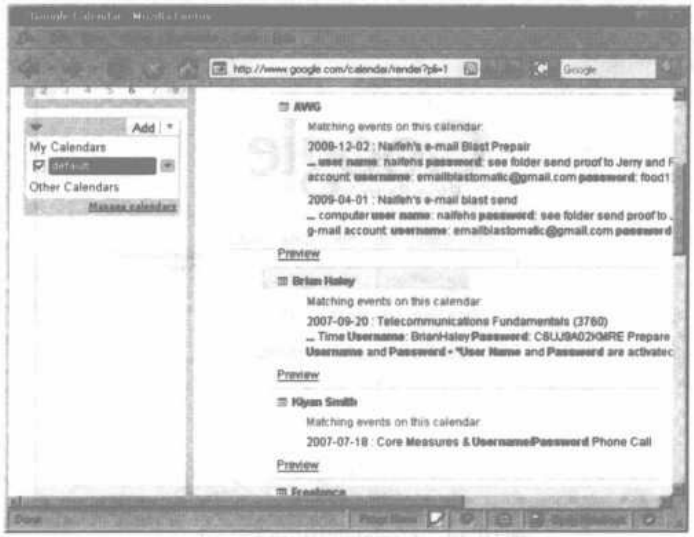


图10-13 用Calendar搜索“username password”

有很多不同的方法导致Calendar服务会被滥用。我们作为用户需要考虑的主要的、大多数的重要的安全事项就是Google事件单元中封闭的信息是否敏感、是否会被用来伤害我们。

10.3 Blogger和Google的Blog Search

Blogger是Google的博客软件，它位于blogger.com和blogspot.com之上。Blogger是使用得最广泛的博客平台之一。它允许多个博客的实时创建，并且具有一些与其他软件合作并且避免注释和引用垃圾博客（trackback spam）的功能。

当我们谈及博客时，我们需要注意几点。第一也是最重要的一点是博客是公开的，这意味着可以被互联网上的任何人阅读。这也就是说，不要将关于自己的之后有可能给你生活造成严重影响的信息发布在博客上。只要在Web上出现的东西就会一直留在Web上。博客传播由多个在线服务聚合而成。你在博客中发布的信息一旦发布后就几乎不可能删除了。你博客上的信息将很有可能被你未来的雇主用作在应聘工作时的标准的背景核查，如图10-14所示。我们已经证明了少量简单的查询可以泄露大量有趣的信息。Google简化了在博客中寻找有趣信息的过程。

先来看一下如图10-14的Google的Blog Search。

尽管Google的Blogger服务会有效地阻止内容并且trackback SPAM，这里仍有一个漏洞。当源于博客的SPAM自己发布自己时会出现什么情况？

进入SPLOG.Splogs或者Spam Blogs——它们都是反映从外部删减/收集到的内容的普通的博客，但是也提供了它们的使用者需要的其他信息。

Splog吸引恶意攻击者的原因有很多。第一个原因是因为攻击者不需要写博客，写博客是一个非常耗时的任务，不过它也会吸引人注意并让人订阅。因为Splog的搜索引擎的排名在不断地提升，所以它会吸引更多的访问者。如果攻击者想要利用面向流行的Web浏览器的Splog页面上的漏洞应用的话，他就应该在瞬间接管上百台机器。

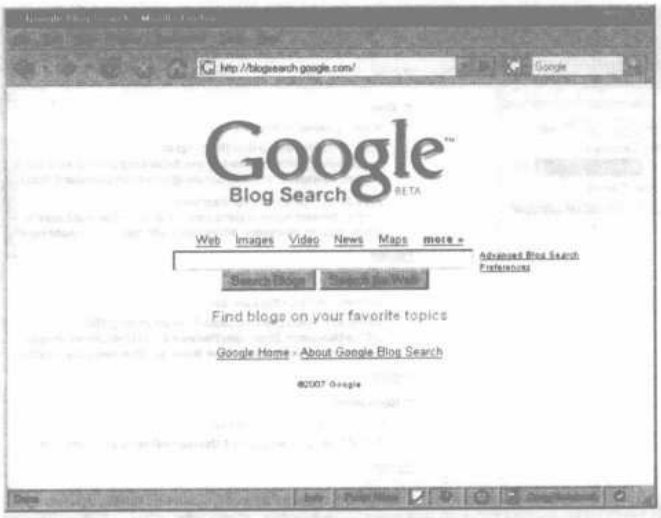


图10-14 Google的Blog Search

另外，Splog可能包含那些为博主带来收入的广告。Splog越流行，收入就会越高。如果每个Splog每天可以赚20美元，那么多个Splog可以赚更多的钱。Splogs是一个全天候不间断的商业模式，可以赚大笔的钱。

不管恶攻击者是否要使用Splog技术吸引受害者或者是赚钱，查看后台发生的事都很有趣。在下一节，我们将要详细介绍Splog是如何工作的。我们将分析一个使用Google的Blogger服务的Splog生成脚本。

Google Splogger

Google有一个非常卓越的应用程序设计接口（application programming interface， API）。最著名的Google服务有来自Google Data的GData。GData允许开发者对Google的服务执行程序化操作。例如，GData可以用来程序化Google Calendar实例的更新。GData也可以用来创建、删除和管理Blogger条目并且管理我们在Google Base上的提交项。这样一来，这个阶段看起来像是为Google的博客服务设置的以用作Splog的基础一样。本节中，我们将说明该过程是如何

完成的，但是请注意，我们这样做不是为了培养罪犯。如果你不怀好意那么请仔细地考虑一下，Google有权阻止你访问它们的服务。如果你执意使用他们的服务来进行恶意攻击，那么他们有权对此诉讼。

本例中，我们将使用GData的Blogger界面。下面的脚本允许我们程序化地登录到Blogger，并且提交一个新的公告。我们可以有效地使用一个类似的接近于自动化接收RSS Feed（RSS文件）的方法，接着把它们上传到一个特别的博客账号上，这个账号以后可以用于Splog目的。

```
# GoogleSplogger
# Copyright (C) 2007 Petko D. Petkov (GNUCITIZEN)
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

__version__ = '1.0'
__author__ = 'Petko D. Petkov; pdp (architect)'

__doc__ = """
GoogleSplogger (GNUCITIZEN) http://www.gnucitizen.org
by Petko D. Petkov; pdp (arhictect)
"""

import atom
import gdata.service

class GoogleSplogger:
    """
    GoogleSplogger

    The power of Blogger in a single object
    """
    def __init__(self, email, password):
        self.client = gdata.service.GDataService(email, password)
        self.client.source = 'Splogger ' + __version__
        self.client.service = 'blogger'
        self.client.server = 'www.blogger.com'
```

```
        self.client.ProgrammaticLogin()
        self.available_blogs = self.get_blogs()

    def get_blogs(self):
        """
        get_blogs -> Dict

        Get a dictionary of available blogs.
        """
        blogs = {}

        feed = self.client.Get('/feeds/default/blogs')

        for i in feed.entry:
            title = i.title.text
            for a in i.link:
                if a.rel == 'self':
                    blogs[title] = a.href.split('/')[-1]
        return blogs

    def post(self, blog_name, title, content, author_name):
        """
        post(blog_name, title, content, author_name) -> ?

        Post a new entry to blog
        """
        if blog_name not in self.available_blogs:
            raise 'blog name not found'

        entry = gdata.GDataEntry()
        entry.author.append(atom.Author(atom.Name(text=author_name)))
        entry.title = atom.Title('xhtml', title)
        entry.content = atom.Content('html', '', content)

        return self.client.Post(entry, '/feeds/' \
            + self.available_blogs[blog_name] + '/posts/default')

    def usage(prog):
        print 'usage: ' + prog + ' -u username -p [password] -P blog ' \
            '-t title -c [content] -a author'
        print ' ' + prog + ' -u username -p [password] -l'
        print '-u      username username for the login'
        print '-p      [password] password for the login'
        print '-P      blog post to blog'
        print '-t      title title for the new post'
        print '-c      [content] content for the new post'
```

```
        print '-a      author author for the new post'
        print '-l      list available blogs'
        print '-h      print this page'

if __name__ == '__main__':
    import sys
    import getopt
    import getpass

    try:
        opts, args = getopt.gnu_getopt(sys.argv[1:], 'hlcpu:p:P:t:c:a:' )

except Exception, e:
    print e
    print

    usage(sys.argv[0])
    sys.exit()

username = None
password = None
action = None

post_blog = None
post_title = None
post_author = None
post_content = None

for key, val in opts:
    if key == '-h':
        usage(sys.argv[0]);
        sys.exit();

    elif key == '-l':
        action = 'list'

    elif key == '-P':
        action = 'post'
        post_blog = val

    elif key == '-u':
        username = val

    elif key == '-p':
        password = val

    elif key == '-t':
        post_title = val
```

```
        elif key == '-a':
            post_author = val

        elif key == '-c':
            post_content = val

if not action or not username:
    usage(sys.argv[0])
    sys.exit()

if action == 'post' and \
    (not post_blog or not post_title or not post_author):
    usage(sys.argv[0])
    sys.exit()

if not password:
    password = getpass.getpass( 'password:  ')

try:
    gs = GoogleSplogger(username, password)

except Exception, e:
    print e
    sys.exit()

if action == 'post' and post_blog not in gs.available_blogs:
    print 'blog not found within the user profile'
    sys.exit()

if action == 'post' and not post_content:
    post_content = sys.stdin.read()

if action == 'list':
    for i in gs.available_blogs:
        print i

elif action == 'post':
    gs.post(post_blog, post_title, post_content, post_author)
```

注意

GoogleSplogger.py要求为Python提供Google的GData API库。该库可以从以下URL地址获取：http://code.google.com/p/gdata-python-client/。下载完GData API库以后，提取存档的内容，并且通过命令行进入文件夹。请确保你已经得到了安装Python模块所需的允许，并且键入：python setup.py。

setup.py脚本应该在没有任何问题的情况下安装其余的API。

有几种方法可以用来运行我们在此列出的脚本。例如，为了列出当前在我们的配置下注册的博客名称，可以键入如下的命令。

```
python GoogleSplogger.py -l -u username -p password
```

注意，如果你不为-p（password）标记提供值的话，那么在运行时，系统要求你键入该值。如果你不想让你的密码在系统和命令日志文件中留下痕迹的话，那么不妨在运行时才为-p标记提供值。命令行的抽样输出如图10-15所示。

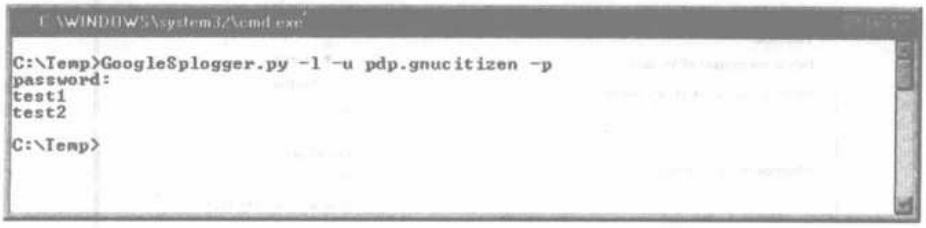


图10-15 枚举当前博客

一旦我们知道了博客的名称，便可以在这些博客上进行发布。例如：

```
python GoogleSplogger.py -u username -p -P blog_name_here -t title_for_the_post -a author -c
```

在执行这行命令后，系统会要求你输入密码以及发布内容。当你输入了发布内容后，可以通过在Windows中按下Ctrl+Z或者在UNIX中按Ctrl+D来终止输入。如图10-16所示。

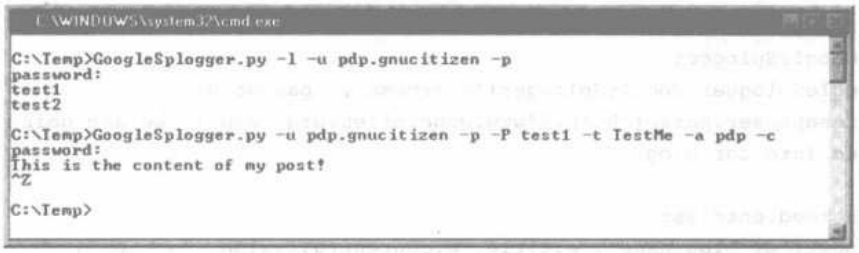


图10-16 在Blogger进行发布的命令

这一切都很简单，但是进程可以在以后进行改进。以下是另一种发布新博客条目的方法，这次使用密码内联：

```
python GoogleSplogger.py -u username -p password -P blog_name_here -t
title_for_the_post -a author -c << EOF
```

一旦你在一个新的命令行上写出了post类型EOF，那么，公告便可以由文件来提交：

```
python GoogleSplogger.py -u username -p password -P blog_name_here -t
title_for_the_post -a author -c < file.txt
```

在Blogger中程序化插入新的公告并不那么有用。但是下面的示例将会演示抓取某个人的博客数据以及将它放入自己的博客有多么容易。为了这个目的，我们将需要另一个python工具，该工具基于一个来自http://cheeseshop.python.org/pypi/FeedParser/4.1的名为FeedParser库。该数据包的安装程序可用于所有的python包。

图10-17 结果

从命令行启动python，并且确保GoogleSplogger.py脚本位于你当前的工作目录。键入以下命令：

```
import feedparser
import GoogleSplogger
gs = GoogleSplogger.GoogleSplogger('username', 'password')
feed = feedparser.parse('http://www.gnucitizen.org/feed')# we are going to import
this feed into our blog

for e in feed.entries:
    gs.post('my blog name', e.title, e.content[0].value, 'author')
```

输入的如图10-18所示的脚本将会把来自GNUCITIZEN的博客的所有的条目导入你的博客，有效地创建一个如图10-19所示的垃圾博客。

图10-18 导入博客条目

注意，我们使用这些内容创建一个新博客有多快速。

图10-19 新的Splog

这是完全不同的一组策略，可以被采纳来用来创建Splog并且达成它的目的，但是因为本主题的篇幅，我们不可能在书中讲解所有的细节。重要的是理解与Splog相关的安全以及伦理。再一次要提醒你的是，用Google或者其他服务进行垃圾服务都是与这些服务的条款相违背的。你可能会损害你的账号以及账号里的所有数据。

10.4 信号警报

通常，我们都需要跟踪Coogle结果集的变化。例如，假设我们要跟踪某个特定站点的漏洞。我们应该怎么做呢？偶尔，我们可以简单地运行扫描程序，但是这是一个干挠性的练习，一定会占用大量的时间。相反，作为一个专业的Google黑客，我们可以使用Google本身，并且使用少量强大的Google dork来搜索我们感兴趣的事情，而不需要使用那些自动扫描的软件。接着，我们可以设置一个cron任务来监视Google返回的结果，并且在发现变化时通过E-mail通知我们结果。

接下来，我们将使用如图10-20所示的Google Alerts来进行演示。

Google Alerts是一个强大的工具，可以用来发觉查询结果何时变化。该系统可以被修改为每天或者每周都发送更新，抑或是有更新时发送更新。记住，只有前10个条目（第1页）才会纳入考虑范围。然而，Alert系统在优化后可以工作得十分出色。

这是一个很出色的工具，它还可以用于更有趣的目的。假设，我们知道目标服务器使用MsSQL作为数据库终端。我们可以使用Google Alerts来查询目标，并且在出现错误时搜索错误信息。该查询可能看起来很类似于下面的语句：

```
"[SQL Server Driver][SQL Server]Line 1: Incorrect syntax near" -forum -thread -
showthread site:example.com
```

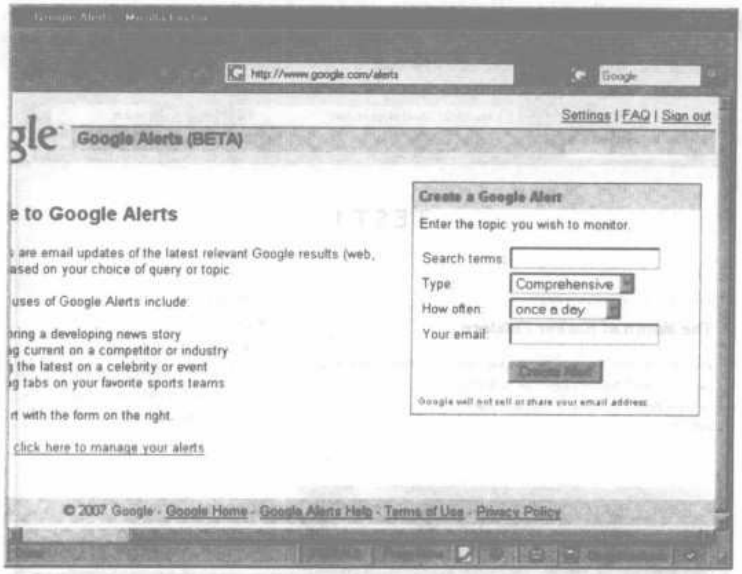


图10-20 Google Alerts

为警报类型选择Web——通常这是默认的选项。选择警报的频率以及你的E-mail地址并且单击Create Alert（创建警报）。

注意，我们为该警报使用的查询是受域限制的（site:example.com）。同样还要注意实际的Google dork。显然，我们是在寻找那些看似是在发送给后端的SQL查询中产生的故障的信息。这些类型的信息被作为SQL注入漏洞资源接收。

恶意用户可以使用该服务来警示目标站点上出现的漏洞和有趣信息。这个警示的配置非常低，并不会让目标有所警惕；该业务发生在用户和Google之间。攻击者甚至可以碰到Google Hacking Database中所有条目的警报。尽管这有些被滥用了，但是数据库中的某些条目确实揭露了极为敏感的信息，在这些敏感的信息的帮助下，攻击者只需要少许的力气便可以收获颇丰。

10.5 Google Co-op

Google Co-op（www.google.com/coop）是一个非常强大的服务，允许你创建一个强大的自定义搜索引擎。你不需要为了使用该服务而注册为Google用户，但是如果你需要创建一个引擎，则需要注册。在下面的内容中，我们将指导你使用该服务的一些非常有趣的功能，而且我们将向你展示如何创建你自己的搜索引擎。

让我们先从最简单的搜索引擎开始着手。浏览Google Co-op页面并且单击“Create a Custom Search Engine”（创建一个自定义搜索引擎）按钮，或者简单地浏览www.google.com/coop/cse。我们可以从如图10-21所示的自定义引擎（Custom Engine）配置页面来定义我们需要的功能。

首先键入一个搜索引擎的名称。这里我们把自己的搜索引擎称为“Google Hacking Database Search”。键入一个说明以及一些基本的搜索关键字，这些键入内容都是可选的。这些

关键字主要被Google用来查找最相关的结果。这意味着我们的查询将与这些关键字混合在一起。眼下，我们将暂时不管它。继续来看标题为“What do you want to search”的字段，我们将定义搜索查询的范围。对于本例来说，我们将使用这个名为“Only sites that I select”的默认选项。

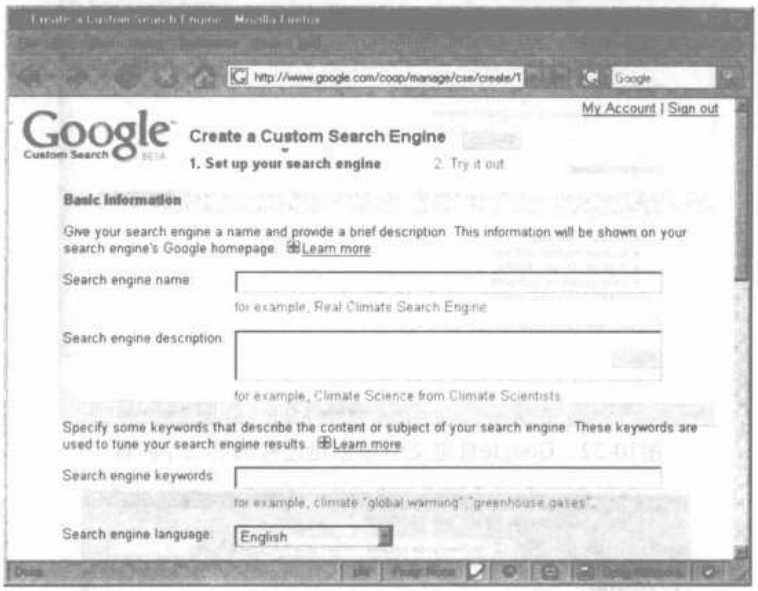


图10-21 Google自定义搜索引擎创建页面

现在，我们到了最有趣的部分，我们需要提交当执行查询时Google要查询的URL。因为我们的搜索引擎将会围绕http://johnny.ihackstuff.com/ghdb.php上的Google Hacking Database来做事，我们只需要将URL填入到该域中即可。我们可以通过使用通配符来进一步自定义该选项，以搜索与特定语法相匹配的URL。以下是摘自Cop-op文档的部分实例：

```
www.mysite.com/mypage.html - look for information within mypage.html part of the
www.mysite.com domain
www.mysite.com/*-look for information within the entire context of www.mysite.com
www.mysite.com/*about*-look for information within URLs from www.mysite.com that
has the about keyword
*.mydomain.com - look for information within sub-domains of mysite.com.
```

本例中，主页位于http://johnny.ihackstuff.com/ghdb.php，但是为了让Co-op从该位置下降几级，我们必须将站点的URL更改为http://johnny.ihackstuff.com/ghdb.php*（注意星号在最后）。这是因为个人数据库条目的URL包含如下格式的附加在ghdb.php脚本名称后的数据上的参数：

```
http://johnny.ihackstuff.com/ghdb.php?function=detail&id=64
```

此时，不需要设置Go-op自定义引擎创建页面的其他选项。同意Google的服务条款，并且单击next（下一步）按钮。便可以看到与图10-22所示的页面相似的页面。

我们现在将要来测试这个搜索引擎是如何工作的。键入一些类似“index”或者“secret”的查询，你将看到一些示例结果。如果所有的查询都能按预期的那样工作，那么请点击finish

（完成）按钮，自定义搜索引擎的外形如图10-23所示。

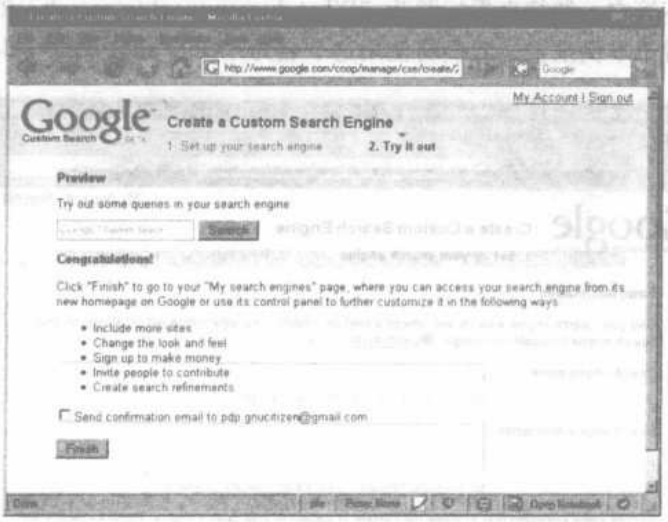


图10-22 Google自定义引擎创建过程的第二个阶段

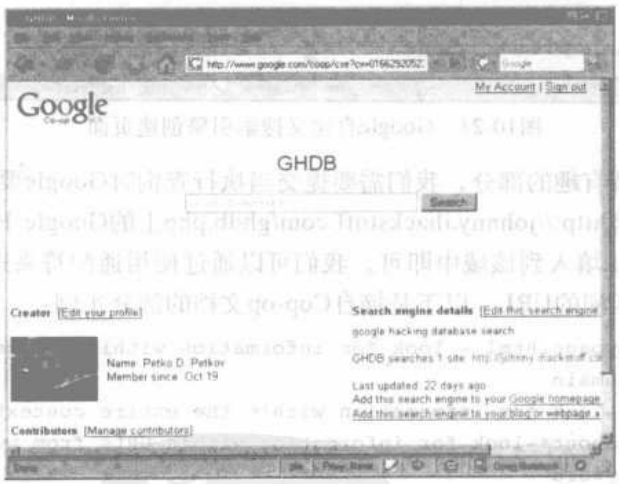


图10-23 GHDB自定义搜索引擎

搜索引擎的目的是在Johnny Long的优秀的Google dork的集合中查找有趣的查询，该集合不支持搜索（在编写此书时）。例如，查询passwd会返回与如图10-24所示的结果相似的结果。

创建其他自定义搜索引擎也很简单。例如，我们可以搜索在www.phenoelitus.org/dpl/dpl.html上创建的默认口令的Phenoelit数据库。通常，这会花很长的时间来加载文件，大多数浏览器都不能处理这么庞大的尺寸。让我们自定义一个搜索引擎来查找这个优秀的页面。

重复之前的步骤，我们将使用一个受限站点www.phenoelitus.org/dpl/dpl.html来替代http://johnny.ihackstuff.com/ghdb.php*。继续完成该引擎的制作并进行一个测试。图10-25显示了查询cisco的结果。

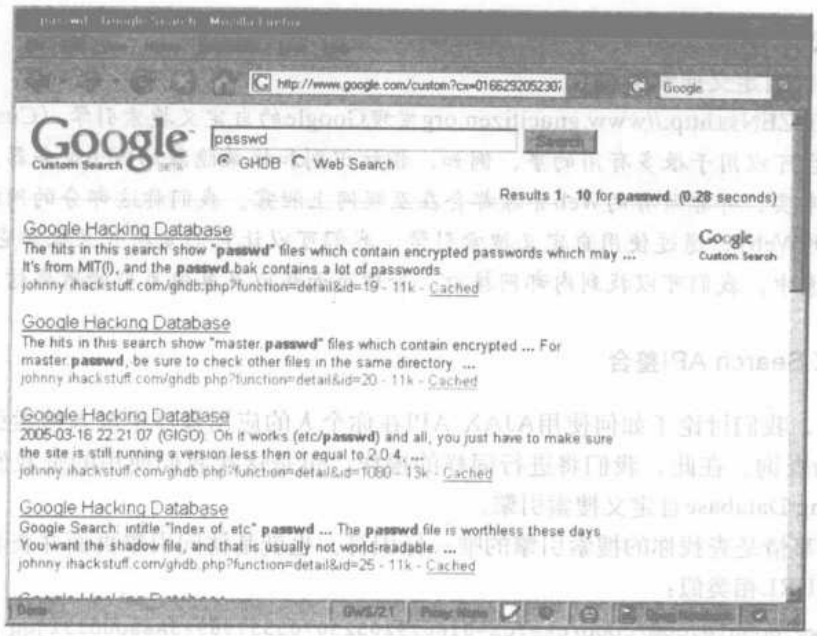


图10-24 搜索结果

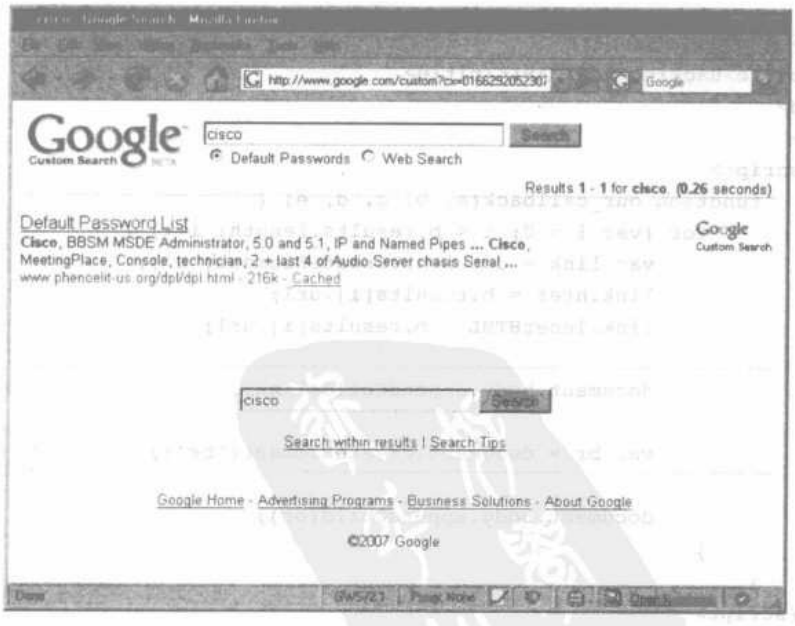


图10-25 默认密码列表搜索引擎

注意，结果页面包含所有我们需要的细节，如用户名和密码。我们可以通过添加更多的默认密码列表来改善引擎。

背景知识

Google的自定义搜索引擎

GNUCITIZEN组http://www.gnucitizen.org发现Google的自定义搜索引擎（Custom Search Engine）平台可以用于很多有用的事，例如，指纹识别和枚举隐藏的Web服务器。这是一个众所周知的事实，并非所有的Web资源都会在互联网上泄露。我们称这部分的网络为Hidden Web（隐藏的Web）。通过使用自定义搜索引擎，我们可以让它们复原并且枚举它们的内容。在收集的信息中，我们可以找到内部网接口、管理员面板以及其他类型的敏感信息。

Google AJAX Search API整合

如前所述，我们讨论了如何使用AJAX API在你个人的应用程序中嵌入搜索引擎工具，甚至是执行自动查询。在此，我们将进行同样的操作，但是这次我们使用的是我们已经创建的Google Hacking Database自定义搜索引擎。

最重要的事情是查找你的搜索引擎的唯一标识符。也就是访问引擎页面并查看URL栏。它应该与下面的URL相类似：

```
http://www.google.com/coop/cse?cx=016629205230705557969%3Assouol31jqq
```

cx参数是该引擎的唯一标识符。请记住该值并且在下面的实例中使用它来替换占位符。

```
<html>
    <head>
        <title>Hacking AJAX API</title>
    </head>
    <body>
        <script>
            function our_callback(a, b, c, d, e) {
                for (var i = 0; i < b.results.length; i++) {
                    var link = document.createElement('a');
                    link.href = b.results[i].url;
                    link.innerHTML = b.results[i].url;

                    document.body.appendChild(link);

                    var br = document.createElement('br');

                    document.body.appendChild(br);
                }
            }
        </script>
        <script type="text/javascript"
src="http://www.google.com/uds/GwebSearch?callback=our_callback&context=0&rsz=large
&q=test&key=internal&v=1.0&cx=016629205230705557969%3Assouol31jqq"></script>
    </body>
</html>
```

这里有很多有趣的、十分有价值的可以使用AJAX Search API和Google Co-op完成的事情。它不过就是一种假设，是黑客和计算机安全专家最擅长的。

10.6 Google Code

Google Code是对允许提供全免费项目的开放源代码社区的最大的恩惠。在功能方面，该服务与很多人都知道的Sourceforge十分相似。通过SVN，为研发者提供了Wiki来寄存项目文档、Bug跟踪体系以及版本控制。在本书的写作过程中，每个用户可以进行的项目的数目都受到了限制。然而，这个逻辑限制可以被很容易地绕过。

Google Code不只是一个开发环境，它还是一个免费的寄存提供者。我们可以使用该系统来隐藏此处的所有信息。

为了打开一个Google Code项目，你应该先拥有一个Google账号。你只面访问http://code.google.com/hosting/createProject，填写必要的信息，并且要上传如图10-26所示的内容。

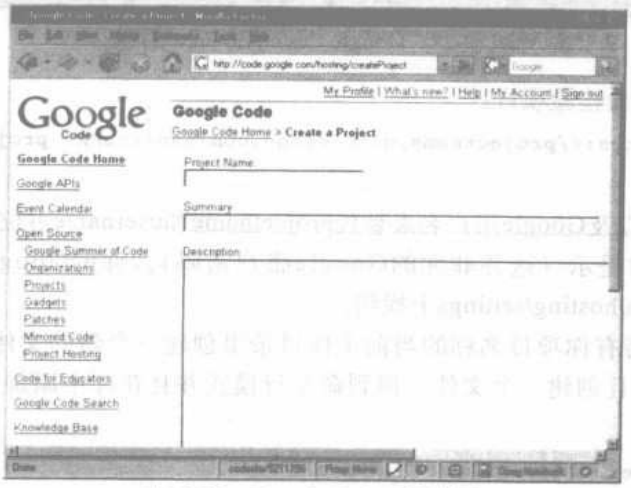


图10-26 Google Code项目注册

如前所述，Google Code运行于SVN（Subvesion）的顶部。为了上传内容，你需要与大多数Linux/UNIX Linux/UNIX分发版本捆绑在一起的svn客户程序。Windows用户可以http://tortoisesvn.net/downloads获得svn客户程序，也可以通过安装Cygwin（www.cygwin.com）并且选择svn包来获得svn客户程序。在本节的其余部分，我们打算通过命令行svn实用程序从控制台进行操作。

10.6.1 SVN简介

在我们继续之前，先让我们来简单地看一下管理系统的SVN版本。

一旦你准备好了发布你的项目，就可以登录到Google Code，并且点击Source选项卡。你将进入项目源页面。该页面显示了有关检验你的项目文件夹的使用说明，如图10-27所示。

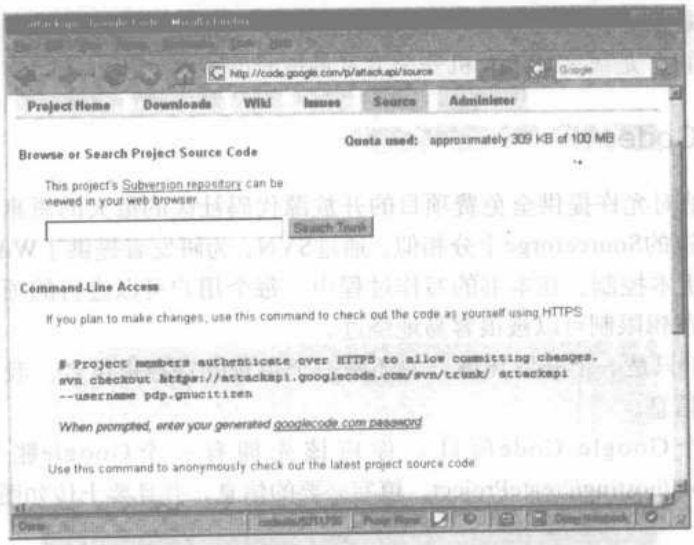


图10-27 Google Code的Source页面

下面的svn命令将会检验项目：

```
svn checkout https://projectname.googlecode.com/svn/trunk/ projectname --username
username
```

使用你的项目名以及Google用户名来替代projectname和username占位符。你将得到请输入Google Code密码的提示（这并非你的Google账户密码）。你的Google Code密码可以在http://code.google.com/hosting/settings上找到。

svn命令将会在带有你项目名称的当前工作目录里创建一个新的文件夹。要想添加文件，可以更改项目目录并且创建一个文件。回到命令行模式并且在库中添加这个文件，命令如下所示：

```
svn add filename
```

一旦你对所有的更改以及新添加的文件满意了，便需要提交该项目。可以通过以下命令行来获得：

```
svn ci -m 'description of the commit'
```

为提交信息提供一个不同的信息（-m）。这个提交信息更详细地描述了你所做的更改。

10.6.2 在线获取文件

一旦你的项目提交到源库中，你便可以在线访问它的内容。你可以通过访问http://projectname.googlecode.com/svn/trunk来获取提交的项目。记住，提交的文件可以作为内容形式的文件/明码或者内容形式的应用程序/八位字节流（如图10-28所示）使用，这样一来便可以防止它们在浏览器中被解析。这意味着，理论上，你应该能查看/预览上传的图片或者HTML文件。

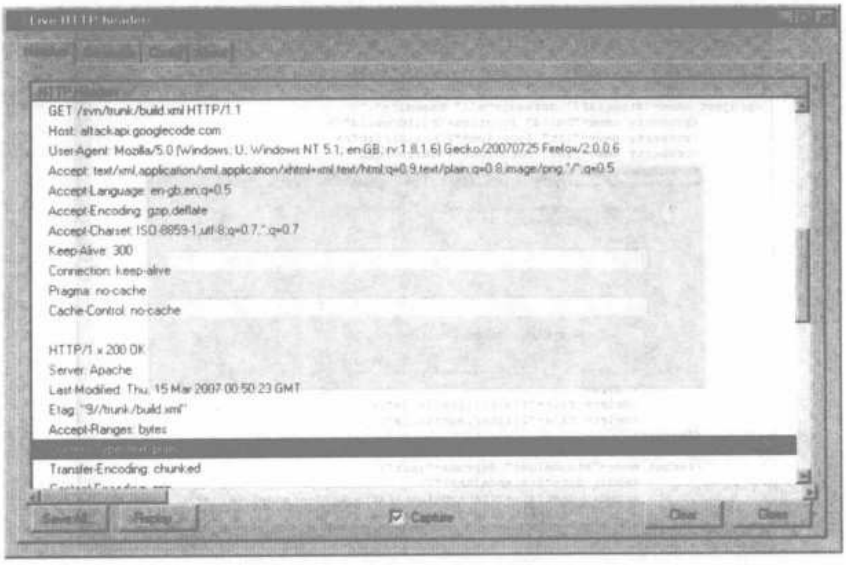


图10-28 用Google Subversion版本输出的Live HTTP Headers

尽管这样，攻击者仍旧可以寄存恶意脚本，该脚本可以利用带有漏洞的浏览器，允许系统控制访问者的浏览器。这正是我们着手查看Google Code开发平台真正潜力的入口。没有任何东西可以防止攻击者在线寄存他们的恶意文件、防止攻击者使用它们来攻击无辜的人。因为ISP（互联网服务提供商）不能完全地阻止Google来停止恶意传播，所以这种情况很值得关注。很多用户的处境都不容乐观。

那些熟悉IDS（侵入检测系统）和IPS（侵入预防系统）的人之所以会反对是因为恶意软件也可以通过将签名设置为可以在流行防火墙产品以及诸如Snort之类的开放源项目中找到的签名来查找。因为Google Code的加密选项，在绝大多数时间，攻击者都不会被察觉。正如我们知道的，加密通信能确保隐私的安全。Google为寄存的项目提供了SSL连接。以下是一个示例：

```
https://projectname.googlecode.com/svn/trunk/path/to/file
```

通过使用https替代URL中的http，我们可以使用加密会话的https协议，在IDS和IPS系统的关注下将数据藏在会话中。因为开发者要使用https接口，所以，Google将会弹出如图10-29所示的认证输入提示。

对于想要寄存浏览器漏洞利用代码的攻击者而言这并非是最好的情形，但是一些HTTP技巧可以帮助你解决它。以下的URL将预提交认证：

```
https://username:password@projectname.googlecode.com/svn/trunk/path/to/file
```

一旦攻击被发觉，任何人都可以使用已提交的证书进入Subversion库并且将文件恢复到非恶意状态。然而，鉴于当今AJAX/XSS蠕虫可以在几个小时内传播给数百万用户的事实，建议的设置是大多数攻击者都乐意执行的一个折衷的办法。

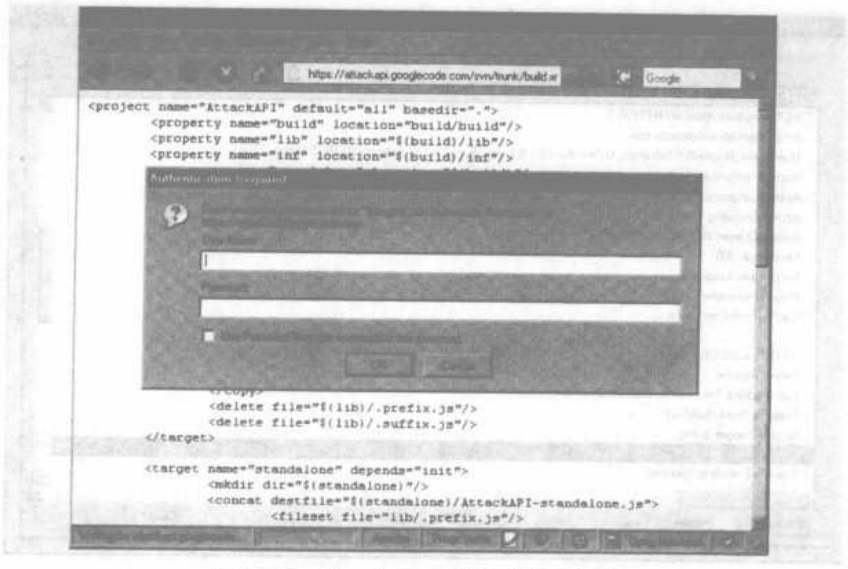


图10-29 Google Code基本认证对话框

注意

记住，所有存储在源代码库中的文件都位于公共域中。不要保存任何可能包含敏感信息的文件。

10.6.3 查找代码

到现在为止，我们已经学习了一些很好的从Google的大量索引中恢复有用信息的技巧。我们也看到了搜索工具不太精确的地方，我们通常要精确查询以得到更好的结果。使用正则表达式查找到那些对于我们而言最为有用的信息不是很好吗？尽管Google Search不能为我们提供这些，但是Google Code可以。访问http://www.google.com/codesearch进入Google的Code Search服务，如图10-30所示。

图10-30 Google Code Search

在我们要查找代码摘要以借用或者枚举普通漏洞时，代码搜索尤其有用。让我们来看一下怎么做。

打开Google Code Search界面，并且键入如下的查询：

```
echo\s*.*?PHP_SELF lang:php
```

注意，该语法与我们通常看到的有一些不同。这就是我们所说的正则表达式（regex），详情参见以下URL：http://en.wikipedia.org/wiki/Regular_expression。该正则表达式搜索会返回与图10-31所示的页面相似的结果。

图10-31 搜索PHP_SELF漏洞

让我们来进一步地了解正则表达式的作用。查询的第一部分会查找关键字回应（echo）。接着，我们要确定这里有没有空格（\s*）。之后的部分将确定在遇到最终的分隔符（.*?）之前都在搜索未定义字符数量。最后，我们将使用关键字PHP_SELF来终结查询。注意特殊参数lang。我们指定只查找PHP脚本。通常，查询会查找如下所示的内容：

```
echo $PHP_SELF
echo($PHP_SELF)
echo ($PHP_SELF)
echo $_SERVER['PHP_SELF']
echo($_SERVER['PHP_SELF'])
```

PHP_SELF的不正确使用得到了一个非常有名的XSS（跨站点脚本）漏洞。这个错误在PHP应用程序中非常常见。大多数开发者都假定PHP_SELF不由任何用户控制。事实上，它是由用户控制的，并且非常容易使用。以下是一个实例：

```
http://target/path/to/script.php/"><script>alert('xss')</script><!--
```

注意，我们将把附加的路径添加到包含符号“><script>alert('xss')</script><!—的script.php之后。由于PHP_SELF通常用来查找当前脚本的URL，所以它很有可能作为元素属性的一部分。

这正是我们使用">组合符号来摆脱附加元素的原因。我们使用<!—来结束，是为了弥补它的左部残缺。

接着，让我们来尝试另一个查询，但是这一次，我们将查找SQL注入漏洞（SQLI）：

```
mysql_query.*?_GET lang:php
```

该查询结果如图10-32所示。

图10-32 查找SQL注入漏洞

该查询以关键字mysql_query开头——mysql_query是PHP里的标准函数。接着，我们使用.*?序列查找未定义的字符数。最后，再查找用来标示HTTP GET参数的关键字_GET。通常，我们查找的SQL查询由$_GET控制。对基于$_POST的SQL注入攻击应用也可以采用相似的策略。记住，本章的实例仅是我们可以进行测试的实例的一小部分变体。Google Code Search是一个非常有用的可以在很多语种中查找漏洞的工具。

注意

我们可以使用Google Code Search在我们自己的项目中查找字符串。如果我们有一个巨大的数据集要分析，我们可以简单地将它上传到代码中，并且等待Google机器人程序将它找出来。接着，我们可以使用标准的正则表达式查询来查找我们最感兴趣的数据。

第11章 Google Hacking案例

11.1 简介

自尊的Google黑客会花上好几个小时在互联网上查找丰富的资源。通过一轮又一轮的搜索，他们会因找到清晰易读的、有意义的、精简的搜索结果而欣喜若狂。之所以我了解这些是因为我亲眼目睹了这一过程。作为Google Hacking Database（GHDB）和http://johnny.ihackstuff.com网页上的搜索引擎攻击论坛的创建者，我经常会为GoogleHacking社团提出的内容而惊愕不已。这表明传闻是真的——具有创造力的Google搜索可以泄露医药、金融、所有权甚至是机密信息。尽管政府颁布了法令、规章及保护法案，诸如HIPPA，且不断设置安全装置，这一问题仍然存在。信息仍会驱使网络中这一问题的发生，而这正好被Google黑客逮个正着。

为了强调这一威胁的重要性，我开始在在Blackhat和Defcon等安全会议上就Google攻击这样问题发表演讲。Googel Hacking，第一版，带来了一定的影响。不过并没有什么像“Google Hacking案例”（我那声名狼藉的Google攻击会议演讲中有趣的部分）那样引起骚乱。该案例对我而言份量并不大——除了我所看到的一些疯狂的Google攻击的屏幕快照外，再无他物。借了从我创建的有趣的Google搜索池中借用了一些东西外，我还借用了来自社区的许多搜索；我抓取了屏幕快照，并且每次只陈列一个快照，并配合乏味的评论。在每一次陈述案例的时候，我都设法引导观众轻视那些仅借助一个浏览器和一个搜索引擎的攻击者荒谬的效率。这很有趣，也很有效。每次演讲之后，人们都会就那些屏幕快照谈上好几个月。毕竟，这些是Google攻击者的劳动成果。那些照片代表了Google攻击威胁的重心之所在。

在此之后，我再将这些案例放入Google Hacking的这一版本中就没有问题了。本章保留了案例展示的快照的原始格式，并侧重通过图片而非文字来表达内容，因为这些图片本身就能够说明问题。本章中的一些屏幕快照已经过时了，有一些已经无法在网页上找到，不过这是一个好消息。这意味着世界上的某个地方，某个人（或许不经意地）从googledork（著名搜索引擎）水平毕业了，并且向更佳的安全状态靠近了一步。

不管怎样，我保留了很多过时的快照，以作为对那些有责任保护在线资源的人的提醒。它们作为这一威胁的证据极具渗透力——每个人都可能遭受该威胁的攻击，历史证明，几乎每个人都遭受了该威胁的影响。

因此，不用再那么麻烦，享受Google Hacking案例的这一印刷版本即可，这是我带给你的，同时也有Google Hacking社区的贡献。

11.2 低级信息

本节将讲述有关计算机信息的内容。这是有关技术信息的内容，低级信息的内容。我们将

要一同来看一下由Google黑客揭露的一些更有趣的技术。我们将从查看多种确实与在线无关的、除非你的目标是帮助黑客的工具着手进行。接着，我们将要查看开放的网络设备，并且打开应用程序，它们都不涉及任何真正的可以使用的攻击。

11.2.1 工具

任何自尊的黑客都有一个由他自行支配的工具基金，但是本节索要探讨的该工具的有趣部分是：它们是在线的——它们在网络服务器上运行，且允许黑客有效地试探对该网络服务器的搜索。为了让事情变得更糟，这些运行应用软件的网络都安装了聪明的Google查询。我们将从方便获取的PHP脚本（如图11-1所示）开始，该脚本允许网络访问者在互联网上ping任何目标。ping并不必定是一件坏事，不过为何要向匿名访问者提供该服务呢？

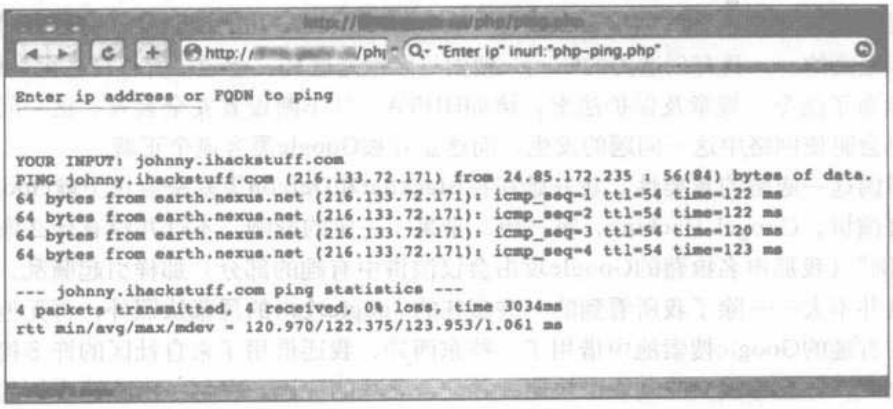


图11-1 Php-ping.cgi提供免费ping服务

与该ping工具不同，finger工具已经很久没有投入使用了。这一烦人的服务允许攻击者查询UNIX机器上的用户，允许枚举各种信息，如用户连接时间、起始目录、全名以及其他的信息。输入finger CGI脚本，尝试一下“网络化”（webify）这种不愉快的服务。如图11-2所示，一个设置良好的Google查询会查找到该脚本的安装信息，为Web访问者提供一个允许他们在远程计算机上查询该服务的finger客户端。

pings和finger查找都是相对善意的；大多数系统管理员甚至都不会注意到它们正在遍历他们的网络。相反，端口扫描（port scans）几乎从来不被认为是善举，猜疑的管理员（或者某款防御软件）都会关注端口扫描源。不过大多数现代的端口扫描软件提供了非常多允许隐蔽操作的选项，少许Google Hacking还可以做得更好。图11-3揭示了一个由Jimmy Neutron提供的Google搜索，该搜索会找到那些允许Web访问者允许对目标进行端口扫描的站点。

记住，以这种方式执行的扫描起源于Web服务器，而非攻击者。最具猜疑心理的系统管理员会奋起跟踪用此方法发起的扫描。当然大多数攻击者都不会放弃端口扫描的举动。他们更倾向于继续使用多种可以泄露目标真正位置的网络工具来刺探目标。然而，如果攻击者搜索一个类似到图11-4所示的Web页面（由Jimmy Neutron提供），便可以通过寄存在远程服务器上的WebUtil Perl脚本来引导各种网络刺探工具。这样一来，这个刺探工具又看来像是来自Web服务

器而非攻击者。

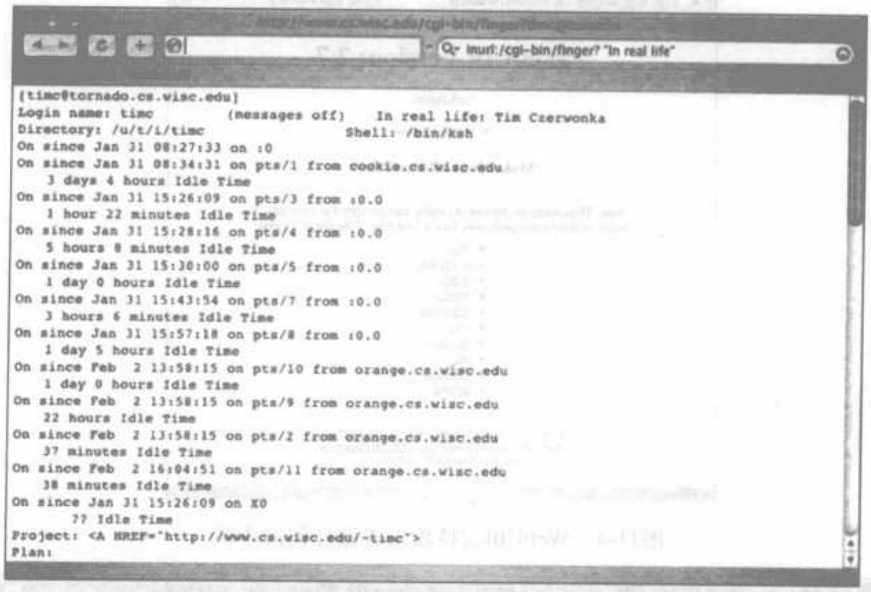


图11-2 Finger CGI脚本允许远程finger

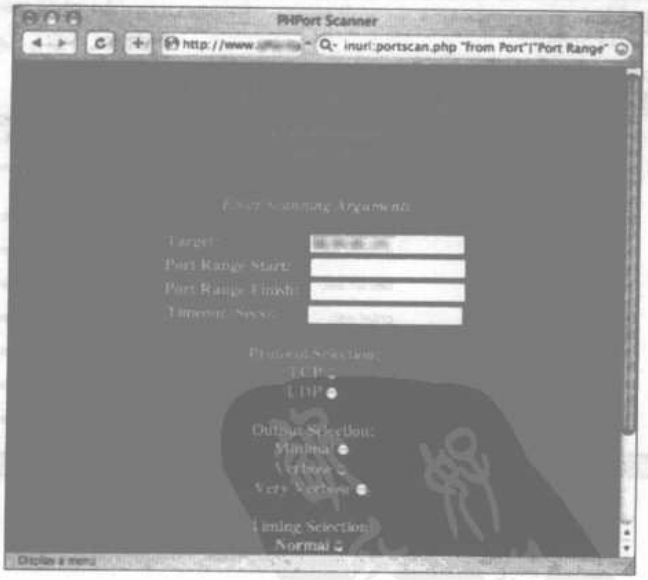


图11-3 PHPort Scanner——极好的基于Web的端口扫描程序

在图11-5中列出的Web页面（由Golfo提供）列出了某所学校的“学生注册”系统的名称、地址和设备信息。在界面点击可以泄露有关网络结构的更多的信息，以及与之相连的设备。通过合并到一个易读的界面并且使用Google搜索查找，该页面可以简化攻击者的搜索工作。

图11-4 WebUtil支持攻击者做任何的事情

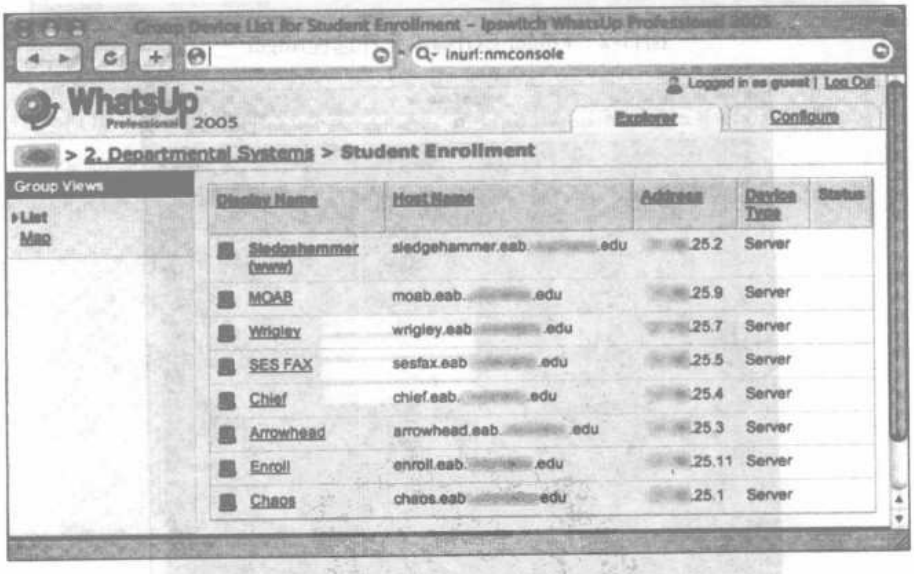


图11-5 WhatsUp状态屏幕为来宾提供了大量的信息

11.2.2 开放的网络设备

当你可以正确地定位到一个开放的网络设备或者单击进入一个开放的网络设备时，为什么还要攻击网络服务器或者网络设备呢？诸如由Jimmy Neutron提供的如图11-6所示的管理设备通常会列出多种设备的各种信息。

当m00d提交了如图11-7所示的查询时，我真地没有想太多。很显然，SpeedStream路由器

是一台家庭用户使用的轻量级设备，但是我很震惊的是，它们竟然在互联网上不受任何保护。我个人很喜欢点对点一览表中的按钮。今天，你想与谁断开连接？

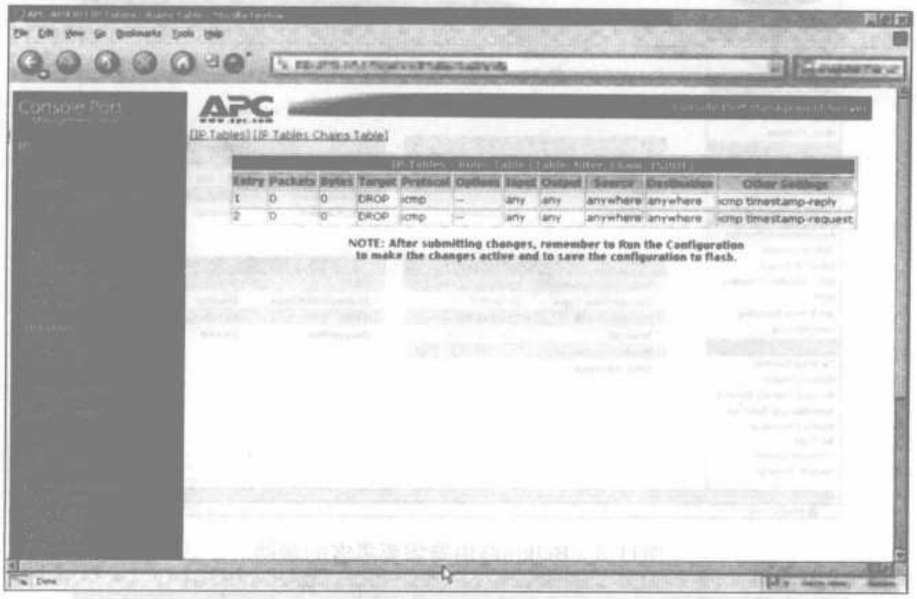


图11-6 开放的APC管理设备

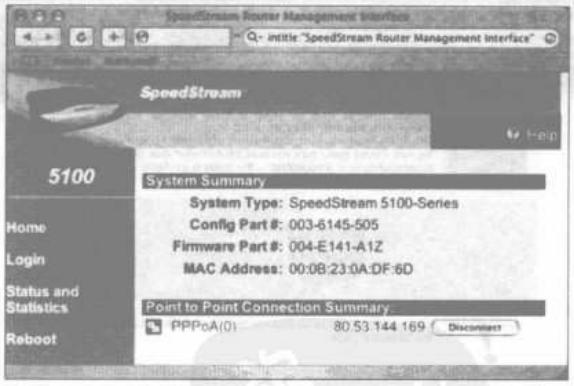


图11-7 开放的SpeedStream DSL路由器允许远程切断

Belkin是家庭网络设备中家喻户晓的品牌。它们拥有易用的基于Web的管理界面，因此，最终，如图11-8所示的页面会被Google抓取到。就算是没有登录凭证，页面也可以泄露大量的对于潜在攻击者而言非常有用的信息。看到该页面的Features部分我忍不住笑了。虽然防火墙启用了，但是无线接口却大打开着而且没有加密。作为一个有社会公德的“黑客”，我的第一本能是在该访问点上启用加密——试图保护这些可怜的用户。

Milkman为我们提供了如图11-9所示的查询，该查询可以挖掘出Smoothwall个人防火墙的配置页面。这是关于Google攻击某人的防火墙时出现的问题。

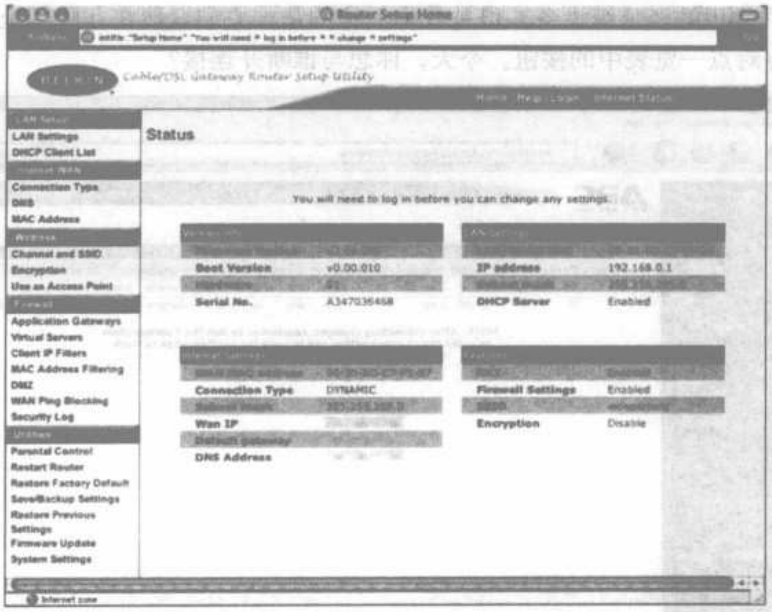


图11-8 Belkin路由器需要黑客的帮助

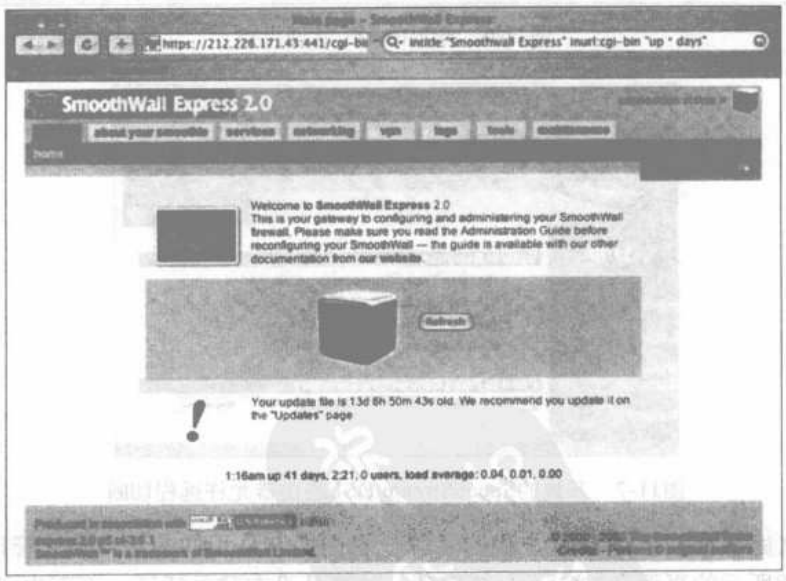


图11-9 Smoothwall防火墙需要更新

正如Jimmy Neutron在下面两幅图片中揭示的那样，甚至是那些大牌的设备商如Cisco也会不时地揭Google缓存的短。尽管它相貌平常，但是如图11-10所示的交换界面还是留给了我们一些想像的空间——所有的配置及诊断工具都列在了主页面上。

Cisco的第二幅屏幕快照看起来很像是Cisco小丑。我不知道为什么，但是Cisco命名法则让我想起了好莱坞的下三滥电影。我们几乎都可以听到一个合成的计算机声音："欢迎来到Level 15。"

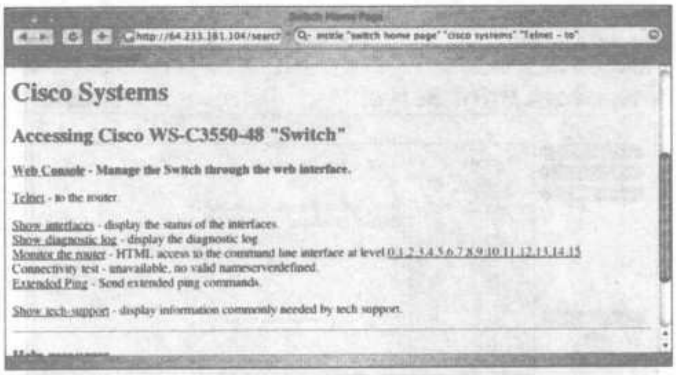


图11-10 开放的Cisco Switch

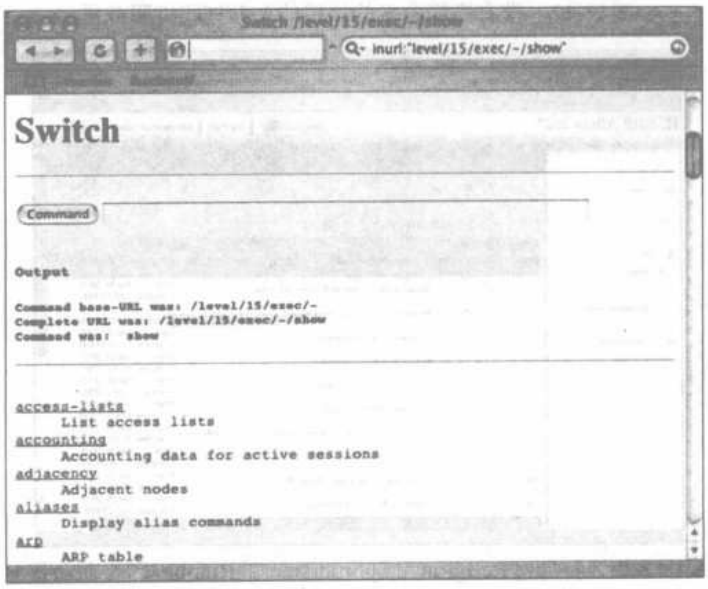


图11-11 欢迎来到Cisco Level 15

图11-12中显示的搜索（由Murfie提供）搜索到了Axis网络打印服务器的界面。大多数打印机界面真的都很让人讨厌，但是这一个却尤其引人注目。首先，这里有一个名为configuration wizard的按钮，我很肯定它会加载一个配置向导。接着，这有一个标注为Print Jobs的便利链接，它可以列出打印任务。在你还没有猜到的情况下，Google Hacking有时也会留下少许想象空间。

打印机并非是完全无趣的东西。先来看一下图11-13所示的Web Image Monitor。我特别喜欢关于Recent Religion Work的文档。这是一个非常让人自豪的追求，除了它与有关Aphrodisiacs文档结合外。我真的希望两个文档没有任何关联。之后，这些天我都没有碰到什么令人惊讶的事。

CP有一种我认为很可笑的查找Google攻击的方法，图11-14也不例外。是的，这是一个基于Web的连接到市政饮水器的界面。

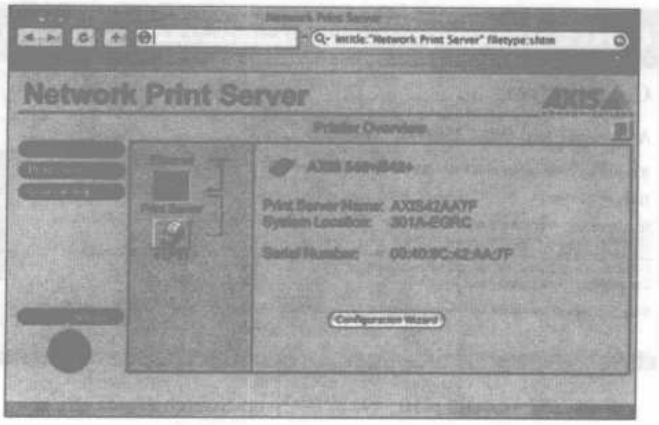


图11-12 带有模糊含义的按钮的Axis打印机服务器

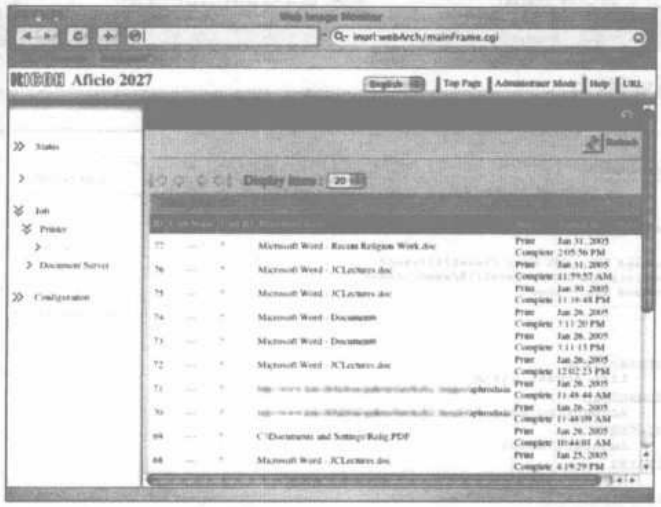


图11-13 打印过Religion和Aphrodisiacs的Ricoh打印机服务器

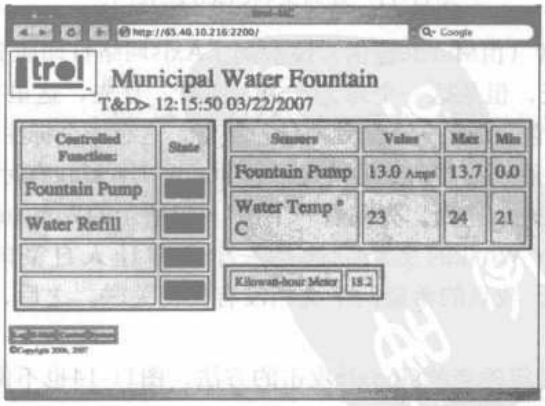


图11-14 出于玩笑和利益攻击饮水器

在观察水温波动几秒钟后，你肯定想单击Control（控制）链接来看一看它是否能真正控制市政饮水器。正如图11-15所揭露的那样，它确实可以远程控制市政饮水器。

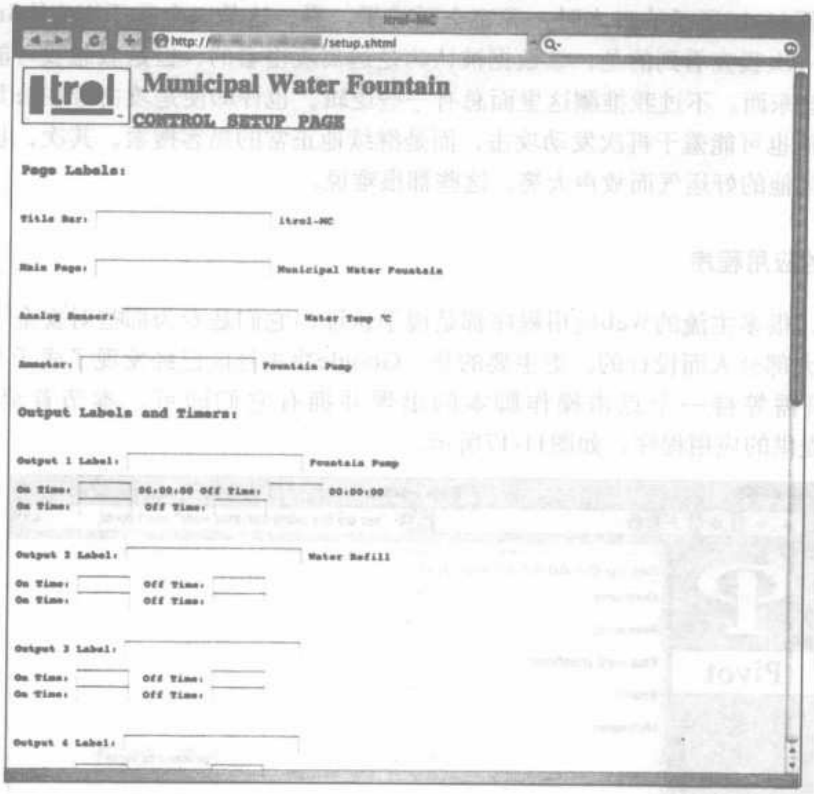


图11-15 更多的关于饮水器的趣事

如果你是碰巧闯入了这些界面，那么给你一个小小的忠告。不要更改水存储系统的功率。我想这一定会构成恐怖行为。

接下来看一个更传统的网络设备，先来看一下图11-16中捕捉到的屏幕快照。

图11-16 Acid上的IDS管理器

到现在，我已经从事安全工作多年了，我对这个行业的任何一个领域都不是非常在行。但都稍有了解，且有一件事我敢肯定，那就是安全产品是用来保护信息的。这是理所当然的。但是当我看到如图11-16所示的日志时，我完全疑惑了。看，这是一个基于Web的Snort侵入检测系统界面。上一次我查看到的是，该数据被认为是远离攻击者的，但是我想我可能遗漏了一封E-mail或者某些东西。不过我推测这里面总有一些逻辑。也许即使是攻击者在公共网页上看到了它的漏洞，他也可能羞于再次发动攻击，而是继续他正常的黑客搜索。其次，也许他和他的攻击伙伴仅会为他的好运气而放声大笑。这些都很难说。

11.2.3 开放的应用程序

相对而言，很多主流的Web应用程序都是傻子认证，它们是专为那些对安全了解甚少而只懂点击操作的大部分人而设计的。更主要的是，Google攻击社区已经发现了成千个开放的在线应用程序，只需等待一个点击操作脚本的出现并拥有它们即可。本节首要讲述的是由Shadowsliv所提供的应用程序，如图11-17所示。

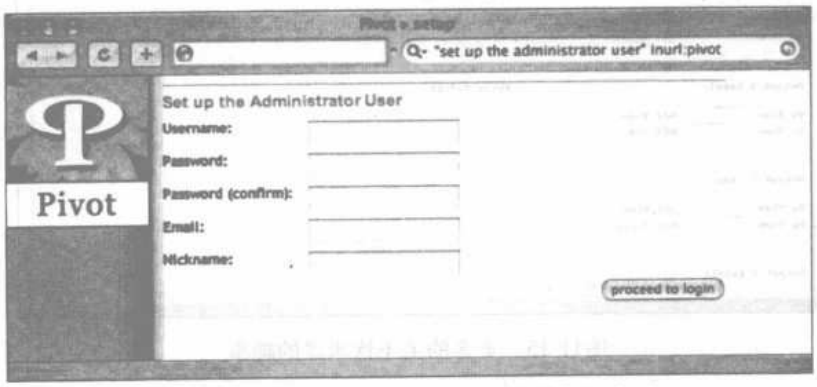


图11-17 复杂的Pivot攻击需要正确填写5个域

坏消息是，如果黑客可以断定需填入的内容，那么他将拥有自己的Pivot网络日志。好消息是，绝大多数有经验的攻击者都不会去管这一站点，他们认为任何软件将此列为不被保护的范围都必定是一个陷阱。攻击可以被简化成点击操作这么简单的事，实在让人悲痛，不过图11-18中Arrested的查询表明拥有整个网站会是一件相当容易的事。

这比开放的Pivot安装要少填一个域，该配置页将创建一个PHP-Nuke管理者账户，并允许任何访问者下载内容到该页面上，就像是他们自己的一样。当然，这会带有一些网络访问者的恶意。能够确定的是，他或她正在网址上创建一个并不属于自己的管理者账户。然而，图11-19中所显示的页面文本有一些不明确。

页面中间的加粗部分内容让我崩溃。我能够想象哪个家伙可怜的祖母进入了该页面并大声朗读这一段文字的情景。“考虑到安全因素，最好的方法就是单击此处马上创建特级用户。”我的意思是，哪个正常的人会放弃出于安全因素考虑而采取的举措呢？因为所有的祖母都知道，她或许避免了恶意攻击并拯救了世界……通过攻入某个可怜的傻子的PHP-Nuke安装程序。

图11-18 PHP-Nuke攻击需要正确填写4个域

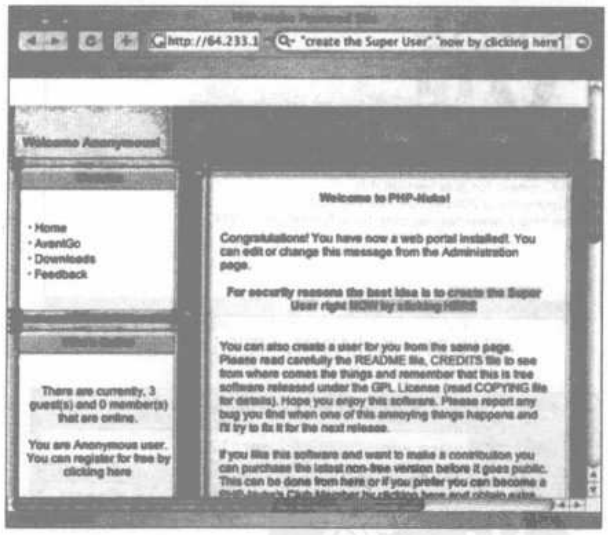


图11-19 “基于安全原因”攻击该PHP-Nuke安装

好像拥有一个网站还不够酷，图11-20（由Quadster提供）向我展示了一个作为根用户登录的phpMyAdmin安装，它提供了对MySQL数据库的自由访问。

有了网站安装和SQL数据库，Google黑客想要最终控制系统也就成了自然而然的事情。VNC安装提供了系统键盘和鼠标的远程控制。图11-21，由Lester提供，显示了来自RealVNC的基于Java语言客户的查询。

然而，搜索客户只是方程式的一部分。攻击者还需知道VNC服务器的地址、端口以及（任选的）密码。如图11-22所示，该Java客户本身往往会在就近弹出的窗口中提供了该方程式的三分之二。

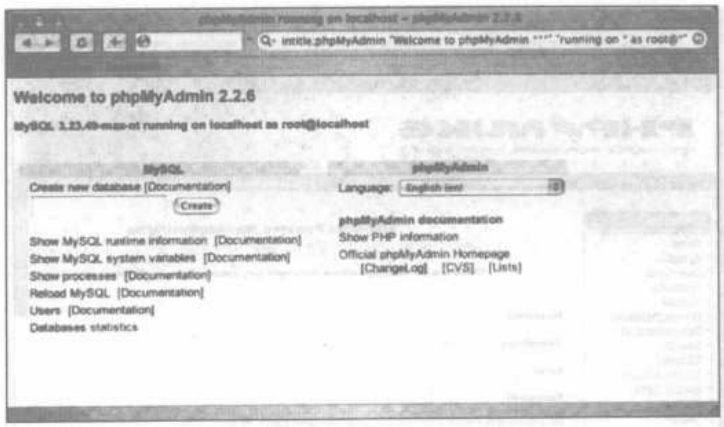


图11-20 用做案例模型的开放式phpMyAdmin-MySQL

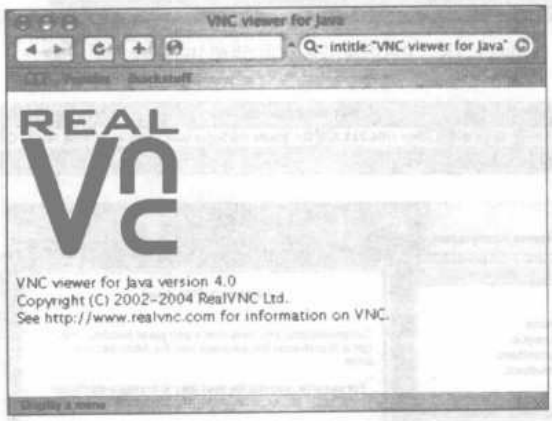


图11-21 攻击VNC，获取远程键盘

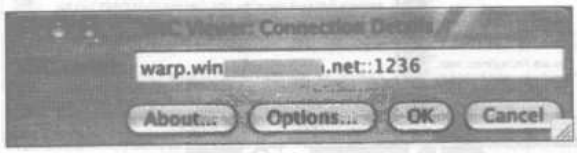


图11-22 与搭配的“薯条”一同递呈的VNC

如果幸运的黑客无意中发现了一个未设置密码保护的服务器，那么他将会为明确应该单击上述链接窗口中4个按钮中的哪个而犹豫不决。这有一个小提示：不是Cancel（取消）键。

当然，在未设置密码的情形下运行是非常愚蠢的事情。不过要记住密码也是一件困难的事，很明显软件供应商注意到了这一问题，所以提供了如图11-23所示的的密码提示。

在登录弹出窗口中公布默认的用户名/密码组合简直就是发疯的举措。遗憾的是，这并非只是一个孤立的事件。请看由Jimmy Neutron提供的图11-24。你能猜出默认的密码吗？

要想进一步提升黑客级别，需要花费一定的功夫。请看图11-25所示的用户屏幕，由Dan Kaminsky提供。

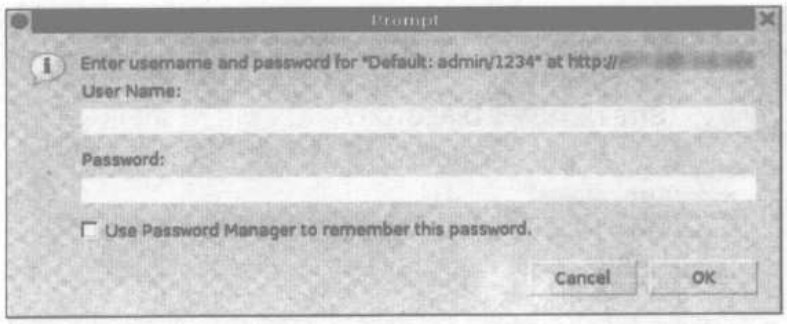


图11-23 方便的密码提示器，以防黑客忘记

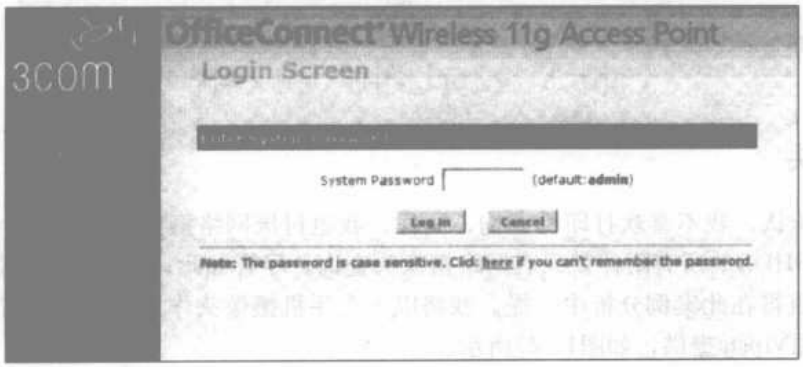


图11-24 如果你还不能猜出该默认密码，那么你就太失败了

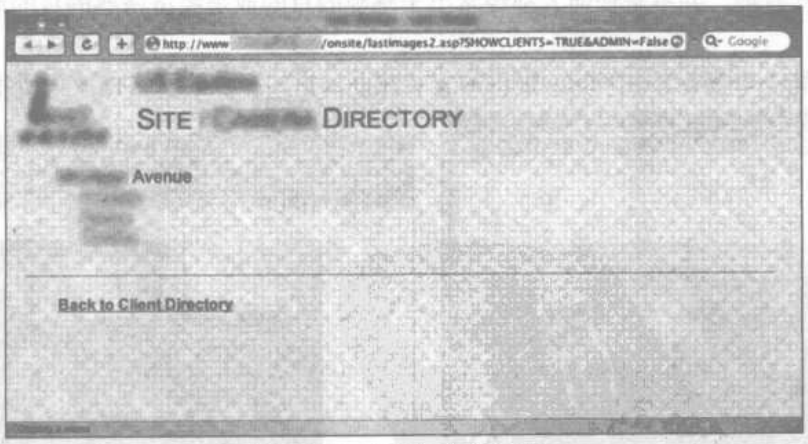


图11-25 欢迎来到来宾界面

如果仔细观察，你会发现URL包含了一个名为ADMIN的特殊域，该域被设置成False（错误）。像黑客那样去思考一下，并设想你如何才能进入该页面的管理员界面。图11-26列出了受损的界面。

单击发亮的Exit Administrative Access按钮结束。通过将ADMIN域的值更改为True（真），该应用程序会让我们进入了管理员访问模式。我保证攻击真得很难。

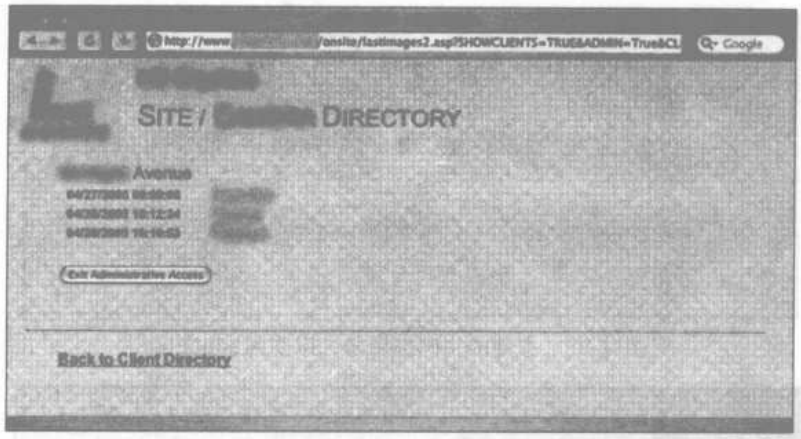


图11-26 通过URL修补进入管理者界面

11.3 摄像头

我不得不承认，我不喜欢打印机查询，同样，我也讨厌网络摄像头（Webcam）查询。曾有一段时间，GHDB的所有附件就只有网络摄像头查询。尽管如此，有些网络摄像头搜索还是十分有趣的，值得在此案例分析中一提。我将以一个手机摄像头作为案例模型着手进行讲解，这个案例模型由Vipsta提供，如图11-27所示。

这是一张看上去相当严肃的车辆撞车事件的有趣照片，除此之外，Google搜索摄像头手机图片网站也同样有趣。并不是任何人都经常出于感观的目的或是经济诱惑的目的而使用这种信息的。呃哼。

继续查看图11-28所示的由Klouw提供的安装在办公室的开放的网络摄像头所拍摄的图片。

图11-27 Google搜索交通惨案

图11-28 Remote Shoulder Surfing 101

这确实是一个有趣的网络摄像头。它不仅可以展示办公室内的一切活动，而且它看起像是

专程为允许远程肩窥而设计的。黑客习惯于走出户内来参与这一经典的运动。目前，他们所需要做的只是进行少许Google搜索。

由Jimmy Neutron提供的图11-29，展示了美国军用核潜艇的I.T.防卫措施。

图11-29 非真实的美国军用核潜艇

好吧，这不是真的。它可是只是一个核反应堆或者强信号栅极控制中心，或者甚至是哥伦比亚（马里兰）的毒枭仓库。这也许是因为我阅读了太多的Stealing The Network之类的书的缘故。不管怎么样，这依然是一个很酷的查找结果。

不过，由JBrashars提供的图11-30还是很清晰的。显然它是一个停车场摄像头。我不明白具体为什么，摄像头会对准残疾人泊车位，但是我猜想这可能报道了准残疾人泊车位滥用的问题。设想一下保安现场目睹了CIO停车时的欣喜之情，他从自己的敞篷车里跳出来，然后跑进了大楼。这些都是保安的传奇故事。

图11-30 残疾人泊车位Gestapo摄像头

WarriorClown向我提供了图11-31中所示的图片搜索。该图片展示了码头的情形，以及一片放有炸药容器的区域。

尽管一开始看上去很枯燥，但是网络摄像真的很有趣。点击图片右上方的有趣按钮退出。我非常肯定，只要点击该按钮，就会有激光束射向该白色的炸药容器，并造成最大程度的损害，不过这只能进行一次操作——除非你将其设置成再生（respawn），那么它们就会自动恢复。噢，等等。这只在Halo 3中的Forge模式中才会出现。好吧，所有这些网络摄影都开始让我失去理智了。为了控制我的想象力，我列出了相当简明的安全摄像头视图，如图11-32所示。

图11-31 远程爆炸容器的快乐

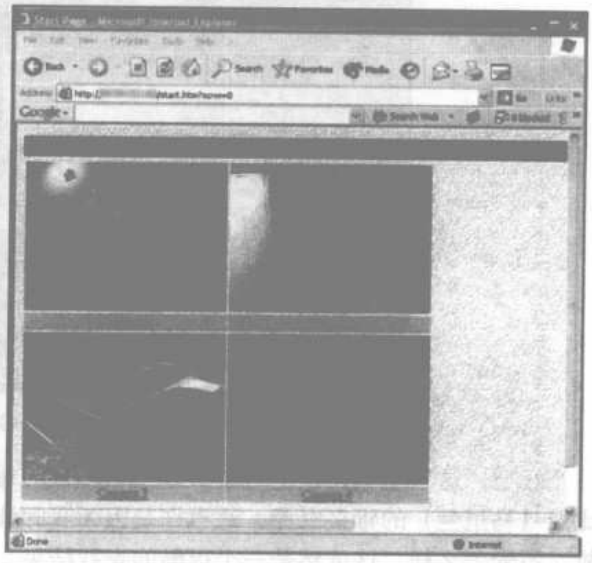


图11-32 开放的网络“安全”摄像头

我不可能是唯一认为将开放的安全摄像头所拍摄的内容放在互联网是精神错乱表现的人。当然，这一直出现在好莱坞的电影当中。看上去似乎受雇的黑客的第一份工作就是接入该视频监督中心。不过电影将这一过程编排得很复杂且具有一定的技术含量。我从未看到一个好莱坞黑客使用Google来攻击安全系统。其次，那样也不会像使用纤维光学摄像机、剪钳和鳄鱼夹那样酷。

继续，图11-33所示的搜索（由JBrashars提供），为开放的Everfocus EDSR applet提供了不少的点击量。

Everfocus EDSR，是一种带有网络界面的多信道的数字视频记录系统。它是一种相当好的监视产品，同样，它也将密码保护设置为默认状态，如图11-34所示。

遗憾的是，正如某一匿名供稿者所透露的那样，出厂前默认的管理者用户名和密码为很多这类系统提供了准入途径，如图11-35所示。

一旦进入，EDSR applet则会提供多个实时视频输入以及任何之前的历史实况记录视频。再次，如同上演好莱坞奇迹一般，不存在任何黑客的聪慧。

EDSR不是唯一的被Google黑客锁定的多信道视频系统。如Murfie所展现的那样，通过搜

索I-catcher CCTV，我们会找到很多类似图11-36所提供的系统。

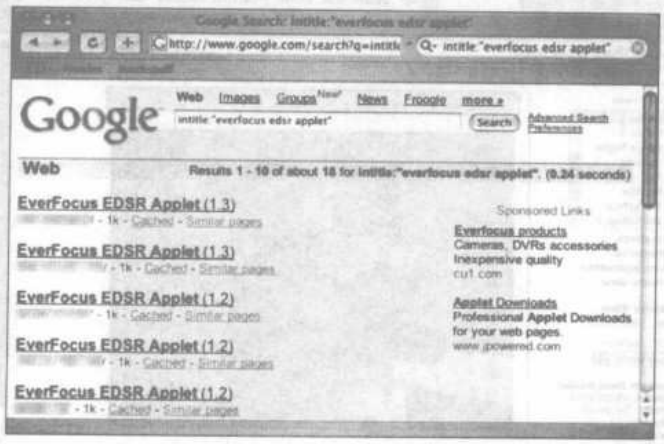


图11-33 EDSR看上去足够的驯服

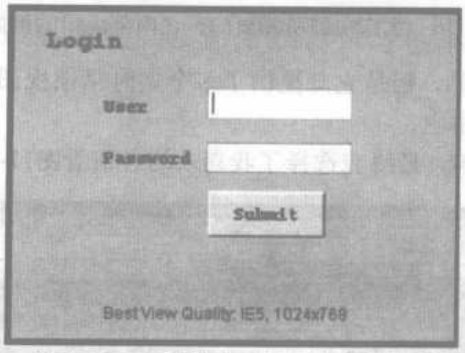


图11-34 密码保护：安全的金本位制

图11-35 欢迎来到监管中心

图11-36 家庭主妇所需要的，在内部应用的摄像头

尽管该界面看上去很简单，但是它却提供了多个实时摄像视图，包括其中叫做“Woodie”的，就个人而言，我不敢点击它。

这些摄像头都很有趣，不过最终我选择了我喜欢的。请看图11-37。

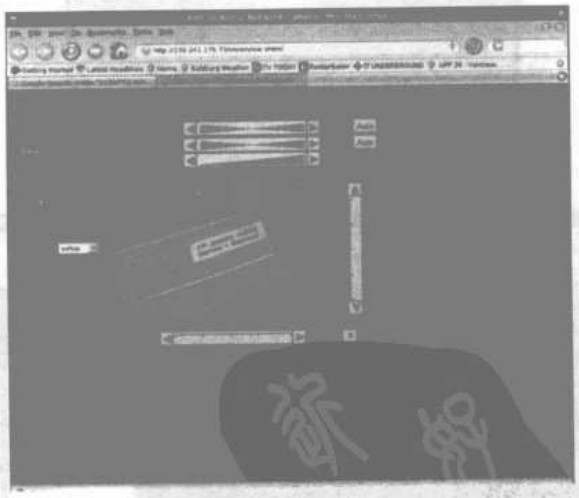


图11-37 肩窥遇到网络摄像头、遇到密码不干胶条

这种摄像头为网络访问者提供了开放式访问途径。它被放置在一个计算机实验室，其远程控制能力允许匿名访问者四处偷窥，平移镜头并拉伸镜头以接近想要获取的内容。这不仅允许大量的肩窥，而且上面屏幕快照中的不干胶条已经让我从椅子上掉下来。它列出了该实验室的在线FTP服务器的用户名和密码。不干胶条所列举的用户名和密码已经够糟了，不过我想知道是谁出的主意，要把开放的网络摄像指向它们。

11.4 电话设备

我从来没有做过真正的电话盗用者（电话黑客），不过感谢Google强大的搜索功能，我不需要有这类经历就进入了这一阴暗的操作队伍中。正如JBrashar的搜索结果所示，图11-38，VOIP服务产生了大量新的网络电话界面。

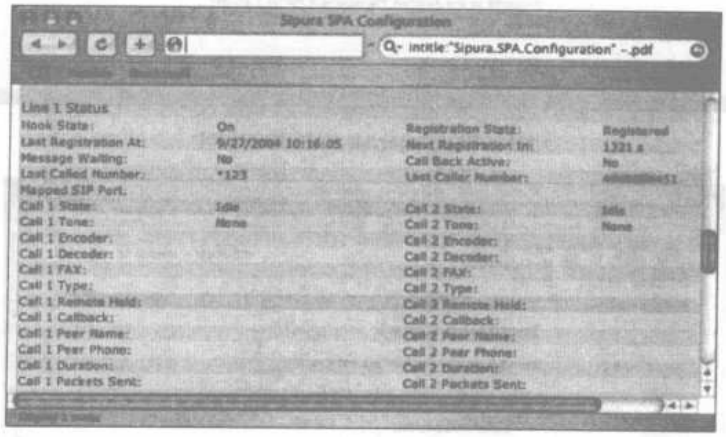


图11-38 Google攻击所依赖的电话系统

我认为攻击者仅使用Google就可以获得最后呼出和最后呼入电话号码等通话记录信息很有意思。通常，Sipura SPA软件能够更好地保护该信息，不过这一特殊的安装进行了不适当的配置。另外，通过点击网络界面上的链接，会泄露更多的技术信息，如图11-39所示。

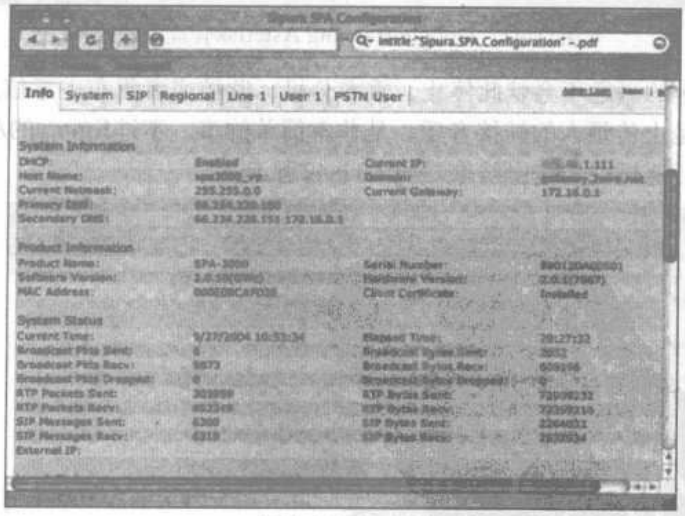


图11-39 Redux

VOIP设备有很多，以致无法全部覆盖，不过新的VOIP服务块的确是Asterisk。在核查该Asterisk管理入口文档之后，Jimmy Neutron发现了如图11-40中所示的有趣的搜索。

黑客可以从这个开放的入口更改Asterisk服务器，包括转发来电，如图11-41所示。

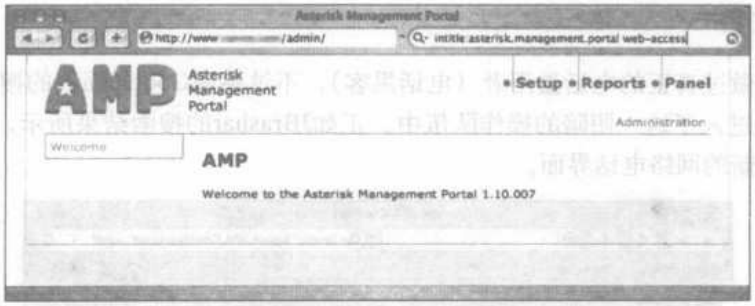


图11-40 Asterisk，VOIP的国王

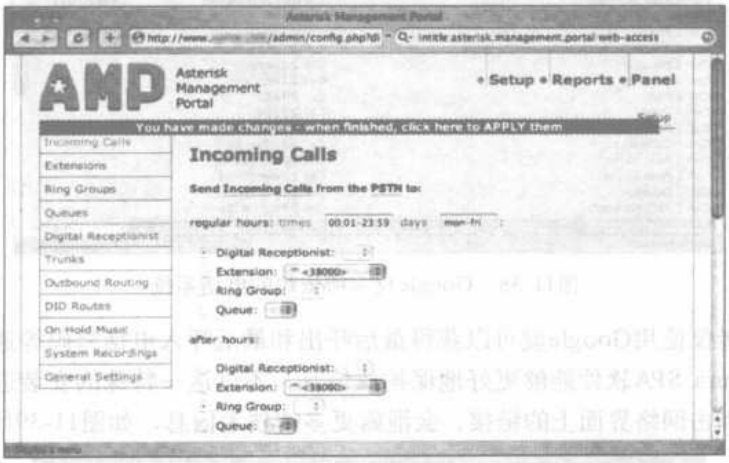


图11-41 Google Hacking Asterisk管理入口

遗憾的是，黑客的兴趣不会就此停止。重布分机、监控或者重布主意信息、启用或者禁用数字接线员，甚至是上传烦人的转接音乐，是非常简单的事。不过Jimmy的Asterisk VOIP挖掘并不仅限于此；之后他提交了搜索结果，如图11-42所示。

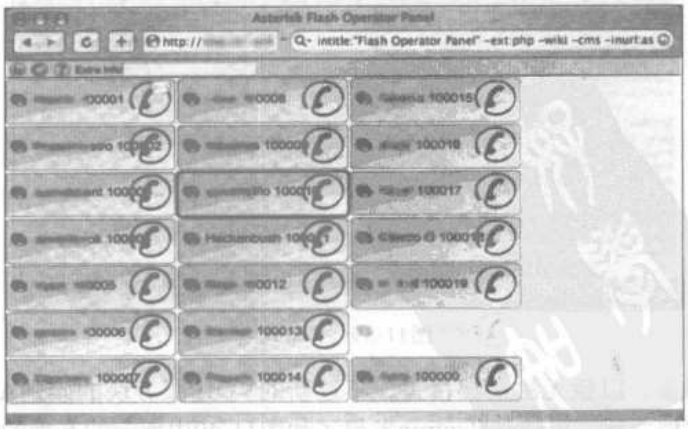


图11-42 Redux. HackenBush. Heh.

这个基于Flash的操作面板提供了近似权能的访问，我们再次发现，该界面是对任何一个访问者开放的。

随后，Yeseins提供了如图11-43中所示的有趣的搜索结果，其定位于视频会议管理系统。

图11-43 攻击视频会议系统

该管理系统允许网络访问者连接会议电话、断开会议电话连接及监控会议电话，为与会者抓取快照，甚至是更改line（路线）设置，如图11-44所示。

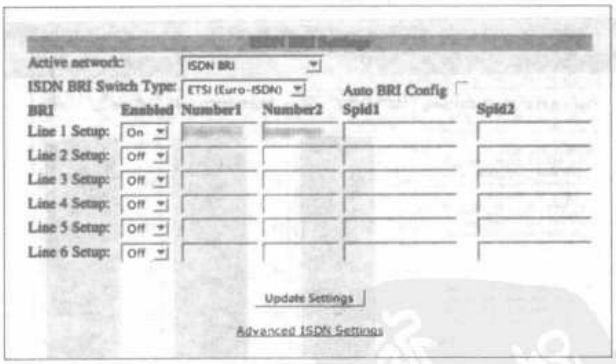


图11-44 更改视频会议路线

恶意的黑客甚至会更改系统名和密码，锁定合法管理者使其无法进入自己的系统，如图11-45所示。

除了所有我们看过的新近流行的网络界面，Google攻击还填补了与旧的系统之间的代沟，如图11-46所示。

该前端专为在旧式的PBX产品上安置新外观而设计，不过客户安全似乎已经是后事了。请注意，该界面要求用户“退出”该界面，显示该用户已经登录了。另外还请注意标有“Start Managing the Device（开始管理设备）”的隐藏按钮。在开始Google搜索后，所有恶意的黑客

不得不做的事就是弄清应该点击哪个按钮。多么让人难以置信的困难的任务！

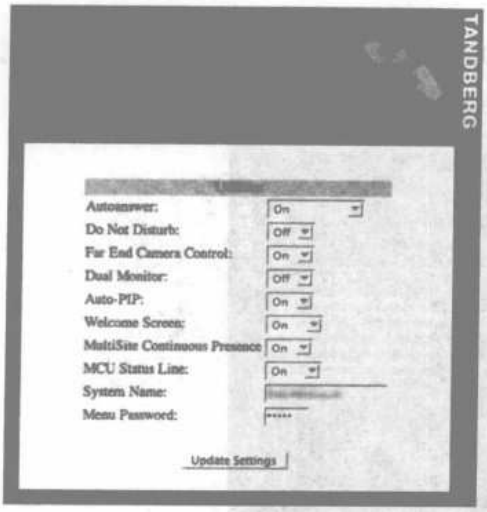


图11-45 视频会议系统

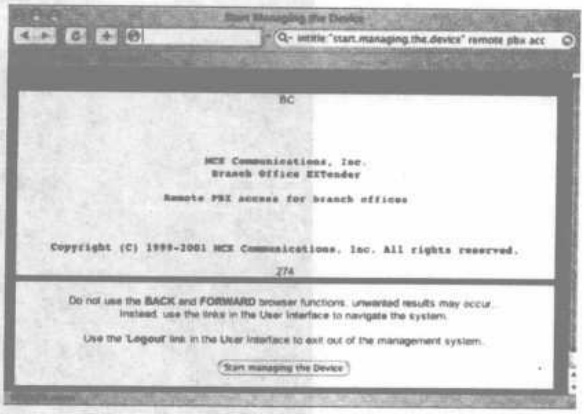


图11-46 Google Phreaking旧学院派样式

11.5 电源

当我谈及使用Google攻击电源系统时，很多人都会感到怀疑。大多数人肯定在想我是在说诸如图11-47所示的由Yeseins提供的UPS系统。

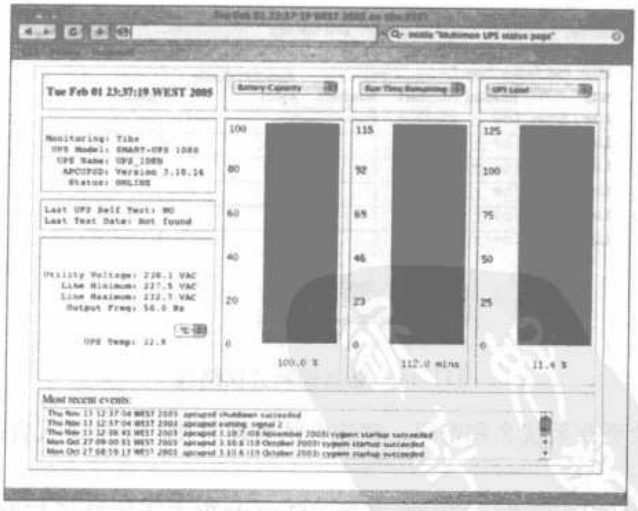


图11-47 怎么啦

这是很巧妙的Google查询，但是它只是一个不间断电源（UPS）监视页面。这会很有趣，但是正如Jimmy Neutron在图11-48中展示的那样，这里会有许多更有趣的电源攻击机会。

AMX NetLinx系统专为控制电源系统而设计。上面的图像看起来像是在暗示Web访问者可

以控制戏院、家庭和居住者的主卧室的电源。问题是Google搜索只找到了少许结果，而其中的大多数结果又受到了口令保护。作为另一种备选方式，Jimmy提供了如图11-49所示的搜索。

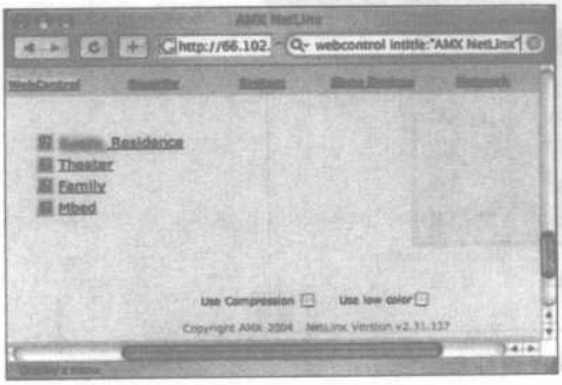


图11-48 用做案例模型的卧室攻击

图11-49 口令是俏皮话，尤其是默认的口令

尽管该查询会得到一个很长的受口令保护的站点列表，但是很多站点仍旧会使用默认的口令来提供如图11-50所示的对控制面板的访问。

图11-50 Google攻击电灯插座？原来如此

控制面板在那些名为Power和Restart有趣的按钮旁列出了电源插座，，即便是最笨的黑客也能明确地弄清楚。这个界面的问题在于它并非那么有趣。翻阅那些未命名的电源开关肯定会让黑客感到无聊，除非他也毫无疑问地找到一个开放的能观察这一趣事的网络摄像头。如图11-51所示的搜索似乎能解决这一问题，其对每个设备进行了命名以便更方便地引用。

当然，即便是大多数恶意的黑客都会认为掌握其他人的圣诞灯光（Christmas lights）是一件很粗鲁的事，但是并不会有正常的黑客会抵制如图11-52所示的开放的HomeSeer控制面板。

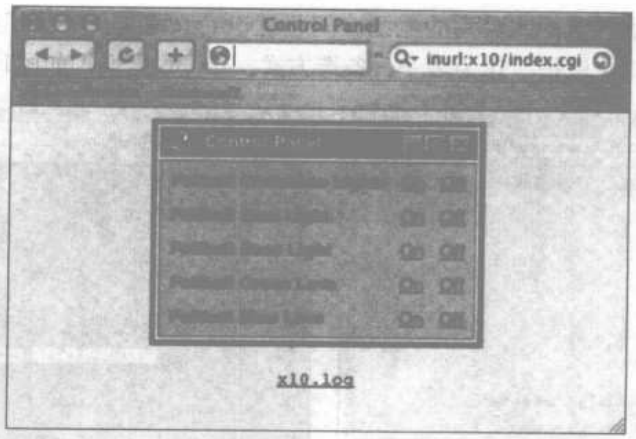


图11-51 远离圣诞灯光

图11-52 Bong Hacking

HomeSeer控制面板让电源攻击也变得有趣了，它为每个控件，还有On、Off以及可用元件的滑动开关列出了说明。这个列表中的诸如Lower motion和Bathroom的某些元件非常有趣。然而最有趣的一定是Electric Bong。如果你是负责逮捕该系统所有者的Secret Service（秘密服务）会员，那么我会建议你在闯入其住宅之前先采取Google袭击。先是让灯光变暗，再接着控制运动传感器。最后但并非最不重要的是，打开electric bong以防你其他电荷无法穿入。

11.6 敏感信息

敏感信息是一个专业术语，本节包括在Google上冲浪时发现的各种敏感信息。我们将从使用如图11-53所示的由Jorokin提供的VCalendar搜索开始。

图11-53 让我来查看他们的日历

至少，这些日历文件很可能都是为了共同使用的目的创建的，但是由Digital_Revolution提交的Netscape历史文件不可能如此，如图11-54所示。

图11-54 IBM的热链接

对于初用者，文件包含了用户的POP E-mail用户名以及编码后的密码。接着，他的URL历史记录会存在问题，该历史记录不仅包含倍受推崇的IBM.com还包含不那么有名的hotchicks.com，我非常确定后者是NSFW（即Not Safe For Work，不安全）。

下面来看一个由Harry-AAC提供的MSN联系人列表，如图11-55所示。

该文件列出了在某人联系列表中找到的联系人姓名和E-mail地址。最多，这个文件是垃圾邮件发送者的理想目标。这里真的不缺E-mail地址列表、电话号码列表以及Web上的更多资料，但是最让人惊讶的是多少文档包含此类信息——该信息为共享信息而创建。先来看一下如图

11-56所示的屏幕界面，该界面由CP提供。

图11-55 想盗窃我的好友

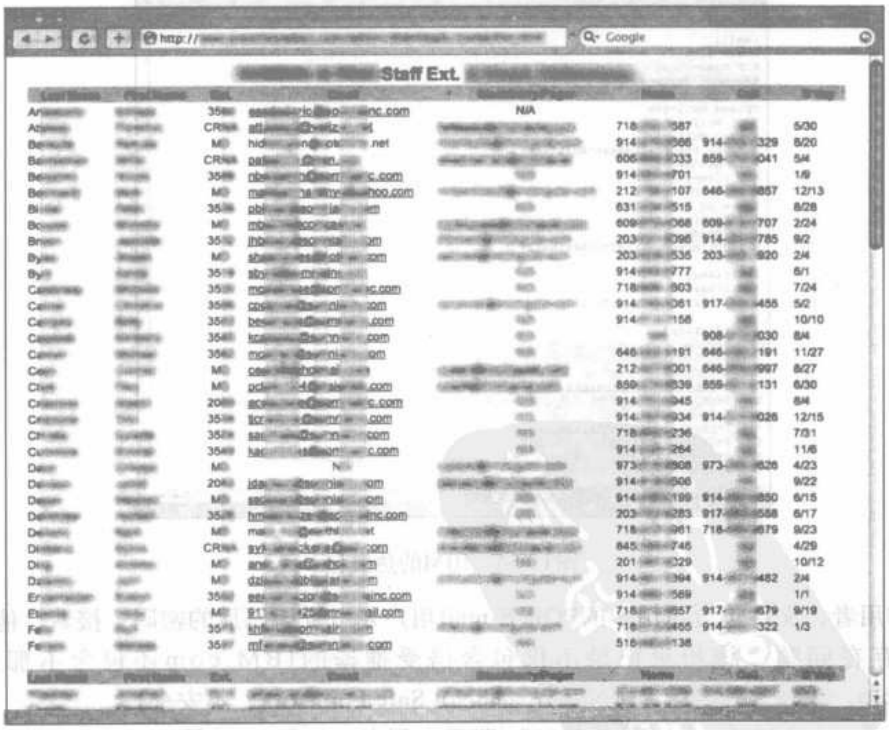


图11-56 给所有职员打电话和发E-mail并祝他们生日快乐

这个文档就是职员名录，仅为内部使用创建。唯一的问题是它是在公共Web站点中找到的。虽然这看似并不会构成十分严重的隐私信息，但是如图11-57所示的搜索（由Maerim提供）还

是揭露了少量的敏感信息：口令。

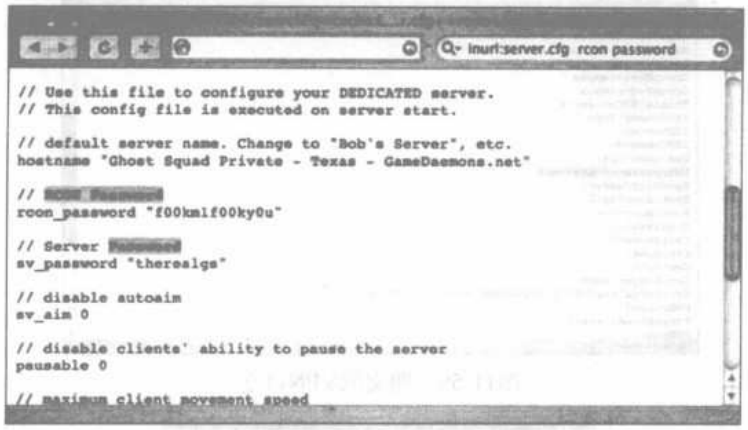


图11-57 我认为RCON口令是用希腊语编写的

这个文件为Ghost Squad的私有Counterstrike远程管理控制台列出了明文口令。你不妨向任意一个CS玩家询问这会带来什么麻烦。但是攻击一个游戏服务器相当乏味。不过，先来看一下由Barabas提供的图11-58。

图11-58 已编码的VPN口令

该文件列出了Cisco虚拟网（Virtual LAN，VLAN）的信息以及已编码口令。比泄露你的VLAN的已编码的口令更糟的一件事就是泄露你的VLAN的明文口令。只要查询就会收到。先来查看一下由Barabs提供的图11-59。

是的，这是嵌套在大学的配置文件中的明文口令。但是有用的口令可以在任何场合找到，例如由MBaldwin提供的如图11-60所示的Windows自动安装文件。

该文件也泄露了安装软件的产品密钥，该密钥可以被反复用于非法软件的安装。最后但并非最不重要的是，让我们来看一下由CP提供的图11-61。

图11-59 明文的VPN口令

图11-60 安装之前的Windows安装文件

图11-61 嗨，我能获取你所有的Web口令吗

该文档列出了不同Web站点的用户名和密码。文档存放在Web站点上，大概可能允许所有人轻易地对它进行远程访问。然而，在某些情形下，文档的位置是公开的，Google可以找到它。记住，公共的Web站点通常都是公开的。在将公开的数据与私有的数据组合在一起前，一定要慎重。

警务报告

从我的理解来看，大多数警务记录都是公开记录。因此当我看到类似图11-62所示的警务报告一点也不惊奇。

图11-62 警务报告是公开的记录

但是，当我找到一份与图11-63类似的警务报告时，我很怀疑张贴出未经筛选的警务记录的行为是否正常。

该警务报告记录了盗窃女性钱包的细节。问题是女性钱包的内容列举得太过于详细了，包含她的“维多利亚的秘密”账号！这不是唯一一个能在Web站点上找到的详细的警务报告。图11-64展示了另一个更容易泄露的报告。

这个报告详细地记录了另一件小型的盗窃案。这次还列出了被盗的Visa信用卡和Master信用卡的账号。在报道了卡片被盗后，很可能所有的卡片都会被取消，但是如图11-65所示的警务报道还列出了个人号码，这些号码不那么容易更换。

在本例中，不只受害人的驾驶证号被公布了，他们的社保号也与他们妈妈的驾驶证号一同被公布出来，所有这些都在公共的Web站点上公布了出来，对于身份证盗窃者来说这真是太理

想了[⊖]。

图11-63 这意味着你的“维多利亚的秘密”账号信息也是如此

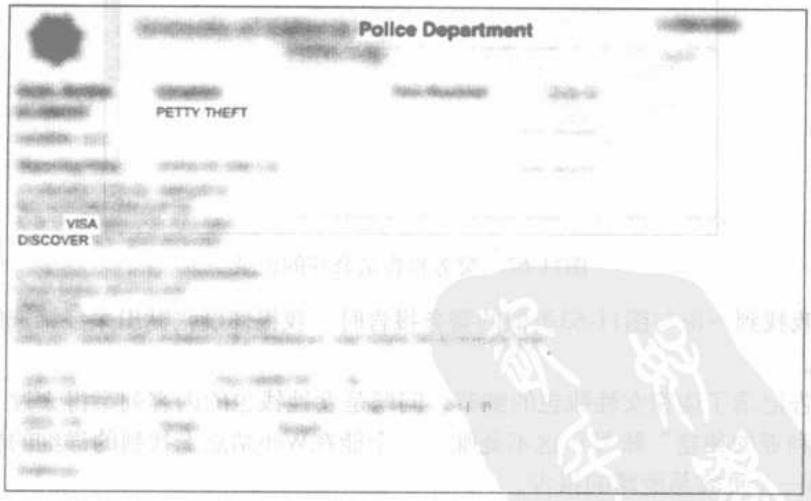


图11-64 被抢了两次，这要感谢公开的警务报告

⊖ 在此，我们处境显然非常尴尬，因为这里真的存在很危险的搜索。这里的所有认证信息以及接下来的搜索都进行了模糊处理，所有可能会引申出新的Google查询的信息也都被删除了。此外，本章中提供的大多数敏感文档已经从Web中删除。

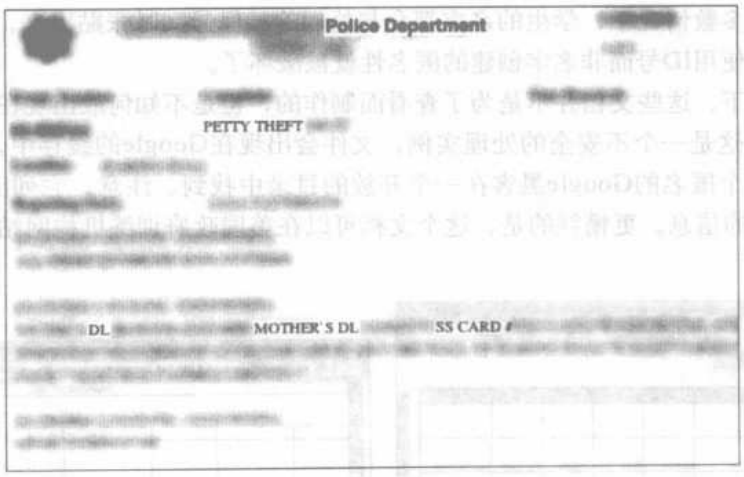


图11-65 有关三次被抢或者有关“妈妈，我有一个坏消息”的警务报告

11.7 社保号码

社保号码（SSN）是美国公民拥有的最敏感的信息。即使一个外行的罪犯也可以使用一个被盗的SSN来开设一个银行账号，设置信用卡的最高限额以及其他的信息——所有的均是以受害人的名义进行的。本节中，我们将要看一下几种网上登出个人SSN号的情形。就像是本书中对其他的敏感搜索的建议的那样，可以采用迷惑选定文档和迷惑Google搜索的所有办法来搜索它们。

在大多数教育机构，通常都采用为学生们指定ID号来保护他们的年级号和个人信息的私密性。然而，如图11-66所示，这些识别号通常都会使用学生们的社保号。

SSN自身并非是一个重要的信息，当与学生的年级一同张贴出来时（如图11-67所示），这个系统可以很好地保护学生成绩的隐私。

图11-66 将社保号码作为学生的ID号

图11-67 “匿名的”学生号以及年级记录

然而，在大多数情况下，学生的名字都会与他们的社保号同时张贴出来，如图11-68所示。这样一来，通过使用ID号而非名字创建的匿名性就被破坏了。

在某些情况下，这些文档并不是为了查看而制作的，但是不知何故出现在了面向互联网的网站上。当然，这是一个不安全的处理实例，文件会出现在Google的缓存中。如图11-69所示的文档将会被一个匿名的Google黑客在一个开放的目录中找到。注意，它列出了学生的名称、SSN和其他更多的信息。更糟糕的是，这个文档可以在美国政府训练机构网站找到。后来删除了这个文档。

图11-68 名字与社保号又一次一同出现

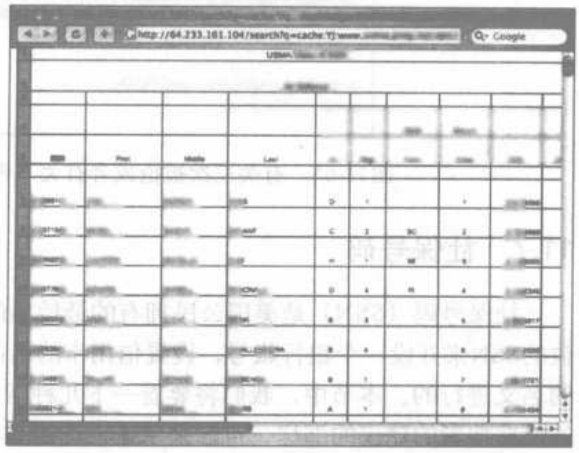


图11-69 ID盗窃者的生日礼物：SSN和名字

社保号码还会在网站上以其他的不太会引人注意的方式出现。如图11-70所示的履历申请，在信息社区布告板上列出了个人的SSN。

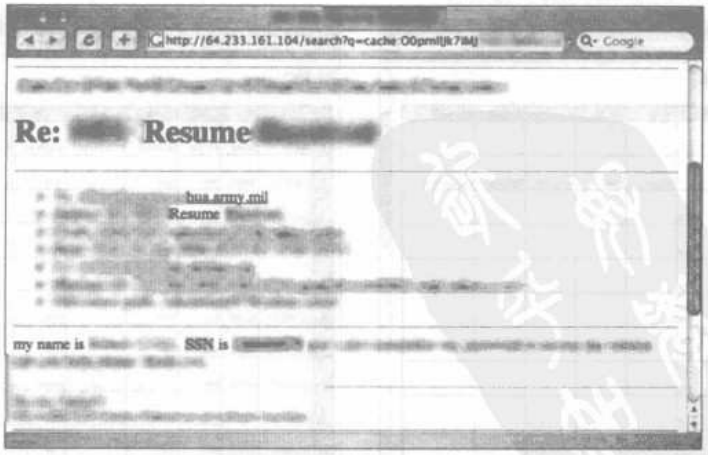


图11-70 注意这个家伙，这里有他的SSN

如图11-71所示的文档是个人履历（CV）。我不是很确定是什么类型的CV，但是经过一段

时间的调查，我发现它是某个真正的精英的个人履历。

图11-71 我很聪明，想看我的简历吗

至于我，我想我会保留自己的简单的旧履历的，尤其是如果要维护简历意味着我必须公开我的生日以及社保号。最后，看来一下如图11-72所示的电子数据表格，该表列出了公司雇员的姓名、出生日期、性别、受雇日期以及SSN。

图11-72 泄露的雇员信息

信用卡信息

信用卡号显然是非常有价值的，而且应该被妥善保管。不过，如前所述这些号码可以在网

站上毫不费劲地找到。图11-73是一个相对而言比较小的文档，它同时列出了Visa卡号与相关的到期日期。

图11-73 Google攻击信用卡信息

如图11-74所示的是一个更大的文档，它不仅列出了信用卡号和相关的到期日期，还列出了通常用于验证卡为合法持卡者所有的卡认证号（CVV）。

图11-74 Google攻击更多的信用卡信息

如图11-75所示的是一个超大的文档，它包含了成百上千个受害人的个人信息，如姓名、地址、电话号码、信用卡信息、CVV代码以及到期日期。

然而，信用卡号和到期日期并非Web上唯一敏感的金融信息，如图11-76所示。

图11-75 Google攻击众多的信用卡信息

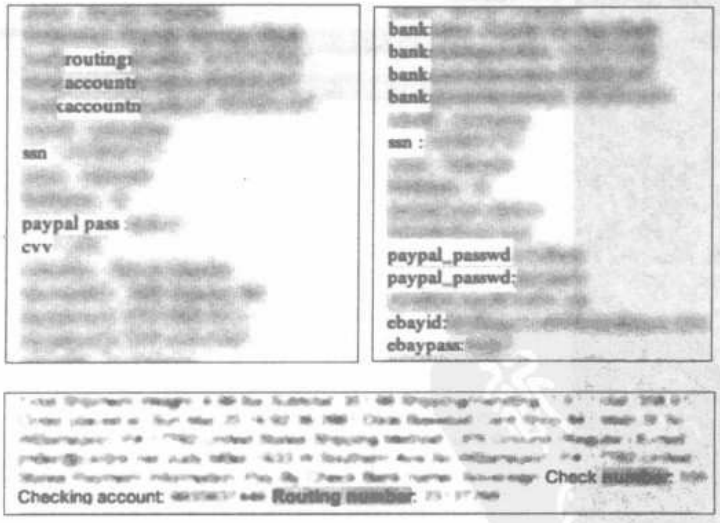


图11-76 没有任何东西是神圣的

注意

通常，像这种信息都是由钓鱼者（使用电子通讯请求个人信息的罪犯）收集的，并且放在一个在线列表或者数据库中。在很多情况下，调查者都会搜索这些列表或者数据库，并且在在线讨论组中张贴它们的链接。当Google的爬行者追踪链接时，被抓取的数据就会

暴露给Google黑客。在其他的情况下，卡持有者（信用卡号交易者）会在公开的Web讨论的网站上张贴出该数据，之后Google会抓取到它并对它进行缓存。要想了解关于钓鱼的更多的信息，可以查看来自Syngress Publishing的Phishing Exposed。

这些示例是从各种网站收集到的，其中包含银行的Routing number、PayPal用户名和密码、eBay用户名和密码、银行账号和Routing number以及更多的其他的信息，多半是由钓鱼者收集。

11.8 Google之外的信息

在某些情况下，Google都是长长的攻击链中的第一步。正派的黑客通常都会在下一步中采用非Google操作。本节中，我们将快速查看一些有趣的Google攻击，在攻击的过程中都采取了额外的一些步骤。尽管执行起来很简单，但是这些实例却揭示了黑客所能施展的创造性才华。

如图11-77所示的第一个屏幕快照（由CP提供）向我们展示了因为隐私目的而从Web上删除的职员目录。

图11-77 职员通讯列表被删除了

这不是一个坏主意，但是问题是旧的文档文档也必须从站点删除，或者类似archive.org站

点将会长时间地提供住文档的链接。图11-78显示了从源Web站点收集到的职员通讯文档，这归功于来自archive.org的链接。

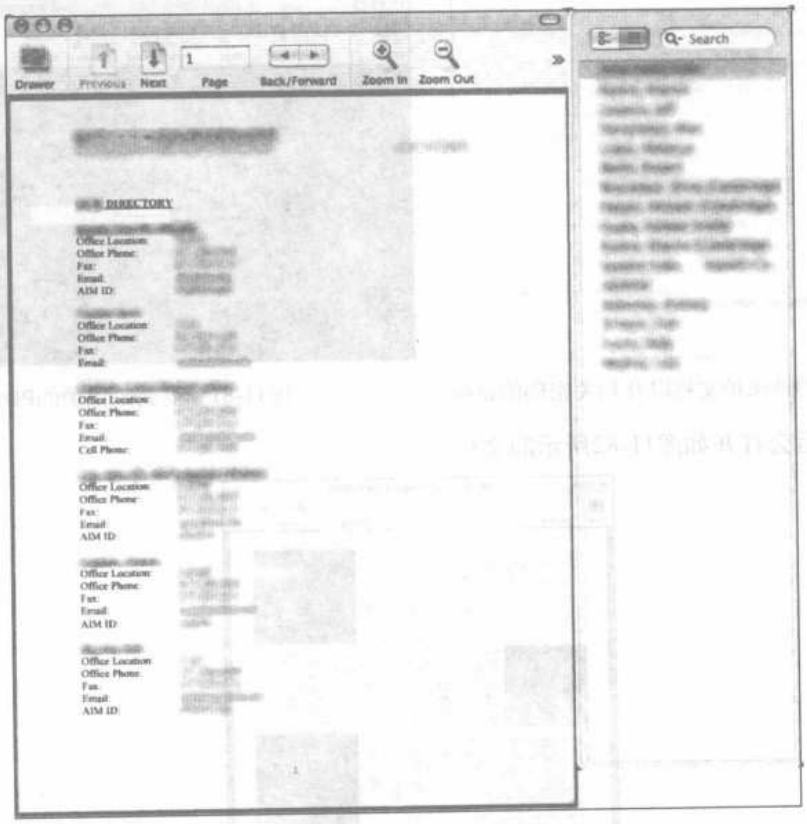


图11-78 恢复的职员通讯列表

在下一个示例中，Google黑客注意到了位于PDF文档中的密码引用，如图11-79所示。

图11-79 PDF文件密码引用

下载后，发现PDF文件确实包含一个密码引用。本例中，会出现一个如图11-80所示的连接到带密码保护的PDF文档的链接形式。

正如在图11-81中看到的那样，引用的PDF文件真的受到了密码保护。

图11-80 连接到带保护文档以及相关密码的链接

图11-81 受密码保护的PDF文档

输入密码后会打开如图11-82所示的文档。

图11-82 使用被盗密码打开的敏感文档

对文档添加口令然后公布口令没有任何意义，但是在本例中却出现了这个问题，因为包含密码引用的源文档并不打算对外公开。在本例中，故障会导致敏感政府文档的泄露。

11.9 总结

本章的所有内容都是关于Google攻击威胁被忽略后可能导致的严重错误。不论你何时身处严重的威胁都可以参看本章内容。请帮助传播这一信息，并且成为解决方案的一部分而非问题的一部分。在向Google发送暂停或终止文件之前，记住——如果你的敏感数据使它成为在线内容，那么就不是Google的错了。

第12章 防卫Google黑客

12.1 简介

这本书的目的在于帮助你理解Google黑客在攻击时可能采用的手段，这样你就能合理地保护你自己和你的客户免于遭受这种似乎没有危险的威胁的侵害。从我们的角度来看，实现这一目的的最好方法就是告诉你一个欲攻击Google搜索引擎的黑客能够做些什么。有一点我们必须明确讨论的就是怎样防止这类信息泄露或者是怎样补救已经存在的泄露。这一章就是讨论怎样保护站点（或者客户站点）免受攻击。

我们会从几个角度来讨论这一话题。首先，理解在互联网上发布数据的强效策略的价值是十分重要的。这并不是一个技术话题，而且也有些无趣，但是在增强任何一个网站的安全性时绝对都需要一种安全策略。其次，我们还会看一些技术话题，它们描述了怎样针对Google的（或者其他搜索引擎的）“爬行者”（crawler）来增强网站的安全。然后我们再来看看一些能够用来帮助检查网站信息泄露情况的工具，并花些时间来讨论用Google来支持防卫的几种方法。

Google搜索背景知识

详细介绍在哪儿

有许多类型的服务器和配置都展示了他们是如何保护自己的。对于Web服务器安全的讨论可以很容易地延伸成一整本书。这里我们将从更高的层面来看服务器安全，我们关注的是为了保护自己免受Goolge黑客的威胁你可以采取的策略。要了解更多的细节，请查阅“网站链接”一节中所提供的参考。

12.2 完善且坚固的安全策略

如果你没有一种有效的安全策略，即使你能用钱买到最好的硬件和软件配置，也不能保护你的资源。在实施任何软件保证（software assurance）之前，最好花些时间来审查你的安全策略。一种好的、合理地强制执行的安全策略大致包括你要保护的资源，怎样安装保护机制，操作风险的可接受级别以及在遇到攻击或者灾难时该怎样处理。没有一种坚固的、强制执行的安全策略，就相当于在打一场注定要失败的仗。

12.3 Web服务器安全防护

有几种方法可以阻止喜欢探测的Web crawler深入挖掘你的站点。但是，要记住的是Web服务器是最适合于存储用于公众访问的数据的。尽管对所有的数据都做了最好的保护，信息泄露还是会发生。如果你是真得在意保护你的私人敏感信息，那么就不要把这些信息放在公共Web

服务器上。把那些数据转移到内部网中，或者转到某个特别的服务器上，这个服务器能够以安全、负责和强制执行的策略的方式来保护这些数据。

不要根据访问级别而把公共Web服务器分成各种不同的角色。用户可以很容易地把数据从一个文件复制到另一个文件，这会导致某些基于目录的保护机制失效。同样，来看一个公共的Web服务器系统被攻陷之后会带来的后果。在一个考虑细致、结构合理的环境中，公共Web服务器的攻陷只是导致公开信息的泄露。适当的访问限制能够阻止攻击者利用Web服务器入侵其他机器，这使得攻击者想进一步渗透以获取敏感信息变得更为困难。如果敏感信息和公共服务器上的公开信息保存在一起，那么如果该服务器被攻陷，也就意味着很有可能会泄露大量的敏感信息。

我们先来看一起相当简单的措施，这些措施可以从Web服务器内部增强其安全性。这些都是通用的规则，它们并不能提供一种完全的解决方案，只是强调防卫中常见的关键区域。我们不讨论任一特定类型的服务器，而是给出任何Web服务器都通用的建议。我们也不会钻研保护Web应用程序过程中的某些特定的方面，而是探讨更为通用的、已经证明能够有效地阻止Web crawler程序的方法。

12.3.1 目录列表和缺失的索引文件

我们已经了解了目录列表会带来的风险。虽然信息泄露得比较少，但是目录列表能够允许Web用户查看目录中大多数的（如果不是全部的话）文件，以及任何下级的子目录。与只能浏览受限制的事先准备好的页面相比较而言，目录列表能够允许更为自由的访问。取决于许多因素，例如文件和目录的权限以及服务器允许访问的文件设置，Web浏览器都可能能够随意访问不应该公开的文件。

图12-1给出一个目录列表的例子，这个目录列表泄露了一个htaccess文件的位置。通常，这个文件（应该叫做.htaccess，而不是htaccess）是用来保护目录内容不被非授权的用户查看的。但是，由于服务器的不当配置使得这一文件显示在了目录列表中，甚至还是可读的。

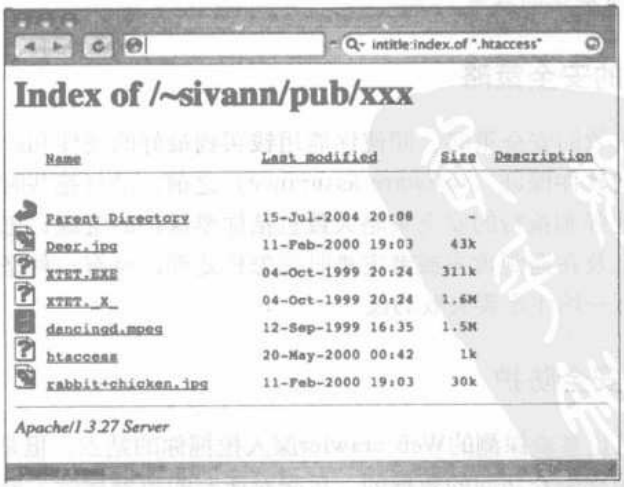


图12-1 目录列表提供了非公开文件的“路线图”

应该禁止目录列表，除非你想让访问者以FTP风格的形式仔细查阅这些文件。在一些服务器上，如果索引文件（根据你的服务器配置来定义）缺失，就有可能会出现目录列表。在每一个应该给用户显示一个页面的目录中都要有这些文件，例如index.html、index.htm或者default.asp。对于Apache Web服务器，你可以在httpd.conf中的单词Indexes之前加上一个破折号或者减号来禁止目录列表。如果禁止了目录列表（或者是Apache称作的“Indexes”），那么这一行应该类似于下面：

```
Options -Indexes FollowSymLinks MultiViews
```

12.3.2 利用Robots.txt阻止Crawler

Crawler也称为robot或者bot。robots.txt文件为自动的Web Crawler提供了一个命令列表。robots.txt文件标准位于http://www.robotstxt.org/robotstxt.html。这个文件允许你非常精确地定义哪些文件和目录是禁止Web robot访问的。这个文件必须放在Web服务器的根目录中，并且要赋予Web服务器读取该文件的权限。文件中的注释语句以#开头，注释将被忽略。其他任何不以#开头的语句，必须以User-agent或者disallow声明开头，后接一个冒号和一个可选的空格。这些语句用于禁止某些Crawler访问特定的目录或文件。每一种Web Crawler都应该向服务器发送一个user-agent域，这个域给出了Crawler的名称或者类型。例如Google的user-agent域的值为Googlebot。如果要禁止Google，那么user-agent语句应为：

```
user-agent: Googlebot
```

根据最初的robots.txt规范，通配符*可以用在user-agent域中，表示所有的Crawler。disallow语句精确描述了Crawler不能查看的地方。这个文件的最初规范非常不灵活，一个disallow语句只能描述一个完整或者部分URL。根据规范，Crawler会忽略任何以指定字符串开头的URL。例如，Disallow: /foo会命令Crawler不但忽略/foo，而且会忽略/foo/index.html，而Disallow: /foo/命令Crawler忽略/foo/index.html而不是/foo，这是因为foo后面存在一个正斜杠。如下是一个合法的robots.txt文件：

```
#abandon hope all ye who enter
User-Agent: *
Disallow: /
```

这个文件表示不允许任何Crawler访问该网站，也就是说完全排除了所有的Crawler。robots.txt的读取规则是由上到下按照有序的规则进行读取。robots.txt文件中是没有allow语句的。如果要允许某个特定的Crawler，只要不禁止它访问任何地方就可以。这可能类似于回溯逻辑，但是下面的robots.txt则是禁止了除叫做Palookaville之外的所有Crawler：

```
#Bring on Palookaville
User-Agent: *
Disallow: /
User-Agent: Palookaville
Disallow:
```

注意在Palookaville的disallow语句中并没有正斜杠。（Norman Cook的歌迷会高兴地注意到

Palookaville周围的斜杠和点都没有了。）没有任何禁止就意味着是允许该user-agent的，这样说虽然有些让人迷惑，但事实就是如此。

Google支持robots.txt标准的扩展。例如，Disallow句型可以包含*来匹配任意个数的字符。另外，$符号表示名称的结束。例如，要阻止Googlebot抓取所有的PDF文档，可以使用如下的robots.txt文件：

```
#Away from my PDF files, Google:
User-Agent: Googlebot
Disallow: /*.PDF$
```

在你放置了一个robots.txt文件之后，可以通过浏览www.sxw.org.uk/computing/robots/check.html.的Robots.txt Validator来检查它的有效性。

Google搜索背景知识

Web Crawler与Robots.txt

黑客并不需要遵守你的robots.txt。实际上，Web Crawler也不一定非要遵守robots.txt，虽然大数多知名的Web Crawler都是遵守的，除非是出于某些特殊的目的。一种相当常见的黑客技巧是首先查看一个网站的robots.txt文件以了解该服务器上的文件和目录是如何映射的。事实上，如图12-2所示，一个快捷的Google查询就能够暴露出许多网站的robots.txt文件。当然，这是一个错误的配置，因为robots.txt应该呆在幕后。

图12-2 Robots.txt不应该被抓取到

12.3.3 NOARCHIVE：缓存“杀手”

robots.txt能够阻止Google访问站点的某些区域。但是，有时你想要Google抓取某个页面，

却不想让Google保存该页面的缓存版本或者在它的搜索结果中出现一个“缓存版本”的链接。这可以通过META标记来实现。要阻止所有的（同时的）Crawler保存某个文档的存档或缓存，需要在文档的HEAD节放置如下的META标记：

```
<META NAME="ROBTOS" CONTENT="NOARCHIVE">
```

如果你只想阻止Google保存该文档的缓存，可以在文档的HEAD节中使用下面的META标记：

```
<META NAME="GOOGLEBOT" CONTENT="NOINDEX, NOFOLLOW">
```

可以用这种把Crawler的名称指定为META NAME的方法来阻止任一合作的Crawler。需要知道的是这个规则只针对Crawler。Web访问者（和黑客）仍然能够访问这些页面。

12.3.4 NOSNIPPET：去除摘要

摘要（snippet）是列在Google结果页面的文档标题之下的一段文本。当你在许多结果中查找需要的文档时，摘要都能够提供返回的文档的一些内部信息，这是很方便的。但是，有些时候需要去除摘要。先来看一下订阅型的新闻服务这种情况。虽然这类网站想要Google给其做些宣传，但是它需要保护它的内容（包括内容摘要）以防止那些没有付费的订阅者查看。这样的网站就可以通过把NOSNIPPET META标记和基于IP的过滤机制结合在一起的方法实现这一目标，让Google的Crawler安全地浏览网站的内容。要阻止Google显示摘要，只需在文档中加入下面的代码：

```
<META NAME="GOOGLEBOT" CONTENT="NOSNIPPET">
```

NOSNIPPET标记还有一种有意思的作用，即使用了NOSNIPPET标记之后，Google同样不会对该文档进行缓存。NOSNIPPET同时去除了摘要和缓存页面。

12.3.5 口令保护机制

Google不能填充用户认证表单。当页面中出现一个典型的口令表单时，Google似乎只是简单地从该页面离开，然后只在它的数据库中保存该页面的URL。虽然曾经谣传Google能够绕过或者有些神奇地绕过安全检查，但是那些谣言从没有被证实。这些事情更多的只是时间上的问题。

如果Google在页面被保护之前或者口令保护被取消时抓取了该具有口令保护的页面，那么Google会保存这个受保护的页面的一份缓存图片。如果点击原始的页面则会出现输入口令对话框，但是其缓存页面却没有这么做，造成了一种Google绕过了该页面的安全保护的错觉。另外，Google新闻搜索会提供订阅站点的新闻的摘要（如图12-3所示），但是当点击该新闻的链接时会出现一个要求注册的界面，如图12-4所示。这也造成了Google能够神奇地绕过令人讨厌的口令对话框和注册界面的错觉。

如果你真的很想阻止普通公众（以及像Google这样的Crawler）访问你的数据，可以考虑一种口令保护机制。Apache就有一种基本的口令认证机制，即htaccess。一个htaccess文件和一个htpasswd文件允许你定义一组用户名/口令列表，以允许相应的用户访问特定的目录。你可以在http://httpd.apache.org/docs/howto/htaccess.html中浏览Apacher htaccess教程，或者用Google来

搜索htaccess howto文档。

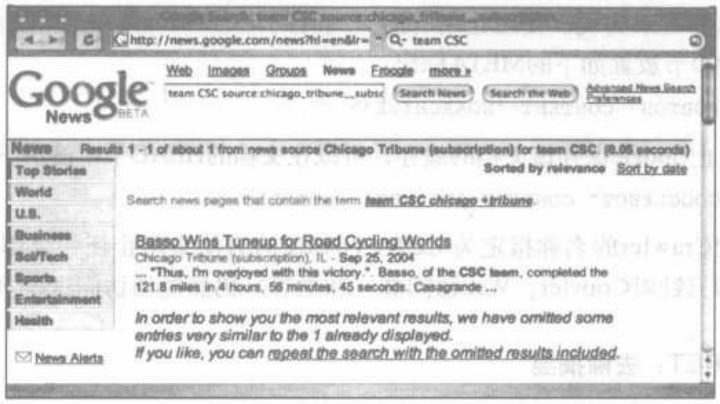


图12-3 来自受保护站点的Google抓取信息

图12-4 一个保护密码的新站点

12.3.6 软件默认设置和程序

在整本书中我们都可以看到，即使是最初级的Google黑客也只是需要少许的努力就能够搜索到默认的页面、词组、页面标题、程序和文档。牢记这一点，并且还要从任何你所安装的软件中删除这些项。同样，确保删除了默认的账号和口令以及删除了软件提供的安装脚本或者程序也是一种好的安全实践。由于Web服务器安全是相当广泛的一个课题，所以我们只看看一些常见的服务器应该注意的事项。

首先，对于Microsoft IIS 6.0而言，先来看一下本章后面的“网站链接”一节中列出的IIS 6.0安全的最好实践。

对于IIS 5，Microsoft IIS 5.0 Security Checklist（参见本章后面的“网站链接”一节）列出了许多能够帮助增强IIS 5.0服务器安全性的建议：

- 删除\IISSamples目录（通常位于c:\inetpub\isssamples）。
- 删除\IISHelp目录（通常位于c:\winnt\help\iishelp）。

- 删除\MSADC目录（通常位于c:\program files\common files\system\msadc）。
- 删除\IISADMPWD虚拟目录（可以在c:\winnt\system32\inetsrv\iisadmpwd目录和ISM.dll文件中找到）。
- 删除不常用的脚本扩展：
  - 基于Web的口令修改：.htr。
  - Internet数据库连接器：.idc。
  - 服务器端的包含文件：.stm、.shtm和.shtml。
  - Internet打印：.printer。
  - 索引服务器：.htw、.ida和.idq。

虽然Apache 1.3系列提供了更少的默认页面和目录，但是还要注意如下的几点：

- Web根目录的/manual目录包含默认的文档。
- Web根目录中的一些语言文件是以index.html开始的。这些默认的语言文件在不用时可以删除。

要了解有关保护Apache安全的更多详情，可访问http://httpd.apache.org/docs/2.0/misc/security_tips.html参阅Security Tips文档。

Google搜索背景知识

给系统打补丁

如果说只做一件事就能让任何系统就更为安全，那么就保持更新安装最新的软件安全补丁。虽然这种说法在当今的安全圈子内已是陈词滥调，但是再多强调这一点也不算过分。虽然配置不当也是造成系统安全问题的一个方面，但是如果没有一个牢固的根基，你的服务器根本不可能安全。

12.4 攻击你自己的站点

攻击自己的站点是一种相当好的了解站点的可能安全风险的方法。显然，没有任何一个人能够懂得Hacking所需要的所有知识，这意思是说攻击自己的站点是无法取代一名专业人员实施一次真正的渗透测试的。即便你的职业是一名渗透测试人员，从另一种角度来观察你的安全形势也是无伤大雅的。在Google Hacking领域中，你可以使用一些自动化工具和技术来了解Google是如何对待你的站点的。我们从手动的方法开始，以讨论一些自动化的技术来结束这一小节。

警告

我们将在本章中看到，有几种方法可以实现Google搜索的自动化。要执行自动搜索，必须使用Google提供的带有Google授权密钥（Google License Key）的应用程序接口（API，Application Programming Interface），除此之外的方法Google都不允许。任何违背Google的服务条款，不要求你提供授权密钥的程序，都是要受到Google的惩罚。更多详情可访问www.google.com/accounts/TOS。善待Google，Google也会善待你！

12.4.1 用Site操作符搜索自己的站点

我们已在本书通篇讨论了site操作符，但还是要知道site允许你把搜索精确到某个特定的域或者服务器。如果你是（给你印象最深的）工具NIKTO的设计者和cirt.net的管理员sullo，查询site:cirt.net便会列出所有来自cirt.net服务器的Google缓存页面，如图12-5所示。

图12-5 Site搜索是一种测试Google泄露你的信息的方法

你当然可以逐一点击这些链接或者简单地浏览结果列表，以判断这些页面是否是能够公开访问的，但是这个任务太浪费时间，尤其是当结果数超过几百条时。显然，你需要这一过程能够自动处理。让我们来看一些自动化的工具。

12.4.2 Gooscan

由Johnny Long编写的Gooscan是一种基于Linux的能够进行大量的Google搜索的工具。这个工具没有使用Google API，所以违背了Google的服务条款。是否在明白会违背Google的服务条款的情况下还使用这个工具来检查源自你的站点的信息泄露情况，需要做出判断。如果你决定使用一种非基于API的工具，那么要明白Google会禁止某些IP段使用它的搜索引擎（但是很少这样做）。同样还要知道这个工具只是用来增强你自己的站点的安全性的，不要用来攻击其他人的站点。除非你经常游走于法律的边缘，否则只能把Gooscan的代码作为一种学习工具使用，而且不要运行它！

可以从http://johnny.ihackstuff.com获得Gooscan。不要对华丽的界面和简单易用的特性期望太高。这个基于UNIX的工具仅提供命令行界面，而且需要具备一些技术知识才能安装和运行。好处是Gooscan能够帮你做些费力的活，而且目前它也是用来代替只适用于Windows平台的工具的最好选择。

1. 安装Gooscan

在安装Gooscan之前，首先要下载相应的tar文件，然后用tar命令对其解压缩。Gooscan带有

一个C程序，一个README文件和一个存储了一些数据文件的目录，如图12-6所示。

从tar文件中提取出这些文件之后，你必须用编译器编译Gooscan，例如GCC。Mac用户要先安装XCode包，可以从http://connect.apple.com/的Apple Developers Connection 站点获得。Windows用户可以考虑一种更为“图形化”的工具，例如Athena或者SiteDigger，因为Gooscan目前还不能在诸如CYGWIN这样的环境下编译。

2. Gooscan的选项

在运行Gooscan时，如果没有选项（或者是使用了不正确的选项），Gooscan就会列出其用法，如图12-7所示。

图12-6 提取和安装Gooscan

图12-7 Gooscan的用法

Gooscan的最常用的选项都列在了内含的README文件中。我们来看看这些不同的选项是怎样使用的：

- **<-t target>（需要参数）**这是要扫描的Google Appliance或者服务器。参数可为IP地址或主机名。警告：在这里输入www.google.com是违反Google服务条款的，作者不推荐也不允许这么做。
- **<-q query | -i query_file>（需要参数）**要发送的查询或查询文件。Gooscan可以用来发送一个单独的查询语句或者从文件中读取的一组查询语句。-q选项接受一个参数，该参数可以为任何合法的Google查询。如下是一些合法的选项：

```
-q googledorks
-q "microsoft sucks"
-q "intitle:index.of secret"
```

- **[-i input_file]（可选参数）**-i选项的参数是Gooscan数据文件的文件名。使用数据文件能够允许你用Gooscan执行多次查询。参见如下关于内含的Gooscan数据文件信息列表。
- **[-o output_file]（可选参数）**Gooscan能够创建一个美观的HTML输出文件。这个文件包含了每条查询的实际Google搜索结果页面的链接。
- **[-p proxy:port]（可选参数）**这是一个HTML代理服务器的地址和端口。查询会发送到该代理服务器，并且返回到由-t参数说明设备上。其格式类似于10.1.1.150:80或者是proxy.validcompany.com:8080。

- **[-v]（可选参数）**Verbose模式。每种程序都需要verbose模式，尤其是当作者厌烦了命令行模式的调试器时。
- **[-s site]（可选参数）**这一过滤选项是指只从某个特定的站点获取结果，即在每个Gooscan提交的查询中添加site操作符。这一参数对Google Appliance没有任何意义，因为Google Appliance已经做了站点过滤。例如，以下的Google查询：

```
site:microsoft.com linux
site:apple.com microsoft
site:linux.org microsoft
```

- 如果Google允许的话，你可以向下面一样运行Gooscan来得到同样的结果：

```
$ ./gooscan -t www.google.com -s microsoft.com linux
$ ./gooscan -t www.google.com -s apple.com microsoft
$ ./gooscan -t www.google.com -s linux.org microsoft
```

[-x]和[-d]选项只适用于Google Appliance。本书不对Google Appliance做过多的涉及。只要知道大多数适用于Google.com的技术同样也适用于Google Appliance就足够了。

3. Gooscan的数据文件

在多种查询模式中，Gooscan都会从数据文件中读取查询语句。数据文件的格式如下：

```
search_type | search_string | count | description
```

其中search_type的取值如下：

- intitle 在页面标题中查找serach_string。如果是以命令行形式执行该命令时，Gooscan会添加site查询。例如：

```
intitle|error||
```

它会在页面的标题中搜索单词error。

- inurl 在页面的URL中查找search_string。如果是以命令行形式执行该命令时，Gooscan会添加site查询。例如：

```
inurl|admin||
```

它会在页面的URL中搜索单词admin。

- indexof 在目录列表中查找search_string。如果是以命令行形式执行该命令时，Gooscan会添加site查询。例如：目录列表的页面标题通常含有index of字样。Gooscan会生成一个类似于下面的Google查询：

```
intitle:index.of search_string
```

注意

在使用site选项时，Gooscan会自动执行一个针对目录列表的通用查询。该查询类似于intitle:index.of site:site_name。如果这个查询没有返回结果，Gooscan会跳过后面所有的indexof搜索。从逻辑上推断，如果最通用的indexof搜索都没有返回任何结果，那么就可以跳过特殊的indexof搜索。

- filetype 把search_string作为文件名进行搜索，如果以命令行形式执行该命令时，Gooscan会添加site查询。例如：

```
filetype|cgi cgi||
```

这个搜索会查找以.cgi为扩展名的文件。

- raw 这种search_type允许用户创建自定义查询。Gooscan不会对其做任何修改，然后将其提交给Google，如果是以命令行形式执行该命令，则会添加site查询。例如：

```
raw|filetype:xls email username password||
```

这个例子会查找文档中包含单词email、username和password的Excel表格。

- search_string 该项相当直观。它允许使用任何字符串，但不包括字符\n和|。在将其发送给Google之前，系统会将其转化为HTML编码。例如字符A将变成%65，以此类推。也有一些例外，例如空格将转换为字符+。
- count 这个域记录了近似的结果数。不适用于Site操作符。这个值有些特殊，因为它是基于Google提供的近似数，而且取决于何时以及如何执行搜索，所示该值会经常变化。这个值还能为数据文件的排序以及创建自定义的数据文件提供有价值的水印。例如，在运行一个比较大的搜索之前，可以安全地忽略值为0的记录。（目前Gooscan不使用该域。）
- description 这个域描述了搜索类型。目前只有filetype.gs数据文件使用该域。请继续阅读以了解更多关于filetype.gs数据文件的信息。

Gooscan包含了许多数据文件，其用途各不相同：

- gdork.gs 这个文件包含了从http://johnny.ihackstuff.com的Google Hacking数据库（GHDB）中摘录的一些查询。GHDB是互联网上最大的Google Hacking查询数据库，由上千个Search Engine Hacking Forum论坛的成员维护。GHDB每周都会多次更新，目前大约有750条各不相同的查询。
- filetype.gs 这个庞大的文件包含了现有已知的文件类型，主要根据www.filext.com。通过选择该文件的某些行，你能够快速地判断服务器上存在的文件类型。我们建议创建该文件的一个子集（可以在这个域使用一条Linux命令：head -50 filetype:gs > short_filetype.gs）。不要原样不动地使用该文件。它太大了。这个搜索会执行8000多条查询，当然这会花许多时间，而且会消耗目标服务器上宝贵的资源。相反，count域可以告诉你在Google数据库里，大概有多少个站点包含这些文件，你可以根据这一点，只选择那些最常见的或者和你的站点最相关的文件来搜索。这个filetype.gs文件列出了最常见的文件扩展名。
- inurl.gs 这个非常大的数据文件包含了取自最流行的CGI扫描器的字符串。这些CGI扫描器擅长于在Web服务器中搜索程序。这个文件是以Google结果的近似结果数排序的，在最上面列出了常见字符串，在最下面列出了非常少见的CGI漏洞字符串。这个数据文件会在页面的URL中查找相应的字符串。这个文件也不应该原样不动地使用。
- indexof.gs 这个数据文件非常类似于inurl.gs文件，它是在目录列表中查找字符串。同样，只要使用该文件的一部分就可以了，而不是全部！

4. 使用Gooscan

可以有两种不同的方式来使用Gooscan：单查询模式或者多查询模式。单查询模式不比Google的Web搜索功能好多少，除了能够给用户提供更方便移植的格式的Google查询结果。如图12-8所示，对关键字daemon9的所有Google搜索返回2440条结果。要让这个查询专门针对某个站点，例如phrack.org，可以添加[-s]选项。例如：

```
gooscan -q "daemon9" -t www.google.com -s phrack.org
```

注意到，当你选择了www.google.com作为目标服务器时，Gooscan会给出一个很长的免责声明。这个免责声明只会在你提交了一个可能违反Google服务条款的查询时才会出现。标准的Gooscan输出没有太大的价值，它只列出了Google搜索的结果数。你可以利用[-o]选项来创建一个更为美观的HTML输出格式。要想运行带有更为美观的输出的daemon9查询，可以运行：

```
gooscan -q "daemon9" -t www.google.com -o daemon9.html
```

如图12-9所示，HTML输出给出了Gooscan运行时使用的选项、执行扫描的日期、查询列表、实际Google搜索的链接以及结果数。

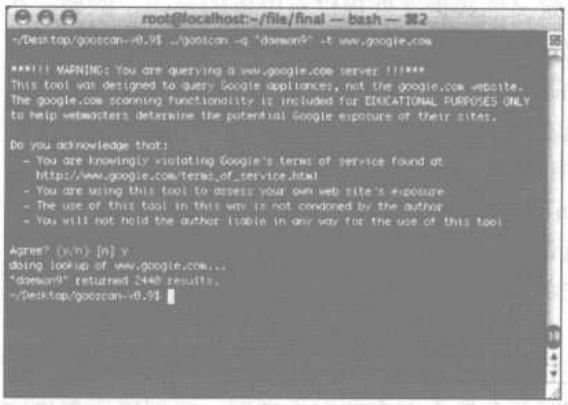


图12-8 Gooscan的单查询模式

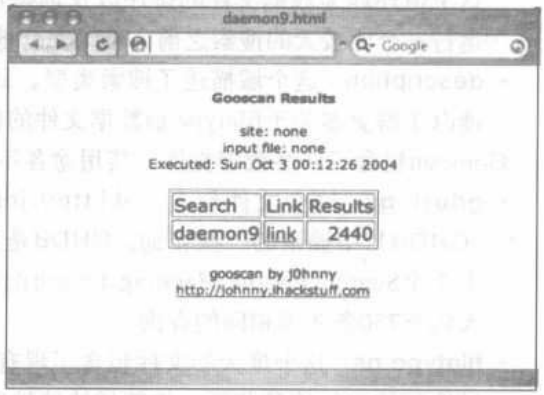


图12-9 单查询模式下的Gooscan HTML输出

HTML输出中的链接指向Google。点击这个链接就能执行相应的Google搜索。不要惊讶，Google页面上的结果数可能会和Gooscan输出中显示的不一样，这是因为Google的搜索结果通常只是一种近似情况。

以多查询模式运行Gooscan显然违反了Google的服务条款，但是如果使用恰当的话也不会造成太多的麻烦。一种让Google对你保持友好的方法是尊重它的服务条款，即只发送少许查询给服务器，而不是一次就发送出庞大的数据文件。如图12-10所示，你可以使用head命令来创建一个小的数据文件。命令`head -5 data_files/gdork.gs > data_files/little_gdork.gs`能创建一个具有4个查询的数据文件，因为gdork.gs文件有一行是注释。

Gooscan的多查询模式输出仍然没有什么价值，因此我们来看一看如图12-11中所示的HTML输出。

使用Gooscan的[-s]选项可以让我们针对某个特定的网站来获取结果，在本例中是http://johnny.ihackstuff.com，命令为：

```
Gooscan -t www.google.com -i data_files/little_gdork.gs -o ihackstuff.html -s
johnny.ihackstuff.com
```

图12-10 使用小的数据文件可以防止Google“不悦”

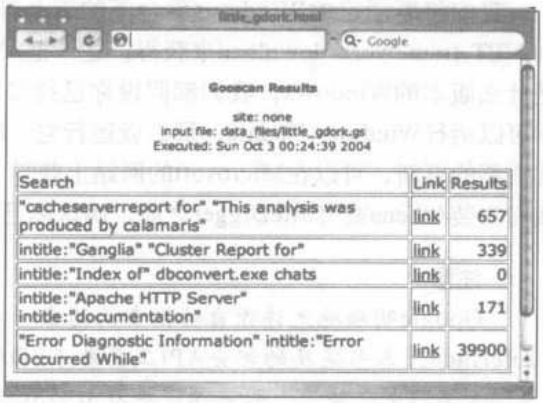


图12-11 多查询模式下Gooscan的HTML输出

其结果如图12-12所示。（不要担心，Johnny那个家伙不会介意的！）

大部分针对特定网站的Gooscan输出都是很整洁的，就和这个例子一样。如果你碰到了看起来让人感到怀疑的结果，那么可以点击相应的链接来看看Google得到的结果到底是怎样的。图12-13显示了完整的Google搜索。

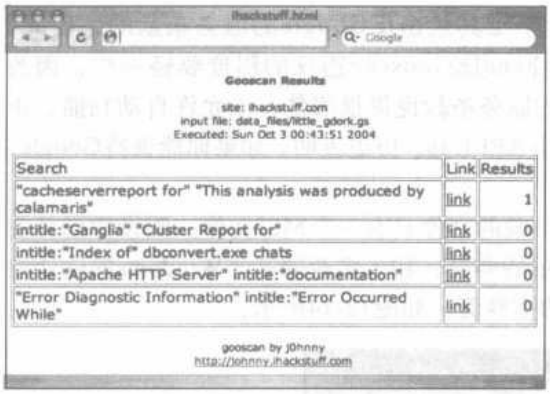


图12-12 针对特定网站的Gooscan输出

图12-13 来自Gooscan的连接到Google搜索结果的链接

在这个例子中，我们成功地找到了Google Hacking Database自身，它包含了一条和我们的Google查询相匹配的引用。其他的搜索都没有返回任何结果，这是因为它们比Calamaris查询更为特殊，而Calamaris查询不能查询标题、URL、文件类型以及其他类似的资料。

总的来说，Gooscan是一个不错的用来检查你的网站的信息暴露程度的工具，但是应该慎重使用，因为它没有使用Google API。最好把你的扫描转换成一些小规模的查询，除非你（不明智的）喜欢忍受权力机构找你的麻烦。

12.4.3 Windows平台下的工具和.NET框架

我们将要学习的Windows平台下的工具都需要Microsoft .NET框架，它可以用Google查询.NET framework download来获得。这一框架的成功安装取决于许多因素，但是不管你运行的是什么版本的Windows，我们都假设你已经安装了最新的服务包和更新。如果你的Windows版本可以运行Windows Update，那么就运行它。Internet Explorer的更新是成功地安装.NET框架最常需要的更新，可以在Microsoft的网站上获得（Google查询：Internet Explorer upgrade）。在下载和安装Athena或者SiteDigger之前，确认你已经正确地安装了.NET框架（版本1.1或者2.0）。

注意

Google明确地允许你自动化查询的唯一方法就是通过Google Application Programming Interface。本书提及的部分API工具基于SOAP API，SOAP API已经被Google停用并使用AJAX API来代替了。如果你还拥有旧的SOAP API密钥，那么你非常走运。该密钥仍旧可以与基于API的工具一同使用。然而，如果你没有SOAP密钥的话，不妨考虑使用SensePost的Aura程序（www.sensepost.com/research/aura）来替代旧的SOAP API。

12.4.4 Athena

Steve Lord（steve@buyukada.co.uk）编写的Athena是一种基于Windows的Google扫描器，它不基于Google API。和Gooscan一样，使用这个工具是违反Google的服务条款的，因此，Google能够禁止你的IP段使用它的搜索引擎。Athena比Gooscan违反的程度要轻一些，因为Athena只允许你一次执行一条搜索，但是Google的服务条款说得很清楚：不允许自动扫描。正如我们在讨论Gooscan时提到的一样，慎重地使用非API工具。历史表明，如果你能善待Google，Google也会善待于你。

可以从http://snakeoillabs.com/下载Athena。下载的文件只有一个MSI文件。假设你已经安装了.NET框架，那么Athena安装程序就是一个简单的向导，和大多数的基于Windows的软件安装程序一样。在安装并运行之后，会出现Athena的主界面，如图12-14所示。

图12-14 Athena的主界面

如图12-14所示，这个界面集成了一个简单的Web浏览器。Refine Search文本输入框允许你输入或者重新定义一个已有查询。它的搜索（Search）按钮类似于Google的搜索按钮，点击之后会执行一次搜索。

要利用Athena来执行基本的搜索，需要加载一个包含所需的搜索字符串的XML文件。只要简单地从Athena中打开这个文件，所有的搜索都会出现在选择查询（Select Query）下拉框中。例如，加载Athena里包含的digicams XML文件会加载一个排列得很整洁的数据照片搜索。从列表中选择一个查询，然后点击Search按钮。例如，在选择1st photo with a PENTAX cam点击Search按钮之后，Athena就会显示该查询的Google搜索结果，如图12-15所示。

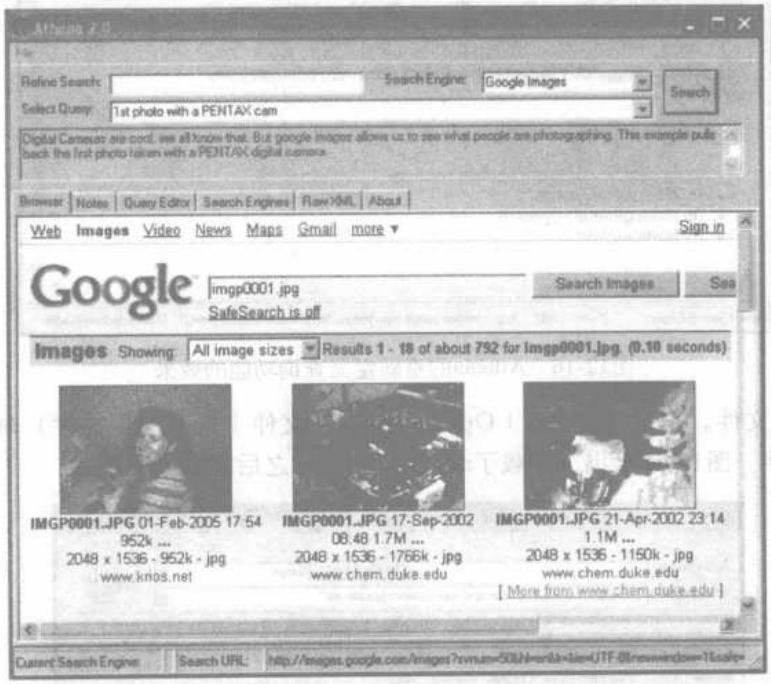


图12-15 基本搜索的结果

Athena也允许你使用Refine Search输入框来重新定义查询。仍然使用前面的查询，在Refine Search输入框中输入inurl:“buddylist.blt”并点击Search按钮，这样搜索结果就更为精确（如图12-16所示）。

结果表明图像并不存在于http://johnny.ihackstuff.com网站。此时，似乎Athena没有什么用处。它的功能只是类似于一个Web浏览器，把查询提交给Google，然后显示结果。但是，Athena最强大的功能在于它的基于XML的配置文件。

1. 使用Athena的配置文件

Athena包含了两个配置文件：Athena.xml和digicams.xml。这些文件包含了自定义的查询以及这些查询的描述。digicams文件中包含了用于查找图片的样例查询；Athena.xml文件中包含了在GHDB中找到的查询。

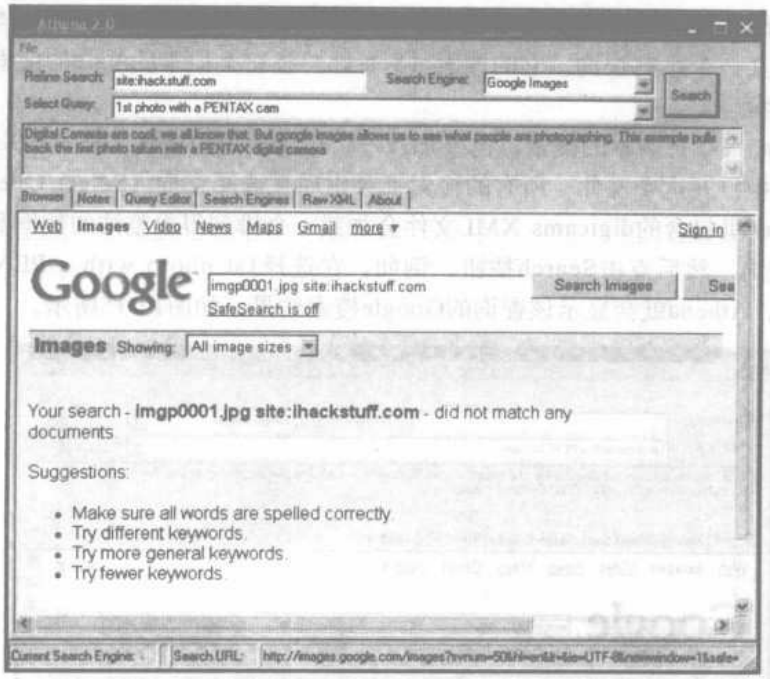


图12-16 Athena的重新定义查询功能的效果

要加载这些文件，可以执行File | Open Config（文件 | 打开配置文件）命令，然后选择Athena.XML文件。图12-17给出了加载了athena.xml文件之后的主界面。

图12-17 Athena加载了Athena.xml

正如前面提到的digicams图片搜索，在GHDB中进行的查询可以被修改并且通过Refine Search输入框重新提交。

2. 构造Athena配置文件

Athena的基于XML的配置文件可以根据你的需要进行修改或者完全自定义。其XML文件有两个主要的部分：searchEngine部分和signature部分。searchEngine部分描述了怎样构造一个特定搜索引擎的查询。一个典型的searchEngine部分如下代码所示：

```
<searchEngine>
        <searchEngineName>Google (UK)</searchEngineName>
        <searchEnginePrefixUrl>http://www.google.co.uk/search?q=
        </searchEnginePrefixUrl>
        <searchEnginePostfixUrl>%26ie=UTF-8%26hl=en%26meta=
        </searchEnginePostfixUrl>
</searchEngine>
```

这一部分负责描述各种搜索引擎如何处理搜索查询。searchEngineName域是一个简单的文本域，它描述了搜索引擎的名称。这个名称会显示在Athena的下拉框中，允许你从各种不同的搜索引擎中选择。searchEnginePrefixUrl域表示搜索URL的前一部分，将发送给搜索引擎。通常是假设在这个前缀之后填充查询部分。searchEnginePostfixUrl域描述了前缀和查询之后的部分。它通常描述各种选项，例如输出格式（UTF-8）。注意，Athena使用<searchEngine>部分，而SiteDigger不使用。可以把这个部分修改为搜索U.S.的Google引擎，修改后的searchEngine部分如下：

```
<searchEngine>
        <searchEngineName>Google (US)</searchEngineName>
        <searchEnginePrefixUrl>http://www.google.com/search?q=
        </searchEnginePrefixUrl>
        <searchEnginePostfixUrl>%26ie=UTF-8%26hl=en%26meta=
        </searchEnginePostfixUrl>
</searchEngine>
```

signature部分描述了将要执行的单独的搜索。一个典型的signature部分如下代码所示：

```
<signature>
        <signatureReferenceNumber>22
        </signatureReferenceNumber>
        <categoryref>T1</categoryref>
        <category>TECHNOLOGY PROFILE</category>
        <querytype>DON</querytype>
        <querystring>intitle:"Index of" secring.bak
        </querystring>
        <shortDescription>PGP Secret KeyRing Backup
        </shortDescription>
        <textualDescription>This query looked for a backup of the PGP secret key
        ring. With this keyring an attacker could decrypt messages encrypted by the
        user. </textualDescription>
```

```
        <cveNumber>1000</cveNumber>
        <cveLocation>http://johnny.ihackstuff.com</cveLocation>
</signature>
```

signatureReferenceNumber是赋给每个签名（signature）的唯一数字。categoryref是用来描述该签名所在的分类（category）环境中的唯一数字，而分类则由一个完整的category部分描述。querystring是将要执行的Google查询。它是HTML格式的，而且要放在发送给Google的URL中的searchEnginePrefixUrl和searchEnginePostfixUrl之间。shortDescription和textualDescription分别是简要的和详细的搜索描述。cveNumber和cveLocation引用了www.cve.mitre.org的常见漏洞和信息泄露列表（Common Vulnerabilities and Exposures list）。

该XML文件头应该包含下面的内容：

```
<?xml version="1.0" encoding="utf-8"?>
<searchEngineSignature>
```

而且该文件也应该以</searchEngineSignature>结尾。

使用这种格式可以相当容易地创建自定义查询文件。这个文件必须符合UTF-8字符集，严格遵守XML规范。这是说像<A HREF>和
这样的HTML标记不但必须和关闭标记相匹配，每个HTML标记的大小写都必须一致。Microsoft的XML扫描器在碰到开头是
标记而后面是
标记时会报错，因为这两个标记的大小写不同。小于号和大于号（< 和 >）如果使用不当也会造成一些问题。如果你的数据含有“grin”的Internet简写形式，即<G>，MS XML扫描器也会报错。

工具和陷阱

当前配置文件

GHDB的维护者编写了用于Athena的当前配置文件。这个文件可以从http://johnny.ihackstuff.com下载。

12.4.5 Wikto

Wikto是由Roloef Temmingh编写的一款令人称奇的Web扫描工具，那时候他在Sensepost（www.sensepost.com）处于领先位置。Wikto可以做很多不同的事，但是本书只着重讲解它在Google Hacking方面的技术。我们先来看一下该工具的Google扫描部分。默认情况下，Wikto的安装向导界面如图12-18所示。

Wikto首先会提示输入你要扫描的目标，以及有关目标服务器的详情。单击Next（下一步）按钮加载如图12-19所示的配置面板。

该面板提示输入代理信息，并且要求你提供Google API密钥。API问题很复杂，是因为Google不再发放SOAP API密钥。如果你已经有一把SOAP API密钥（你很幸运），那么便可以输入域并且进入到下一个页面。否则，不妨来看一下如何使用Sensepost的Aura（www.sensepost.com/research/aura）工具来模拟Google SOAP API调用。下载并安装来自SensePost站点的Aura，接着单击Start SensePost Aura将Wikto定位到Aura代理。在输入API密钥

（或者使用Aura绕过它）之后，再在向导的其他确认屏幕中单击确认通过。之后便会显示Wikto主界面。我们首先来关注一下Googler选项卡。单击Start（开始）按钮将会针对目标站点加载Google扫描，搜索列在File Types（文件类型）文本框中的特殊的文件类型。图12-20显示了对http://johnny.ihackstuff.com的扫描结果。

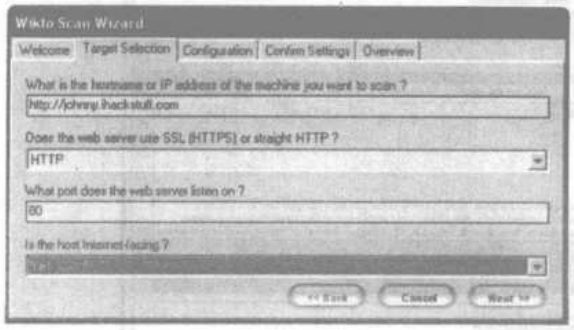


图12-18 Wikto的目标选择面板

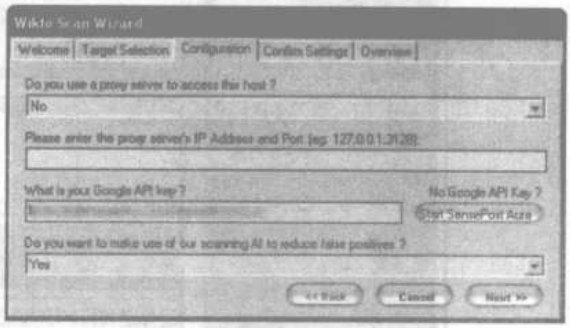


图12-19 Wikto的配置面板

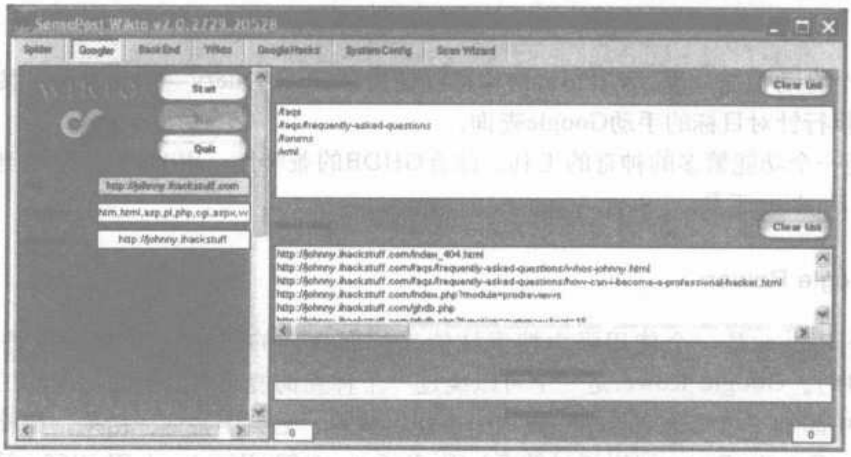


图12-20 Wikto的Google搜索功能

注意输出文本框列出的在目标站点中搜索到的文件以及目录。所有的这些信息都是通过Google查询收集的，这意味着这些事宜是对目标公开的、透明的。Wikto将在之后的扫描阶段使用该目录和文件信息。

接着，我们将来看一下如图12-21所示的GoogleHacks选项卡。

这个扫描阶段由来自http://johnny.ihackstuff.com的Google Hacking Database支持。单击Load Google Hacks Database会加载最新的GHDB版本，为Wikto提供上千个潜在的恶意Google查询。一旦加载完GHDB并按下Start（开始）按钮就会开始对目标站点的Google扫描。这里实质上发生的是Wikto发送了无数个Google查询，每个都带有一个指向目标Web站点的site操作符。GHDB的信息参见最上方的那个面板，所有的结果见最后一个面板。单击最后的那个面板的结果便可以在中间的面板里显示该查询的细节信息（来自GHDB）。在这个情况下，会返回很多

的结果，因为目标Web站点（http://johnny.ihackstuff.com）详细提及了这里的所有查询。

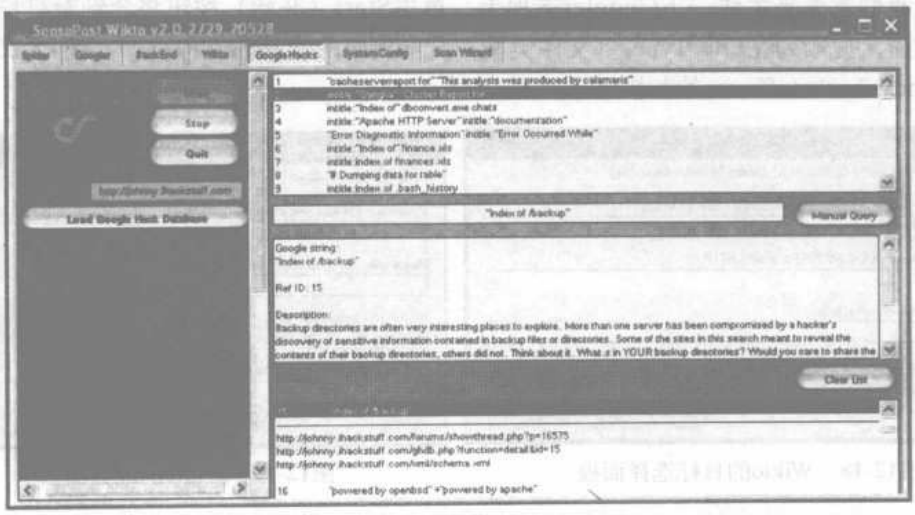


图12-21 Wikto的GoogleHacks功能

除了这个自动扫描过程，Wikto允许你通过使用Manual Query（手动查询）按钮和相关输入文本框来执行针对目标的手动Google查询。

Wikto是一个功能繁多的神奇的工具。结合GHDB的兼容性，Wikto就肯定是当前可用的最好的Google Hacking工具。

12.4.6 Google Rower

Google Rower是一个使用强力搜索技能来扩展查询的Firefox扩展（也是一个卓越的Windows程序）。Google Rower是一个可以绕过一千种查询搜索的限制的伟大的工具。它通过在基础查询中添加“填充数字”来完成这一工作。接着，它会丰富这一结果，删除副本，显示结果。例如，Google Rower可以通过搜索Jeffball55 a、Jeffball55 b、Jeffball55 c等来获取查询JeffBall5的更多的结果。

可以访问http://www.tankedgenius.com来下载Google Rower。安装过程则是一个简单直接的Firefox.xpi文件安装。在安装完Google Rower后，打开Firefox，选择Tools | Google Power（工具 | Google Rower）命令，并且输入如图12-22所示的查询。

图12-22 Google Rower选项界面

输入带有默认选项的ihackstuff查询将可以查询后面跟随了一系列字符的基础关键字ihackstuff。本例中，后面的系列字符为数字1～9。结果的分类及显示如图12-23所示。

另外，你还可以在Firefox中右击并且选择Google Rower。本例中，Google Rower将会发送

基于选中文本填写的查询搜索。

图12-23 Google Rower结果

Google Rower有几个可选择的选项，见表12-1。

表12-1 Google Rower选项

Google Rower选项	说 明
Duplicates/Separation	Google Rower通过查询和删除重复的链接为分离这些链接提供了几个不同的选项。不同的选项会影响该扩展的速度和内存
Print Titles	默认情况下，Google Rower会输出由Google返回的结果的链接。选择该选项将允许页面标题像是由Google报告的那样来输出
No Headings in Output	默认情况下，Google Rower会输出一些标题来显示哪些链接来自于哪些查询。选择该选项将会关闭这些标题的显示。当结果输出到其他程序时，该选项将非常有用

12.4.7 Google Site Indexer

Google Site Indexer（GSI）由Jeffball55（Jeff Stewart）和CP.GSI使用Google的一些Advanced Operator（高级操作符）编写而成，其中的site和inurl高级操作符用来创建目标Web站点的文件和目录映射。通过发送诸如site:tankedgenius.com之类的Google查询，GSI可以增量地索引Google已经编入索引之中的文件。然而，因为Google只检索最大为1,000的结果，所以GSI可以混合使用高级操作符（例如site:tankedgenius.com inurl:cp）来获取单一查询结果的一个更好的混合查询结果。GSI可以从www.tankedgenius.com下载。

安装是一个很简单的事：单击来自Firefox内部的.xpi文件即可触发这个安装过程。动行Google Site Indexer只需打开Firefox并且选择Tools | GSI命令。GSI界面如图12-24所示。

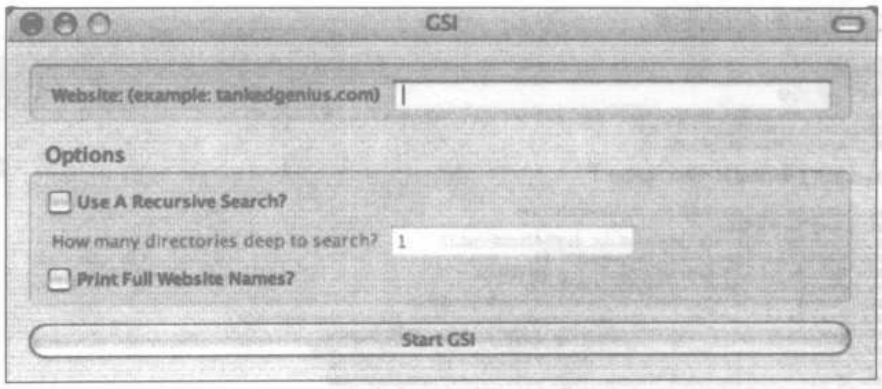


图12-24 GSI选项界面

执行也同样简单。简单地填写目标Web站点的名称，并单击Start GSI（开始GSI）按钮即可。结果将会以图12-25所示的层级格式显示。

图12-25 GSI输出

注意，结果是以代表目标站点文件和目录的层级树的形式呈现的。可以单击每个链接来浏览正确的页面。

另外，你也可以在Firefox内右击，并且选择GSI命令。本例中，GSI将会发送基于选中文本填写的查询搜索。要是没有选中文本的话，GSI将会自动填写当前Web站点的名称。

GSI有几个备选的选项，见表12-2。

表12-2 GSI选项

GSI选项	说 明
Recursive Search	如果你选择使用递归查询（recursive search），GSI将会使用inurl搜索。例如，如果你选择在tankedgenius.com上进行Google Site Index操作。那么它首先会发送一个查询：site:tankedgenius.com。该查询会返回一个来自http://www.tankedgenius.com/blog/cp/index.html的结果。如果递归搜索的级别目前是1，那么它也会发出一个查询site:tankedgenius.com inurl:blog。接着，它会将来自该查询的结果添加到索引中。如果递归级别被设置为2，它也会发送一个查询site:tankedgenius.com inurl:cp并且获得结果
Full website names	默认情况下，GSI会显示一个首排缩进的站点索引，该索引中每个链接只显示了目录名。如果你愿意，你可以设置该选项以便它可以显示整个链接

注意

因为Google具有质疑GSI发送内容的天性，GSI会从Google那里得到403错误。在使用多个操作符查询时，通常都会见到这些错误。

12.4.8 Advanced Dork

Advanced Dork是Firefox和Mozilla浏览器的扩展程序，这两种浏览器都为右击的上下文菜单提供了Google Advanced Operator（高级操作符）的快捷使用方式。Advanced Dork由CP编写，可以通过访问https://addons.mozilla.org/en-US/firefox/addon/2144来下载。

与所有Firefox的扩展程序相同的是，安装很简单：单击地单击连接到Firefox内部的.xpi文件就可以触发安装。

Advanced Dork对上下文环境敏感——右击会基于右击执行的位置来调用Advanced Dork。例如，右击一个链接将会调用如图12-26所示的与链接相关的选项。

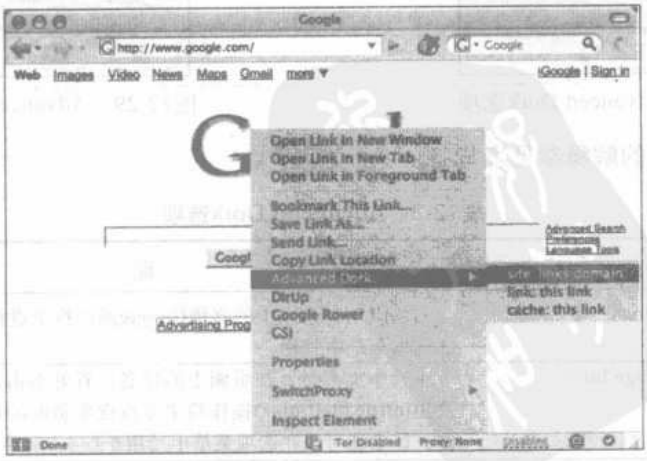


图12-26 Advanced Dork链接上下文菜单

右击高亮显示的文本会调用如图12-27所示的高亮显示的文本搜索模式。

图12-27 Advanced Dork高亮显示上下文菜单

该模式允许你在intitle、inurl、intext、site或ext搜索中使用高亮显示的单词。如图12-28和图12-29所示，几个卓越的选项可以用于Advanced Dork中。

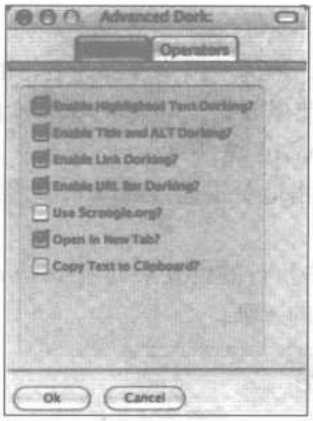


图12-28 Advanced Dork选项

图12-29 Advanced Dork操作

其中的一些选项的解释参见表12-3。

表12-3 Advanced Dork选项

选　项	说　明
Highlight Text Functions	右击以从15个以上的高级Google操作符中进行选择。该功能可以在选项菜单中禁用
Right-Click HTML Page Info	未选中文本时，在页面上的任意位置处右击，Advanced Dork会使用intitle和allintext操作符来重点搜索页面的HTML标题和ALT标签。该功能可以在选项菜单中禁用

（续）

选　项	说　明
Right-Click Links	右击链接即可从site: links domain，link: this link以及cache: this link之中任选一个操作。site: links domain将仅搜索域名，而不是完整的URL
Right Click URL Bar	右击URL栏（地址栏）并且从site，inurl，link和cache任选一个操作。Inurl只对文本高亮显示的部分有用。站点仅搜索域名，而非完整的URL

对于任何认真的Google用户而言，Advanced Dork都是一个高效的工具。你应该将它添加到你的工具库中。

12.5　从Google获取帮助

到目前为止，我们已经了解了各种检查你的站点是否有可能的信息泄露的方法，但是如果你检测到了这些泄露之后，该做些什么？首先也是最重要的，就是应该从你的站点中把泄露的信息删除。这可能是一个比较麻烦的过程，但是如果这么做了，你就能指出泄露源，能确保将来不会发生类似的泄露。信息泄露不是自己就产生的，它们也是发生某些事件带来的结果。找出相应的事件并解决它，这样你才能堵住问题的源头。Google有许多网页会从Webmaster的看法来帮助回答其中一些最常见的问题。www.google.com/webmasters的“Google Information for Webmasters”页面列出了各种常见问题的答案。

解决本地的问题只是成功一半。在某些情况下，Google有一份你的信息泄露的缓存拷贝正等着Google黑客去发现它。有两种方法可以删除页面的缓存版本。第一种方法是位于http://www.google.com/webmasters/tools/removals的自动URL删除系统。如图12-30所示，这个页面要求首先验证你的E-mail地址。尽管这像是登录Google账号，但是Google账号似乎不能提供访问权限。大多数情况下，即使你有一个Google账号，还是要注册。Google Groups账号似乎是个例外，利用这个账号好像没有什么问题就可以访问这个页面。

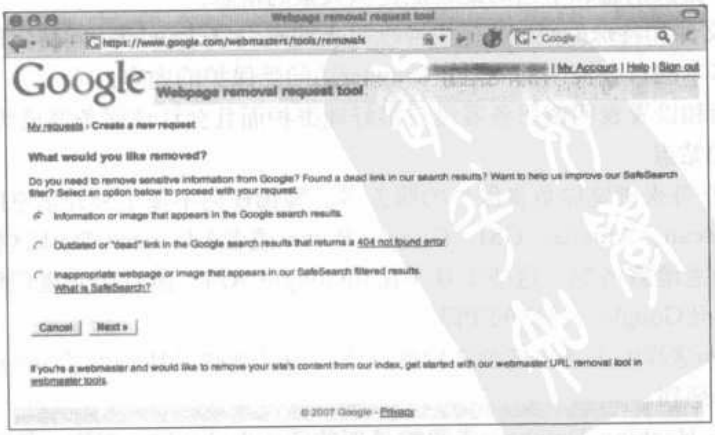


图12-30　Google的自动URL删除工具

该URL删除工具会向你提一系列的问题，以确认你是这些内容的所有者，并确定哪些是你要删除的内容。每个选项都十分明显，但是要记得的是内容删除的责任完全取决于你。你应该确保内容已经真正地从你的站点中删除，并且通过手动检查来继续URL删除过程。

12.6 总结

对任何一本书来说，Web服务器安全都是一个很大的主题。如此众多的各种需求加上各种不同的Web服务器软件、应用软件和操作系统软件，没有一本书能够全面地介绍这一主题。但是，至少有一些通用的规则能够帮助你防止恶意的Google黑客对你所保护的站点进行破坏性的攻击。

首先，要知道在发生异常事件时该如何操作Web服务器软件。缺失索引文件的目录列表以及特定的错误消息都会给攻击用的信息收集打开一条平坦大道。Robots.txt文件、简单的口令认证以及有效地使用META标记都能帮助你禁止Web Crawler访问网站的特定区域。虽然通常认为Web数据都是可公开的，但是当这些数据是以通用Hacking搜索的结果形式出现的，那么就可能会引起Google黑客对你的站点的兴趣。默认的页面、目录和程序能够暗示站点背后有一个水平不高的技术管理员。存在这类默认信息的服务器通常都是黑客的目标。要详细了解搜索引擎利用网站的哪些信息来吸引访问者。可以使用Gooscan、Athena、Wikto、GSI、Google Rower和Advanced Dork之中的任一种工具来帮助你利用Google搜索你的网站的信息泄露情况。如果找到了一个不应该是公开的页面，那么可以使用Google的删除工具从Google的数据库中删除该页。

12.7 快速查找解决方案

完善且坚固的安全策略

- 应该把强制执行的、坚固的安全策略作为所有安全保障措施的基础。
- 如果没有策略，那么你的安全防卫可能就会无效或者无法强制执行。

Web服务器安全防护

- 目录列表、错误消息和不当的配置能提供大量的信息。
- Robots.txt文件和特殊的META标记可以阻止搜索引擎Crawler访问特定的页面或目录。
- 即使是最基本的口令机制都能阻止Crawler访问受保护的内容。
- 默认的页面和设置表明该服务器没有很好地维护而且会让该服务器成为黑客的一个目标。

攻击你自己的站点

- 使用site操作符来浏览你负责保护的服务器。警惕任何不属于公开信息的页面。
- 使用像Gooscan、Athena、GSI、Google Rower或者Advanced Dork这样的工具来评估你的网站的信息泄露情况。这些工具不使用Google API，所以要知道任何滥用或者过分的行为都会导致Google封掉你的IP段。
- 使用像Wikto这样的工具也能攻击你的站点，而且这些工具使用了Google API，不用担心会被Google惩罚。
- 利用Google Hacking Database来跟踪最新的Google hacking查询。把GHDB和Gooscan、

Athena或者Wikto这样的工具结合起来使用。

从Google获取帮助

- 使用Google的Webmaster页面获取特殊的信息。
- 使用Google的URL删除工具从Google的数据库中删除敏感的数据。

12.8 网站链接

- http://johnny.ihackstuff.com Google Hacking Database（GHDB），搜索引擎hacking论坛，Gooscan工具和GHDB导出文件的主页。
- www.snakeoillabs.com Athena的主页。
- http://www.seorank.com/robots-tutorial.htm 使用robots.txt文件的优秀指南。

http://googleblog.blogspot.com/2007/02/robots-exclusion-protocol.html 有关Google的机器人程序策略的信息。

http://www.microsoft.com/technet/archive/security/chklist/iis5cl.mspx IIS 5.0安全检查列表。

http://technet2.microsoft.com/windowsserver/en/library/ace052a0-a713-423e-8e8c-4bf198f597b81033.mspx IIS 6.0安全最佳实践。

http://httpd.apache.org/docs/2.0/misc/security_tips.html Apache安全提示文档。

www.sensepost.com/research/aura Sensepost的AURA，它可以模拟Google的SOAP API 调用。

http://www.tankedgenius.com JeffBall和Cp的GSI以及Google Rower工具的主页。

https://addons.mozilla.org/en-US/firefox/addon/2144 Cp的Advanced Dork的主页。

12.9 常见问题

下面的常见问题，全部都由本书的作者们来回答，它们即可以用来测试你对本章所出现的概念的理解，也可以帮助你在现实生活中实现这些概念。如果希望作者解答你的问题，请浏览www.syngress.com/solutions，然后点击“Ask the Author”表单。

问：什么是no-cache pragma？它能阻止Google的服务器保存我的页面的缓存版本吗？

答：no-cache pragma是一个META标记，可以把该标记输入到一个文档中来命令浏览器不要把该页面加载到缓存中。这不妨碍Google的缓存功能。严格来说它只是客户端浏览器的一个命令。参见www.htmlgoodies.com/beyond/nocache.html获取更多的信息。

问：我很想知道更多的一些关于增强IIS安全性的细节，你能多介绍一些吗？

答：Microsoft提供了一个非常棒的IIS安全计划工具（IIS Security Planning Tool）。你可以在Google中搜索IIS Security Planning Tool。Microsoft也提供了一个IIS 5安全检查列表，同样可以用Google来搜索IIS 5服务检查列表。和IIS 6安全有关的极好的阅读资料可以通过查询“elements of IIS security”来找到。同时，也要经常访问IIS Security Center。可以用查询IIS security center来获得更多信息。

问：IIS的内容已经足够多了。那么和增强Apache服务器安全性有关的细节有哪些呢？

答：Securityfocus.com有一篇不错的文章，“Securing Apache:Step-by-Step”，可以从

www.securityfocus.com/infocus/1694获得。

问：检查我的网站的Goolge信息泄露情况的最好工具是哪个？

答：这个问题很难回答，其答案取决于你的需要。最直接的检查你的网站的信息泄露情况的方法是使用site操作符。例如查询site:gulftech.org会给出所有Google知道的gulftech.org上面的页面。通过浏览每个页面，你就会知道Google保存了你的网站的哪些信息。每一周都要做一次这种工作。

如果觉得这样做很麻烦，你可以考虑一种自动化的工具。比site技术高一步的工具是Athena。Athena能够读取GHDB的完整内容，允许你观察每个查询的执行情况，对每条查询添加site操作符。这允许你遍历整个“不好的搜索”列表以检查你的站点是否受到影响。Athena没有使用Google API，所以并不是真正意义上的自动化。Gooscan在使用不当时可能是最大的Google自动处理违反者了，因为它是基于GHDB的，而且会在相当短的时间内就处理完整个GHDB。它没有使用Google API，而且Google很有可能会提醒你在它允许的范围内使用该工具。不推荐这种用法，因为Google可能会把你当成“敌人”，但是如果慎重地使用并且遵守Google的no-automation规则，Gooscan可能就是最为精湛的工具了。