

4 FREE BOOKLETS

YOUR SOLUTIONS MEMBERSHIP



Secure Your Network for Free

**USING NMAP, WIRESHARK,
SNORT, NESSUS, AND MRTG**

Use Freely Available Tools to Get Big Dollar Security for Your Network!

- Learn How to Implement Best-of-Breed Security for Free
- Companion Web Site Contains Dozens of Working Scripts and Tools
- Special Case Studies of Free Intrusion Detection Systems, Vulnerability Scanning, Automated Network Device Inventory, Firewall Usage Statistics, and More

Eric Seagren

Wesley J. Noonan Technical Editor

VISIT US AT

www.syngress.com

Syngress is committed to publishing high-quality books for IT Professionals and delivering those books in media and formats that fit the demands of our customers. We are also committed to extending the utility of the book you purchase via additional materials available from our Web site.

SOLUTIONS WEB SITE

To register your book, visit www.syngress.com/solutions. Once registered, you can access our solutions@syngress.com Web pages. There you may find an assortment of value-added features such as free e-books related to the topic of this book, URLs of related Web sites, FAQs from the book, corrections, and any updates from the author(s).

ULTIMATE CDs

Our Ultimate CD product line offers our readers budget-conscious compilations of some of our best-selling backlist titles in Adobe PDF form. These CDs are the perfect way to extend your reference library on key topics pertaining to your area of expertise, including Cisco Engineering, Microsoft Windows System Administration, CyberCrime Investigation, Open Source Security, and Firewall Configuration, to name a few.

DOWNLOADABLE E-BOOKS

For readers who can't wait for hard copy, we offer most of our titles in downloadable Adobe PDF form. These e-books are often available weeks before hard copies, and are priced affordably.

SYNGRESS OUTLET

Our outlet store at syngress.com features overstocked, out-of-print, or slightly hurt books at significant savings.

SITE LICENSING

Syngress has a well-established program for site licensing our e-books onto servers in corporations, educational institutions, and large organizations. Contact us at sales@syngress.com for more information.

CUSTOM PUBLISHING

Many organizations welcome the ability to combine parts of multiple Syngress books, as well as their own content, into a single volume for their own internal use. Contact us at sales@syngress.com for more information.

SYNGRESS®

Secure Your Network for Free

USING NMAP, WIRESHARK,
SNORT, NESSUS, AND MRTG

Eric Seagren

Wes Noonan Technical Editor

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively “Makers”) of this book (“the Work”) do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, “Career Advancement Through Skill Enhancement®,” “Ask the Author UPDATE®,” and “Hack Proofing®,” are registered trademarks of Syngress Publishing, Inc. “Syngress: The Definition of a Serious Security Library”™, “Mission Critical™,” and “The Only Way to Stop a Hacker is to Think Like One™” are trademarks of Elsevier. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY SERIAL NUMBER

001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	49HLPWE43W
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY

Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Secure Your Network for Free

Copyright © 2007 by Elsevier. All rights reserved. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

1 2 3 4 5 6 7 8 9 0

ISBN-10: 1-59749-123-3

ISBN-13: 978-1-59749-123-5

Publisher: Andrew Williams

Acquisitions Editor: Gary Byrne

Technical Editors: Wes Noonan and Stephen Watkins

Indexer: Richard Carlson

Page Layout and Art: Patricia Lupien

Copy Editors: Michelle Melani and Audrey
Doyle

Cover Designer: Michael Kavish

For information on rights, translations, and bulk sales, contact Matt Pedersen, Director of Sales and Rights, at Syngress Publishing; email matt@syngress.com or fax to 781-681-3585.



Lead Author

Eric S. Seagren (CISA, CISSP-ISSAP, SCNP, CCNA, CNE-4, MCP+I, MCSE-NT) has 10 years of experience in the computer industry, with the last eight years spent in the financial services industry working for a Fortune 100 company. Eric started his computer career working on Novell servers and performing general network troubleshooting for a small Houston-based company. Since he has been working in the financial services industry, his position and responsibilities have advanced steadily. His duties have included server administration, disaster recovery responsibilities, business continuity coordinator, Y2K remediation, network vulnerability assessment, and risk management responsibilities. He has spent the last few years as an IT architect and risk analyst, designing and evaluating secure, scalable, and redundant networks.

Eric has worked on several books as a contributing author or technical editor. These include *Hardening Network Security* (McGraw-Hill), *Hardening Network Infrastructure* (McGraw-Hill), *Hacking Exposed: Cisco Networks* (McGraw-Hill), *Configuring Check Point NGX VPN-1/FireWall-1* (Syngress), *Firewall Fundamentals* (Cisco Press), and *Designing and Building Enterprise DMZs* (Syngress). He has also received a CTM from Toastmasters of America.

I would like to express my gratitude to several people who have helped me make this book a reality. First and foremost I would like to say thank you to Sandra and Angela, for their support, patience, and understanding during the entire process. I would like to thank Wes, for the quality and consistency of his constructive feedback. I would also like to thank Holla, for providing the original spark of an idea that eventually evolved into this book (specifically Chapters 2 and 7), and Moe, for being supportive when the opportunity presented itself.



Technical Editors

Wesley J. Noonan (Houston, Texas) has worked in the computer industry for more than 12 years, specializing in Windows-based networks and network infrastructure security design and implementation. He is a Staff Quality Engineer for NetIQ, working on the company's security solutions product line. Wes was the author of *Hardening Network Infrastructure* (McGraw-Hill) and was a contributing/coauthor for *The CISSP Training Guide* (Que Publishing), *Hardening Network Security* (McGraw-Hill), *Designing and Building Enterprise DMZs* (Syngress), and *Firewall Fundamentals* (Cisco Press). Wes was also the technical editor for *Hacking Exposed: Cisco Networks* (McGraw-Hill). He contributes to *Redmond* magazine, writing on the subjects of network infrastructure and security, and he maintains a Windows Network Security section called "Ask the Experts" for Techtarget.com (http://searchwindowssecurity.techtarget.com/ateAnswers/0,289620,sid45_tax298206,00.html). Wes has also presented at TechMentor 2004.

Wes lives in Houston, Texas.

Stephen Watkins (CISSP) is an Information Security Professional with more than 10 years of relevant technology experience, devoting eight of these years to the security field. He currently serves as Information Assurance Analyst at Regent University in southeastern Virginia. Before coming to Regent, he led a team of security professionals providing in-depth analysis for a global-scale government network. Over the last eight years, he has cultivated his expertise with regard to perimeter security and multilevel security architecture. His Check Point experience dates back to 1998 with FireWall-1 version 3.0b. He has earned his B.S. in Computer Science from Old Dominion University and M.S. in Computer Science, with Concentration in Infosec, from James Madison

University. He is nearly a lifelong resident of Virginia Beach, where he and his family remain active in their church and the local Little League.

Stephen was the technical editor for Chapter 3.



Companion Web Site



Much of the code presented throughout this book is available for download from www.syngress.com/solutions. Look for the Syngress icon in the margins indicating which examples are available from the companion Web site.

Contents

Chapter 1 Presenting the Business	
Case for Free Solutions	1
Introduction	2
The Costs of Using Free Security Solutions	2
Training Costs	3
Hardware Costs	3
Consulting Costs	4
Hidden Costs	5
The Savings of Using Free Security Solutions	6
Purchase Costs	6
Maintenance Costs	7
Customization Costs	7
Comparing Free Solutions with Commercial Solutions	8
Strengths of Free Solutions	9
Weaknesses of Free Solutions	10
Evaluating Individual Solutions	12
“Selling” a Free Solution	16
Selling by Doing	17
Presenting a Proposal	17
Summary	19
Solutions Fast Track	19
Frequently Asked Questions	21
Chapter 2 Protecting Your Perimeter.	23
Introduction	24
Firewall Types	24
Firewall Architectures	27
Screened Subnet	27
One-Legged	28
True DMZ	30
Implementing Firewalls	31
Hardware versus Software Firewalls	32
Configuring netfilter	32
Choosing a Linux Version	32

Choosing Installation Media	33
Linux Firewall Operation	36
Configuration Examples	42
GUIs	55
Smoothwall	76
Configuring Windows Firewall	85
Providing Secure Remote Access	86
Providing VPN Access	87
Using Windows as a VPN Concentrator	89
iPig	93
OpenSSL VPN	98
Providing a Remote Desktop	108
Windows Terminal Services	109
VNC	113
Using the X Window System	119
Providing a Remote Shell	125
Using Secure Shell	126
Using a Secure Shell GUI Client	128
Summary	130
Solutions Fast Track	131
Frequently Asked Questions	132
Chapter 3 Protecting Network Resources	133
Introduction	134
Performing Basic Hardening	134
Defining Policy	135
Access Controls	137
Authentication	137
Authorization	138
Auditing	138
Hardening Windows Systems	139
General Hardening Steps	139
Users and Groups	142
File-Level Access Controls	147
Additional Steps	152
Using Microsoft Group Policy Objects	153
Account Lockout Policy	159

Audit Policy	160
User Rights Assignment	160
Hardening Linux Systems	164
General Hardening Steps	164
Users and Groups	165
File-Level Access Controls	168
Using the Bastille Hardening Script	172
Using SELinux	173
Hardening Infrastructure Devices	175
Patching Systems	176
Patching Windows Systems	177
Patching Linux Systems	179
Personal Firewalls	180
Windows Firewall	180
Netfilter Firewall	187
Configuring TCP Wrappers	187
Providing Antivirus and Antispyware Protection	188
Antivirus Software	189
Clam AntiVirus	189
Using Online Virus Scanners	196
Antispyware Software	196
Microsoft Windows Defender	197
Microsoft Malicious Software Removal Tool	200
Encrypting Sensitive Data	201
EFS	202
Summary	209
Solutions Fast Track	209
Frequently Asked Questions	212
Chapter 4 Configuring an Intrusion Detection System 215	
Introduction	216
Intrusion Detection Systems	216
Configuring an Intrusion Detection System	217
Hardware Requirements	218
Placing Your NIDS	218
Configuring Snort on a Windows System	221
Installing Snort	222
Configuring Snort Options	225
Using a Snort GUI Front End	231

Configuring IDS Policy Manager	232
Configuring Snort on a Linux System	240
Configuring Snort Options	240
Using a GUI Front End for Snort	246
Basic Analysis and Security Engine	246
Other Snort Add-Ons	254
Using Oinkmaster	254
Additional Research	256
Demonstrating Effectiveness	257
Summary	258
Solutions Fast Track	259
Frequently Asked Questions	261
Chapter 5 Managing Event Logs	263
Introduction	264
Generating Windows Event Logs	264
Using Group Policy to Generate Windows Events Logs	267
Generating Custom Windows Event Log Entries	274
Collecting Windows Event Logs	275
Analyzing Windows Event Logs	277
Generating Syslog Event Logs	279
Windows Syslog	282
Generating Syslog Events	282
Receiving Syslog Events	295
Linux Syslog	297
Generating Syslog Events	297
Encrypting Syslog Traffic	298
Receiving Syslog Events on a Linux Host	311
Analyzing Syslog Logs on Windows and Linux	312
Windows Log Analysis	313
Linux Log Analysis	321
Securing Your Event Logs	327
Ensuring Chain of Custody	328
Ensuring Log Integrity	329
Applying Your Knowledge	331
Summary	333
Solutions Fast Track	333
Frequently Asked Questions	335

Chapter 6 Testing and Auditing Your Systems	337
Introduction	338
Taking Inventory	338
Locating and Identifying Systems	339
Nmap	341
Super Scanner	347
Angry IP Scanner	351
Scanline	352
Special-Purpose Enumerators	355
Locating Wireless Systems	357
Network Stumbler	358
Documentation	361
Network Topology Maps	362
Access Request Forms	364
Business Continuity and Disaster Recovery Plans	365
IT Security Policies/Standards/Procedures	365
Vulnerability Scanning	366
Nessus	367
Running Nessus on Windows	368
Running Nessus on Linux	371
X-Scan	375
Microsoft Baseline Security Analyzer	379
OSSTMM	382
Summary	386
Solutions Fast Track	386
Frequently Asked Questions	387
Chapter 7 Network Reporting and Troubleshooting	389
Introduction	390
Reporting on Bandwidth Usage and Other Metrics	390
Collecting Data for Analysis	392
Understanding SNMP	394
Configuring Multi Router Traffic Grapher	397
Configuring MZL & Novatech TrafficStatistic	400
Configuring PRTG Traffic Grapher	403
Configuring ntop	412
Enabling SNMP on Windows Hosts	418

Enabling SNMP on Linux Hosts	421
Troubleshooting Network Problems	424
Using a GUI Sniffer	425
Using a Command-Line Sniffer	433
Additional Troubleshooting Tools	438
Netcat	439
Tracetcp	439
Netstat	440
Summary	442
Solutions Fast Track	442
Frequently Asked Questions	444
Chapter 8 Security as an Ongoing Process	447
Introduction	448
Patch Management	448
Network Infrastructure Devices	452
Operating System Patches	453
Application Patches	453
Change Management	454
Change Causes Disruption	454
Inadequate Documentation Can Exacerbate Problems	455
Change Management Strategy	455
Antivirus	459
Antispyware	459
Intrusion Detection Systems	460
Vulnerability Scanning	460
Vulnerability Management Cycle	461
Roles and Responsibilities	463
Penetration Testing	463
Obtaining the Support of Senior Management	464
Clarify What You Are Buying	464
Policy Review	465
Physical Security	466
CERT Team	468
Summary	470
Solutions Fast Track	470
Frequently Asked Questions	472
Index	475

Presenting the Business Case for Free Solutions

Solutions in this chapter:

- The Costs of Using Free Security Solutions?
- The Savings of Using Free Security Solutions?
- Comparing Free Solutions with Commercial Solutions
- “Selling” a Free Solution

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

You may be looking for inexpensive ways to solve a security problem and want to know more about the free tools that are available. This book will guide you to some of the best free solutions. In some environments, taking the initiative and implementing any type of security measures can get you in trouble; even with the best planning, problems can arise. This chapter will help you gain the support you need in order to implement a cost saving solution.

Whether you are the person implementing the changes and need to “sell” the solution to your manager, or you’re the person making the decisions and need to understand the true implications of a particular “free” solution, this chapter will help you find solutions to your security problems. This chapter discusses some of the hidden costs associated with free solutions and clarifies what comes from those solutions. This chapter also addresses the fact that in most cases, an apples-to-apples comparison between a free package and a commercial product is not feasible. With all of this information, you should be in a good position to propose a solution and back up your choice with some compelling business arguments.

The Costs of Using Free Security Solutions

In the case of security solutions, few things in life are free. And while you may not pay for a security solution itself, there are costs associated with implementing a solution that are not obvious. In most cases, your security needs dictate which solutions are appropriate; if there is not a free solution available, you have to use commercial tools. Fortunately, there are a lot of high-quality free solutions available. The cross section included in subsequent chapters is aimed at providing a spectrum of solutions with a variety of sophistication levels. If you dive headlong into implementing a free solution without adequate knowledge and research, it could end up costing you more than if you had purchased a commercial solution.

Training Costs

Training costs are one of the biggest expenses when it comes to implementing a free solution. First are the direct training expenses (e.g., sending someone for classroom instruction). Your options may be limited when it comes to training for free software solutions. In most cases, training does not exist in a focused format (i.e., you probably won't find a class on netfilter firewalls). Instead, you may be able to find applicable training indirectly, such as in classes on general Linux use or administration.

Another training cost is materials (e.g., books). Aside from this book, there will likely be areas where you want more specialized information. For example, if you are implementing a Snort intrusion detection system (IDS), this book walks you through setting up Snort. A small library covering the specific software you have deployed is a worthwhile investment.

You will also incur training costs, such as not having access to an employee during training. This time away from work is an expense, because you are paying for an asset that isn't available. The same is true if the employee is on-site and "self-training."

Hardware Costs

A security appliance is a device that doesn't require a computer and is only used for its intended purpose, while all of the free solutions require a system to run on. Luckily, the requirements are usually minimal; therefore, you can often use an old PC. However, connectivity requirements could make using the system in a nondedicated configuration a security risk. Rarely does a system require enough resources to make using the same host for any other function impractical (e.g., the Snort IDS logging capability can quickly eat up disk space, leaving little to no resources for other programs).

If there are no old systems available, there are many online retailers offering older systems at affordable rates. A large portion of the cost for low-end PC's is often for the operating system. Many retailers offer affordable systems that either include Linux as the operating system, or come without an operating system installed. These allow you to purchase a relatively modern

system cheaply, and then install your own OS on it. This can be a viable option for running security tools and providing user workstations.

Consulting Costs

You must carefully weigh and balance where you spend your money. Too little training and you will end up hiring consultants. Implementing, configuring, or fixing your free firewall can cost a lot, more than if you had bought a firewall. With small commercial firewalls costing around \$500.00, it doesn't take long before free isn't so free.

With that said, don't be afraid to call a consultant if necessary. Having a well-paid consultant configure your free solution and make sure that it's implemented using best practices is a steal compared to implementing some proprietary solutions. A consultant can also act as a trainer. You can shadow the consultant and see how and what is being done, and you can ask questions and learn why things are done a certain way. In this way you can have your solution set up by someone who is knowledgeable and experienced, and provide training and guidance to the in-house personnel.

If you have ever had to rely on consultants, you probably know they are not always a "good buy." Sometimes they are not as knowledgeable as you were led to believe. The key is to communicate with the consulting firm, being very clear about what your needs are. A good consultant can save the day.



WARNING

You should always be careful when cutting consulting budgets. I have seen attempts to save money end up costing more. In almost all cases, getting a consultant in quickly is the best course of action and the most cost effective in the long run. If you find a skilled consultant you like, a monthly retainer might be a good investment.

Hidden Costs

What are all the costs of a free solution? For starters, power consumption. I had a Windows 98 system that was only being used as a print server. It occurred to me that the PC cost me approximately \$7 per month in electricity. With a dedicated print server costing only about \$30.00 and using virtually no electricity, I would save money within five months by buying the print server. The Pentium II running Windows 98 was technically “free,” but paying for electricity to keep it running was not the most cost-effective choice. Some security tools are not offered as a commercial appliance, and some are (e.g., small, low cost firewalls that use far less power than a standard desktop PC are available from several manufacturers). Your cost for electricity will vary. Based on your electric bill, you can calculate with a high degree of accuracy what a given device costs.

Another consideration is heating, ventilation, and air-conditioning (HVAC) costs. HVAC is basically the climate controls. Additional computers create additional heat, which costs more money for air conditioning. The same considerations apply as for power consumption. If a stand-alone appliance is not an option, the additional HVAC requirements are an unavoidable cost; however, in those cases where more efficient appliance-based solutions exist, they almost always produce less heat than a normal workstation. This also applies to the difference between an older computer and a newer computer. Newer systems that demand more power and cooling when they are being heavily utilized often incorporate energy-saving characteristics that are superior to those of the older systems.

There is also the cost of real estate. A decommissioned full-sized tower PC takes up a lot more space than a new commercial appliance the size of a cigar box. You may have plenty of room now, but as the server room gets more and more crowded, space could become an issue. A keyboard, video, and mouse (KVM) switch might save more in space than it costs to buy. As the servers become increasingly tightly packed, good air flow and adequate cooling will be inhibited, and physical access to the systems for operation or maintenance will also be difficult.

Inefficiency is another cost of free solutions with respect to the fact that the support staff are likely unfamiliar with the new free solutions. When a

staff member performs a task on a new firewall, it takes longer to do than if they are familiar with the firewall. This inefficiency typically costs only the time to complete a task; however, if an outage or business disruption occurs, this delay could result in lost profit or business. These delays must also be accounted for when planning projects and other activities.

Free solutions are usually produced by small organizations or by an individual. These solutions may do an excellent job in their assigned roles, but may not be well known. This could be a liability if the individual who configured your free solution leaves or is otherwise unavailable. If you have a PIX firewall that needs work, you probably would not have a hard time locating a resource. On the other hand, if you need someone to take over the administration of an obscure free solution, finding someone could be difficult. This difficulty could manifest itself as a hidden cost by increasing the delay before a problem can be addressed, having to pay a premium for a consultant, or any number of other inefficiencies.

The Savings of Using Free Security Solutions

The following section discusses how a free security solution can save you money. The primary savings is obvious: you didn't pay for the product; however, there are additional benefits. This section offers a detailed look into the benefits of using free software. By evaluating the expected savings and costs, you can form a more practical, accurate picture of what will be gained by implementing a free security solution.

Purchase Costs

The purchase cost is one of the single largest cost savings of using free software. The best example of this is with firewalls. A small Linksys or Netgear firewall costs around \$20.00 to \$50.00. They use almost no power, support port forwarding, perform Network Address Translation (NAT), act as a Dynamic Host Configuration Protocol (DHCP) server, and are stateful packet filters. Suppose you use Linux and netfilter to run a firewall for free. Odds are it will cost more to pay for the employee's time to set up the Linux firewall

than the Linksys would cost to buy. Firewalls are one of the best examples of how readily available affordable commercial solutions can be.

You can still save money on purchases. Some types of products, particularly IDSes, network analysis and reporting tools, and commercial virtual private network (VPN) solutions can cost staggering amounts of money. When comparing prices, come as close as possible to comparing like products. Using the most expensive “deluxe” software suite available as the price for decision making is misleading. The free solution will not have the same features and capabilities as the commercial version. Look at the features you think you need as a starting point for which commercial products would be viable options. Use the costs of those products as your basis for determining what the free solution will save you.

Maintenance Costs

Maintenance can be expensive; it is not uncommon for a yearly maintenance contract to cost 10 percent of the purchase price. This price will also fluctuate, as almost all vendors have various support tiers with varying response times and service level agreements (SLAs). The reality is, however, if you opt for the free solution and spend the 10 percent on training instead, you would probably have a very high level of responsiveness from your own in-house staff. Ensuring an equivalent level of responsiveness and availability from the vendor would likely cost you a large sum. Your own support staff could probably go to the office or address the issue remotely far more quickly than all but the largest and most well-established vendors. Even if a vendor can have someone on site in two hours, sometimes getting a live person to return your call and schedule the emergency appointment takes time. You can probably reach your own staff as quickly, if not more so. The level of service you expect should be factored in when estimating the cost savings available by not having to purchase a maintenance contract.

Customization Costs

Customization is an area that can offer huge gains or be inconsequential, depending on your circumstances. If you purchase a commercial product, you may find that there is no way it can be customized for your environment. If

some degree of customization is available, it is rarely free. Often, the hourly rate for such services is at a premium, the assumption being you must really want or need the desired functionality if you are willing to pay to add it. With some free solutions, this customization can be affordable, or even free, if you have the expertise. However, not all free software is customizable. Just because it's free does not always mean it is open source. Open source software is software where the source code (i.e., the programming code used to make it run) is freely available. When software is open source, you can download the source code and edit it to your heart's content. You can add as few or as many custom features as you want.

Obviously, this is an advantage that not everyone will need or have the means to take advantage of. Depending on the software package in question, some are programmed using different programming languages, so even if you have a resource who knows enough to be able to customize the program, they might not know the particular programming language that is required. Customization is also something you don't know you need until you are well into the implementation phase. If you know your customization needs ahead of time you can investigate and weigh the costs accordingly. Generally speaking, even if the cost is the same to customize the free solution as a comparable commercial solution, the level of customization that is possible is often (but not always) equivalent or better with the free solution.

Comparing Free Solutions with Commercial Solutions

When it comes to making an informed decision as to whether to purchase a commercial solution or implement a free solution, there are some additional non-dollar-related considerations to take into account. First and foremost, compare like functionality. Don't compare the deluxe version of the commercial product to the free version; they won't have the same features or learning curve, or require the same hardware. Ultimately, by making the most informed and well-reasoned comparison possible, the best solution will be chosen.

Strengths of Free Solutions

One advantage free solutions often have over their commercial counterparts is that of development speed. This varies from one product to another; not all free products have quick development cycles. The open-source packages often have very fast development cycles and can address the latest security issue more quickly than their commercial counterparts. If you want to stay on the cutting edge, free software (especially open-source software) might be a better path than commercial solutions.

Previously, we discussed customization as a cost savings with some free software. This is because often you can do the customizing yourself instead of paying the vendor to do it for you. Customization is worth mentioning as a strength of its own, above and beyond the cost savings. Again, not all free software is customizable. Sometimes the best software in a particular category uses closed code and there is no way for you to perform any customization. But one of the greatest strengths of the open-source movement is that everyone has the freedom to edit, customize, and improve the software.

A potential strength of free solutions is the speed with which they can be implemented (which is different than the development speed). When I speak of the implementation speed of free software I am referring to the time it takes to get the software loaded and working. This includes not only installation, but also the red tape sometimes involved in making significant purchases. For example, suppose you are trying to form a business partnership that will be beneficial to your organization. The nature of the arrangement is such that time is of the essence; the sooner the partnership is completed the better. The partnership involves network connectivity to facilitate the exchange of information. After reviewing the plans of how it would be done, your potential partner is hesitant to go through with it, because you lack adequate firewall protection. Maybe your current Internet connection is filtered with a consumer-level home router/firewall and you need a separate demilitarized zone (DMZ) with some advanced NATing rules and better logging. You could contact a vendor, wait for a response, get a quote on the price, and pass that to your manager for approval. After your manager approves the purchase, you hand it to accounting and they make the purchase and arrange shipping. Once it arrives, you must install and configure the new firewall and then test

it. A faster approach would be to grab the old PC from the closet, download and install Linux on it, and configure the firewall. If your environment allows it, implementing the free solution could be much faster. In environments where there are restrictions on permitted vendors, permitted software, permitted hardware, and so on, getting approval for a free solution could be more difficult and time consuming than a commercial solution. Ultimately, your environment will dictate whether implementation speed can truly pan out as an advantage or not.

You might think that all free software is produced by some kid after school and will be unstable and lacking the quality control of a commercial software development project. While this is certainly true some of the time, at other times it could not be farther from the truth. The fact is that the larger, well-established open-sourced projects can have hundreds of programmers reviewing, revising, scrutinizing, and modifying the code. Very few commercial companies have the same amount of resources to put into a single software product. This means that in many cases you are getting software that has been through more peer review and testing than the commercial equivalent. This is not always true; in many cases the free software has very little quality control and you, as the user, are really doing the testing. Basically, this means that the quality of free solutions will have a lot of variance. To increase the odds that you are not trying to implement buggy software, do your homework. If you stick to mature products that have a proven track record you will certainly improve your odds. Avoiding new releases that implement major architectural changes may help as well. If the current release of a product you are using incorporates newly added support for the latest chipset, it might be wise to wait for that release to be tested a little more before deploying it in your environment. For an excellent and lengthy article on the merits of free software, refer to www.dwheeler.com/oss_fs_why.html. In reality, some of the free offerings are not fit to be run in any sort of critical role, while others can do so with aplomb. Ultimately, not all free software is “cheap” software; some of the free offerings are of very high technical quality.

Weaknesses of Free Solutions

The single biggest drawback to implementing a free solution in a production environment is one of support, or lack of support. When you download

something for free from the Internet, there is generally no phone number to call and ask questions. This is sometimes mitigated by high quality documentation, and in some cases extensive online user forums where you can ask questions and receive help from the creator of the package or other users. On the other hand, high-quality documentation is the exception rather than the norm, and many of the free utilities have little in the way of documentation. This consideration is one of the biggest concerns for management. Generally speaking, the more mission critical the role of the security software is, the more hesitant you should be about implementing a solution with minimal support. If you are a company that depends on the Internet, you should require a higher level expertise from in-house technical staff before implementing a free Linux firewall, compared with another company that makes money in a storefront and only uses the Internet to surf the Web. This isn't to say that the support cannot be adequate with free software or that you shouldn't use free solutions to fulfill critical needs, only that you need to do so knowingly and after careful consideration and planning.

The management capabilities of free software solutions are typically not as robust as they are with commercial offerings. Your particular product will determine if this is a real consideration or not. Most often the presence or absence of management capabilities is more noticeable with free IDSes, antivirus, and antispyware offerings. The common denominator here is that these products require frequent updates in order to maintain their value and do their job effectively. An enterprise-class antivirus program will offer a lot of control and features around signature updates, such as when and how to perform the updates and how to handle things when a virus is detected. The free solutions are generally more limited, often requiring the scanning or updating process to be performed manually, and responding to a positive detection may have to be an interactive process, rather than an automated one.

Another area where the free solutions are also sometimes lacking is reporting. While some offer excellent reporting, many others offer little to no reporting capability. In most cases, you will be able to manually configure some type of reporting on your own using freely available utilities. Even if you can arrange for some automated logging or reporting to be generated, it won't be as simple or quick as it would be if it were a commercial product

that supported that functionality natively. As you begin considering free solutions, you will want to also consider not only the logging capabilities you *want*, but those you *need*. In many cases, if you are in a highly regulated industry, such as banking, or healthcare, the lack of adequate logging capability is the determining factor that leads to a decision to go with commercial software. If you have auditors you need to satisfy, you will want to research the audit trail you will be able to generate carefully, before coming to a strategic decision on your solution.

Previously, we touched on the fact that the free solutions are often not well known, and how this can translate into a hidden cost in consulting fees. This liability can go beyond consulting fees. If you were hiring a new employee and specified that they need to know Cisco equipment, you could undoubtedly find someone in short order. If you specified you wanted them to be familiar with some little-known free solution you have implemented, you could have a very hard time finding someone. That's not to say that they couldn't be trained, but again, there are costs and disadvantages associated with that. The familiarity (or lack thereof) could also cause the time it takes to implement a solution to be longer than with a more widely understood technology. Speed of implementation was mentioned as a potential asset, but it can easily be a liability if there is no one available who understands the solution. Ultimately, there are advantages to using industry standard solutions over less widely deployed offerings.

Evaluating Individual Solutions

As you do your research, you will need to determine if the free solution is the best solution. There are a whole host of factors which will go into making this determination. The following list briefly summarizes the steps needed to make a determination as to whether or not a free solution is the best solution for you.

1. **Identify Your Options** This can be the hardest part of the process, knowing what free alternatives exist. Hopefully this book will help, but there are also on-line sites to help you find free software. One of the largest sites housing open source software is <http://sourceforge.net/index.php>. Also check out

<http://freshmeat.net/>. You can find a more programmer-oriented site containing only software that runs on Linux at www.icewalkers.com/. A directory of free software is located at <http://directory.fsf.org/>. A similar directory of free software for Microsoft Windows is located at <http://osswin.sourceforge.net/>. Finally, a CD containing some “top picks” of free software for use on Windows is located at www.theopencd.org/.

2. **Research Each Option** Typically, this will mean doing searches on the software. Take note of how many problems people have, and if they have been fixed. Check the developer’s Web site and documentation. See if the documentation is well-crafted and complete. This is when you will weed out the majority of candidates and hopefully be left with a list of quality choices.
3. **Compare Products** The previous step is meant to sort out the best free solutions. This step is aimed at comparing the best free solutions against their commercial counterparts. This is where you may rule out some products as too expensive or too hard to use. Metrics to use for comparison include:
 - **Functionality** The product must meet your business needs to be considered. Pay attention to volumes. The product might do what you want, but not on the scale you want it to. Consider if the product will work with other utilities or if it uses proprietary and closed source methods, protocols, or algorithms. These traits may act as limiters and hinder flexibility later on.
 - **Cost** This is one of the major reasons you are considering a free solution. Try and be as accurate as possible in your estimates of the true costs, including things such as purchase cost, maintenance, training, upgrades, and so on.
 - **Momentum** How well established is the product? Remember this is a consideration for free software and commercial software. The more well established the software is, the better the odds the creators will be around in the future. A larger more well-established project will also likely have better community

support and reliability. Included in the overall momentum is to look at how active the project is. You don't want to invest a lot of time and energy in a product that is likely to just die off and fade away.

- **Support** What does support cost? Is it available? How timely is the support? What format does support take (online forums, e-mail, phone, and so on).
 - **Performance** Which solutions are the best performers? This includes speed, efficiency, and reliability. A powerful software package that crashes every hour isn't a viable option.
 - **Usability** Is the product user friendly? If the learning curve is very high, your training costs will rise. If the product doesn't have a feature or function you like, can you customize it and make it more user friendly?
 - **Security** Even for a security tool, you must consider the security implications. Is the product secure? Will it be handling secure data? Are you opening up any new security risks? What type of auditing and logging can it produce?
 - **Legal and License Issues** Be sure and review the license agreement closely. Many times the free software is not free if you are a business, or there are special restrictions on the number of installations or other criteria. When in doubt have your legal counsel review the license agreement for you.
 - **Individual Criteria** These are any special needs or requirements unique to your environment. What's good for other organizations might not work for you.
4. **Perform Detailed Testing** At this stage, you have hopefully narrowed the playing field down to just one or two selections. It's time to put these products through a real test and see if they do what their manufacturers claim they do. This can be done in a lab or possibly on the production network, depending on the risks involved and the nature of the product. You will need to evaluate how best to perform your detailed testing based on your circumstances.

5. **Come to a Conclusion** After all this research, you can make a decision on what you think the best solution is. Whether you are the final approver or you need to forward your recommendation to someone else for approval, at this point you should have all the facts collected in order for a good decision to be made.

Remember, the preceding steps leave a lot of room for flexibility. They may be performed in a more or less structured fashion. You might not formally cover all the steps, but in one form or another, those steps should occur. The more thoroughly you document the steps, the more you will be in a better position to justify your choices.

Now that we have discussed the many ways that the cost of a free solution may be higher or lower than the commercial equivalent, let's look at an example. Suppose your manager wants you to provide a reporting mechanism to see who is using the majority of the Internet bandwidth. Your manager also wants to know what the user(s) in question are using the bandwidth for. You search around and learn about a product called *nGenius Performance Manager*, which is made by Netscout (www.netscout.com/products/pm_home.asp). According to your research, it will do what you want and more. The graphs and charts it can produce are excellent, and it provides an extremely granular look into the traffic flowing across your network. In the free department, you've looked at *ntop*, and it seems pretty neat, not as granular, but still offering a respectable amount of data and reporting for free. You call up netscout and get some list pricing for the nGenius equipment. The server licenses have a scaled price structure according to the software's capabilities, so you inquire about the most economic server license they offer, which is \$20,000.00 list. You will need at least a single probe to sniff and collect data, which is another \$5,000.00. You will need to run this on a server, and the old one probably won't work, so there's another \$2,000.00. The yearly maintenance contract will be 10 percent of the purchase price, meaning another \$2,500.00, bringing the grand total to \$29,500.00, less any price breaks from list you might get.

If you then went to your boss and used the \$30,000.00 price tag to justify why you should implement a free traffic reporting and analysis tool, your presentation wouldn't be telling the whole story. First off, none of the free prod-

ucts come close to the power and functionality of nGenius, so you are not comparing like products. There are other less expensive alternatives, which would represent a much more accurate comparison to use as a cost savings example. Second, even if money were no object, deploying an enterprise-class solution like nGenius is probably not the best choice. Along with the impressive array of features comes a fairly steep learning curve. After implementing such a solution, your in-house staff might have more difficulty learning how to use it than they would with one of the free (and simpler) solutions. Third, you may not need the level of detail and sophistication that nGenius offers. If *ntop* or a comparable free solution can offer all the reporting and metrics that you are looking for, deploying a more complex solution may not be wise. *ntop* may be the best choice for your organization, but presenting that choice as a \$30,000.00 cost savings is far from accurate.

nGenius is the Cadillac of network analysis tools. It has a staggering array of features and an impressive level of customization you can perform without getting into actual programming. If I had the budget and the need, it is the product I would use. That being said, is *ntop* just as powerful? Not even close. But, in a small organization, the added features *nGenius* has to offer would likely never be used. With a price tag of free, *ntop* or one of the many other free alternatives is likely to do everything that is needed, and with a much smaller learning curve.

“Selling” a Free Solution

If you are in a position where you can implement a new security solution without having to receive anyone else’s approval, you probably don’t need to read this section. If on the other hand, you have to get someone to sign off on your plan, this should be helpful. If you do need approval, you are basically going to try and “sell” your solution, much like a salesman, highlighting the benefits, and realistically noting any disadvantages to your proposed solution. Remember, the objective of presenting a solution is not to “win” by getting to do things your way. The objective is to provide the decision makers with the most complete and accurate information so that the best decision can be made. Your own judgment of the environment and your target audience will play a large part in what constitutes the best approach for you to take. We

hope some helpful guidelines as to how to approach gaining approval can help improve your odds of success.

Selling by Doing

One method of demonstrating the power and effectiveness of a given solution is to actually demonstrate the solution. If the environment allows, and you have the resources, it might be feasible to install the software in question, generate the reports, and present the facts, along with a demonstration of what the software can do. You don't want to do anything that is inappropriate; if the change control procedures don't allow such spontaneity, you will need to revise your approach. Assuming you have the freedom to do so, saying the software generates graphs and reports and traffic usage, broken up by protocol and the computer in question, rarely has the same impact as seeing that same graph. Not only does it provide factual real evidence of the utilities value, it also demonstrates your initiative and forethought.



WARNING

Let's be perfectly clear here; we're not advocating that you go and implement some solution without proper management approval when policy says you shouldn't. You need to evaluate your environment and factor in things such as climate, policy, risks, and benefits to determine if it's wise to implement something without getting all the proper approvals ahead of time. Again, in some environments this would be perfectly okay, and your manager would be elated at your ingenuity and initiative, while in others you could end up looking for a new job. As always, exercise good judgment and when in doubt take the conservative approach.

Presenting a Proposal

If you do not have the luxury of implementing something and then asking for "approval," you will need to create a proposal with all of the relevant information. You can certainly do both, including the sample data from the utility in the proposal. The truth is, "presenting a proposal" sounds very

formal, and it can be, but it doesn't have to be. Some organizations have much more formal procedures in place than others. Presenting your proposal may be as structured as using a standardized template with forms to fill out and submit, and meetings with PowerPoint presentations. It could also just as easily mean talking with your manager over lunch and telling him what you would like to do.

Regardless of the format you employ for your proposal, there are certain common elements you will want to touch upon, verbally or on paper. If you address all these issues as accurately as possible, the odds of your venture being a success should be greatly improved. At a minimum, try and have information and answers covering the following areas concerning your proposed solution:

- **Costs, Training, and Implementation** What will it take? How much will it cost? How long will it take to implement? How much training will be required and of what type? How much will the training cost, and how long will it take? What hardware might be needed, and what will it cost? Will it impact the user experience? If so, how?
- **Performance** What will it do? What are the real capabilities, not just sales hype? Generating actual samples from your environment, or if you can find something online, would go a long way here. Hard data is always better than a sales blurb. What are the technical limitations?
- **Assumptions** What other factors must be in place for this to work as planned? Will you need assistance with the implementation? Will an outside consultant be needed?
- **Caveats** What are the drawbacks? What makes your solution less attractive? What are potential problems that might arise?

Summary

Not all facets of implementing free security solutions are free. There are always costs of one type or another, which vary in magnitude and relevance based on your individual circumstances. Ultimately, you don't want to be yet another person who fell victim to the open-source or freeware hype. These are the people who read or heard about a "free" product and rushed to implement it without doing adequate research, thus ending up with a mess that is expensive to make work or to clean up. With the proper research and planning, free solutions can provide you with some very powerful security solutions without spending a lot of resources. The real value lies in finding free software that is the simplest solution available that can still meet your needs.

Solutions Fast Track

The Costs of Using Free Security Solutions

- ☑ Training costs can quickly skyrocket, especially for classroom-based training.
- ☑ Consulting costs are not always something to be avoided. At times they can provide a very efficient way to implement a given solution while at the same time providing some sorely needed training and documentation.
- ☑ Intangibles can also add up. While items such as HVAC, power costs, and space requirements are not likely to break the bank, these are still considerations you should be aware of in order to make informed decisions.

The Savings of Using Free Security Solutions

- ☑ The biggest savings is that there are no software costs.
- ☑ No maintenance costs

Comparing Free Solutions with Commercial Solutions

- ☑ You can usually implement a free solution quicker than a commercial product, based on the time it takes to make and receive the purchase.
- ☑ A free solution's primary weakness is support. Without a toll-free number to call, you are left to educate yourself or pay someone with the appropriate skills to assist. The often sparse or non-existent documentation can sometimes be a major hindrance to a successful implementation.
- ☑ Many of the free solutions are also open source, allowing you unequal flexibility to customize, alter, change, or even rewrite the software in question.

“Selling” a Free Solution

- ☑ Be informed of the pros and cons of the solution, and be honest about your data. Remember that it's not a contest to implement a particular solution, but rather the objective is to be well informed so that the best solution can be chosen.
- ☑ Real life examples are always better than theory. A sample graph of data from your current network (policy allowing) is always going to drive home the point better than a bullet that says the product will produce the graph.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: How do I know when I have found the best solution?

A: The solution that is “best” today might not be tomorrow. The selection of free software is rapidly changing. While there are certain leaders who will likely continue to be top picks for the foreseeable future, many other free solutions will come and go. The only way to make a good decision is to “do your homework,” and if possible, consult an expert in the area you are interested in.

Q: If some of these free tools are so good, why doesn't everyone use them?

A: In the case of a large organization, the features or functionality the free solutions lack are vital, so a commercial solution may be the only option for some. For smaller organizations for which a free solution can satisfy their needs, it typically comes down to not knowing what the options are. No one is paying to advertise free products in computer magazines, so generally only the more experienced and knowledgeable information technology (IT) people know about all the available products.

Q: Is free software really free?

A: Not in every sense of the word. While the software itself may cost nothing, you have to consider the costs of the hardware required to run the software, the training required to implement the software, and the potential maintenance costs (in terms of man-hours and actual dollars) when considering a free solution. After you add all of this up, some free solutions can be very “cost-effective,” even if not truly free.

Protecting Your Perimeter

Solutions in this chapter:

- Firewall Types
- Firewall Architectures
- Implementing Firewalls
- Providing Secure Remote Access

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

When it comes to securing networks, the first items that come to mind are firewalls, which are the primary gatekeepers between an organization's internal network and the outside world. While a properly implemented firewall can be one of the most effective security tools in your arsenal, it shouldn't be the only tool. The adage "defense-in-depth" means that you should have multiple layers of security. Using a defense-in-depth configuration, if one component of your defense failed or was defeated, there would still be a variety of other fallbacks to protect your network. With the availability of increasingly affordable firewalls such as the popular Linksys cable/digital subscriber line (DSL) router, using the free firewall alternatives may not be as attractive for some. With a little effort, however, you will find the free alternatives are more configurable, allowing greater flexibility and control than the "home office" grade offerings.

This chapter focuses on securing your network perimeter. Remember that although the most common way to implement a firewall is between an internal network and the outside world (often the Internet), you should not limit yourself to placing firewalls only on the network edge. A firewall should be in any place you want to restrict the flow of traffic. With the current trend of security breaches originating from the inside of the network (often employees or ex-employees), companies are increasingly relying on firewalls to isolate and filter traffic between portions of the internal network.

This chapter reviews some basic firewall concepts and briefly discusses the different architectural ways to implement a firewall. Most of this chapter discusses the installation and configuration of free firewalls to run on both Windows- and Linux-based systems. Finally, once the network edge has been adequately secured, we discuss how to create controlled, secure paths through the perimeter for remote connectivity, including administrative access or remote office/work from home scenarios.

Firewall Types

No discussion of firewalls would be complete without a discussion of the different types of firewalls. This is particularly true in this context, because it allows you to better understand exactly where in the spectrum the free firewall

offerings lie. In the networking sense, a firewall is basically any component (software or hardware) that restricts the flow of network traffic. This is a sufficiently broad definition to allow for all of the various ways people have chosen to implement firewalls. Some firewalls are notoriously limited in capability and others are extremely easy to use.

Within the realm of firewalls there are many different ways to restrict network traffic. Most of these methods vary in the level of intelligence that is applied to the decision-making process. For example, to permit or deny traffic based on which network device is the sender or recipient, you would use a *packet-filtering firewall*. In reality, even the simplest packet filtering firewalls can typically make decisions based on the source Internet Protocol (IP) address, the destination IP address, and the source and/or destination port number. While this type of firewall may sound overly simplistic, consider if you have a server running a Web site for use on the Internet. In all likelihood, the only traffic that you need to allow to the server uses a destination port of Transmission Control Protocol (TCP) 80 or 443; thus, you could configure your firewall to permit only that traffic. These ports are used for HTTP and HTTPS, respectively. Because the server is available for the Internet, you can't filter traffic based on the source address or source port, which will be different for each connection.

The primary drawback with a simple packet filter is that the packet-filtering firewall has to rely on very primitive means to determine when traffic should be allowed (e.g., synchronous [SYN] or acknowledgement [ACK] bits being set). While this was adequate in the early days of the Internet when security was not as big of a concern, it won't work any more. It is trivial to set the bits on the packet using freely available software to make the traffic look like it is a reply to another connection. Thus the *stateful inspection firewall* was born of necessity. This type of firewall monitors all connections (inbound or outbound), and as the connection is permitted (based on the firewall's configured rules) it enters this connection into a table. When the reply to this connection comes back, even if the reply uses a port that the firewall was not previously configured to permit, it can intelligently realize the traffic is a response to a permitted session and permit the traffic.

Unfortunately, as the firewalls get better so do the methods hackers use to circumvent them. Suppose you have configured your firewall perfectly and

there are no holes: every permitted port is one you expressly want to allow. Using the previous example, no traffic is allowed to the Web server except Web traffic. Sounds good, but the problem is, if the firewall is completely secure, the server might not be. Flaws in the Web server software could allow the attacker to send the server an HTTP request that is 10,000 characters long, overflowing the buffers and allowing the attacker to execute the code of his choice. The packets used to transport the 10,000-character HTTP request are all legal TCP packets as far as the firewall is concerned: therefore, it would permit them to pass through to the Web server. The next step in firewall evolution serves to combat this type of attack. These types of firewalls are *application gateways*, or layer 7 firewalls.

This type of firewall not only filters network traffic based on the standard network parameters, but they also understand the higher layer protocol information contained within the packet, in this example HTTP. The firewall itself knows what a legitimate HTTP request looks like and can filter out a malformed or malicious request even though, from a network perspective, it might otherwise be a permitted packet. There is a downside to this type of approach, which is that the firewall must be programmed with all the same intelligence needed to filter normal traffic, plus the firewall must fully understand the protocols it is inspecting. This means additional programming for any protocol you want the firewall to understand. Most of the major commercial application gateways offer support for the major protocols such as HTTP, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

With all of this information circulating in your head, you're probably wondering which type is available for free. Generally speaking, you can find many free varieties of firewalls that perform some type of stateful inspection. Application layer gateways are not readily available for free. In reality, few organizations have the funds to use application gateways extensively. One ramification of *not* using an application gateway is that you need to ensure that the service that is exposed to un-trusted traffic is configured as securely as possible and that the server itself is hardened against attack. Keeping the service patches up-to-date will help reduce the odds that an application-level attack will be successful.

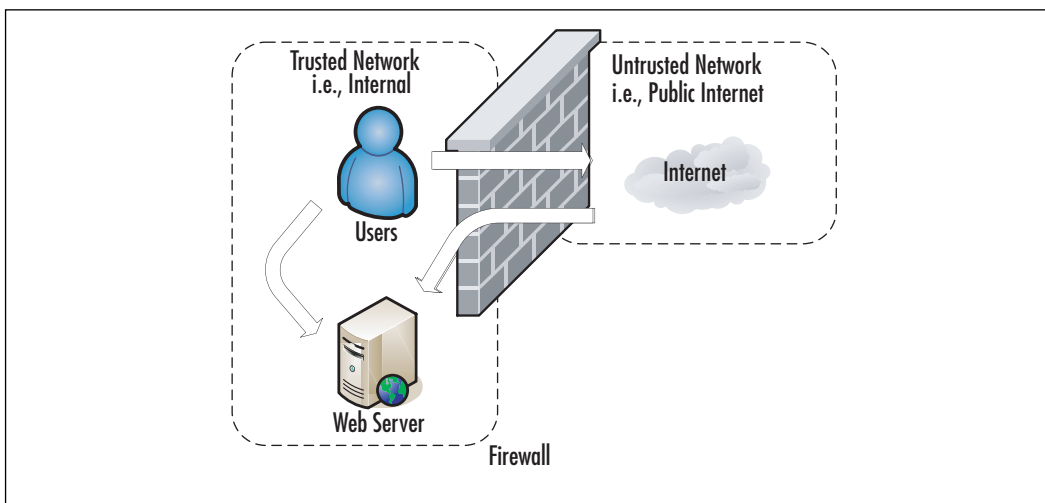
Firewall Architectures

The most securely configured firewall in existence will not provide much protection if a network was not designed properly. For example, if the firewall was installed into an environment that allows an alternate network path that bypasses the firewall, the firewall would only be providing a false sense of security. This is an architectural error that would render the firewall useless. In short, where the firewall is implemented is every bit as important as how it is implemented. The first step to installing anything is always planning. What follows is a discussion of the most common firewall architectures, in increasing order of security. Remember, these sections are discussing firewall architectures independent of the firewall type. For example, you could use a packet-filtering firewall, a stateful inspection firewall, or an application gateway in any of the designs discussed in the next section.

Screened Subnet

A *screened subnet* is the simplest and most common firewall implementation. Most small businesses and homes use this type of firewall (see Figure 2.1). This design places the firewall on the edge of your network, dividing everything (from the firewall's point of view) into internal and external, with nothing in between.

Figure 2.1 Screened Subnet Firewall

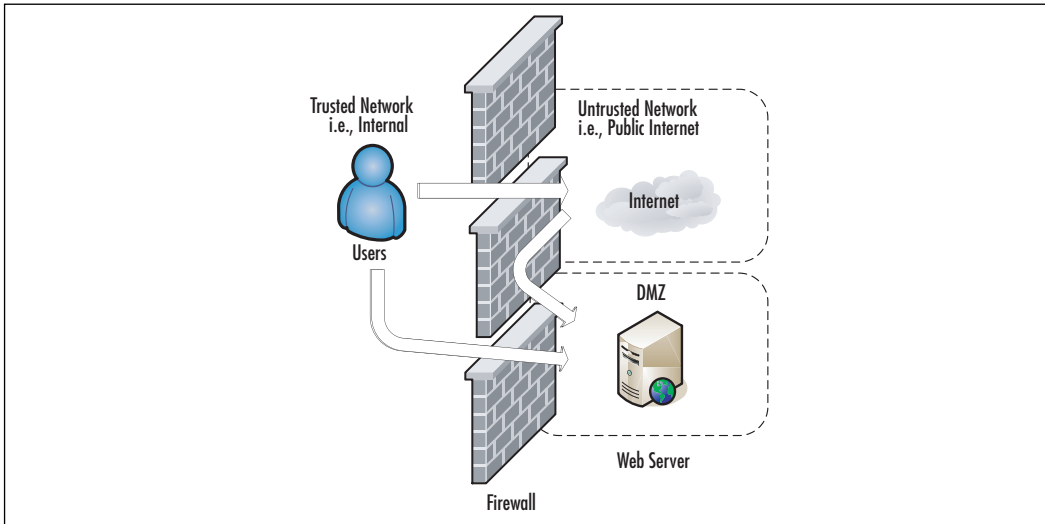


The screened subnet firewall (or *edge firewall*) is as straightforward as you can get. Internet users who need access to an internal server (e.g., Web, FTP, SMTP, and so on) must traverse the firewall to do so. Internal users needing access to those same servers would be able to access them directly. Internet traffic not destined for any Web-based server would be blocked at the firewall to prevent attacks on internal systems. All internal users must also traverse firewalls to access the Internet. This is the same type of firewall architecture you would have at home with a small network behind a Linksys router. This configuration has several advantages. The primary advantage is simplicity. With only two interfaces, the Access Control Lists (ACLs), which are the filters that define the criteria for permitting or denying traffic, are much simpler.

Although this configuration is cost-effective and simple to implement, it is not without its drawbacks. In this arrangement, the hacker has several chances to penetrate your network. If he or she can find a security hole in the firewall, or if the firewall is improperly configured, he or she might be able to gain access to the internal network. Even if the firewall is executed flawlessly, the hacker has a second opportunity to gain access. If the hacker can compromise any available Web-based services and take control of the servers, he or she would then have an internal system from which to launch additional attacks. Finally, if the servers are critical to the business function, by allowing the internal users to access them without going through the firewall, you may lose some audit capability that the firewall might otherwise offer. By far the biggest security weakness in this configuration is that if you are exposing any Web-based services: the servers hosting those services will be attacked frequently, and a compromise of one of those servers may expose your entire network.

One-Legged

The one-legged demilitarized zone (DMZ) still has the advantage of cost, because you are building a DMZ using only a single firewall (see Figure 2.2). Commonly, the firewall interfaces are called Internal or Inside, External or Outside, and DMZ.

Figure 2.2 One-Legged DMZ

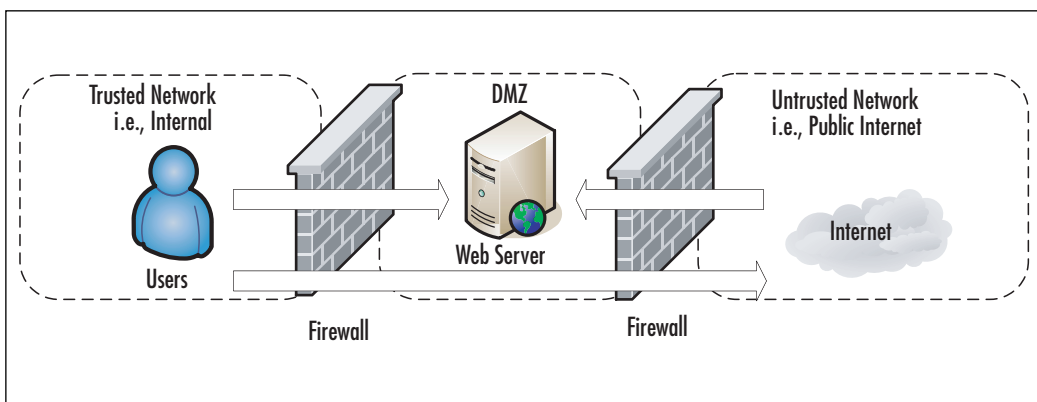
With this type of configuration you get to keep the low cost benefit, but add some isolation to your Internet-based servers. Internal users must traverse the firewall to access the servers or the Internet. External users must traverse the firewall to access the Web-based services. The real strength of this type of configuration is that if the servers that are hosting the Web-based services are compromised, the hacker still needs to contend with the firewall to continue attacking the internal network. As an added feature, because all users (internal or external) must traverse the firewall to access the Web-based servers, you may gain a higher degree of auditing from the firewall logs. If you wanted to provide even further isolation, assuming you have the available interfaces on the firewall, you could implement a separate DMZ for each Web-based server you needed.

The only real disadvantages to this configuration are complexity, and to a small degree, cost. As you add interfaces to the firewall, the configuration will become more complex. Not only does this complexity add to the time and labor for configuration and maintenance, it also increases the chance that an error could be made in the configuration. As you add interfaces there will often be additional costs associated with them. In most cases this cost will be minor and far less than an additional firewall, but with some high-speed interfaces, they can become very costly. Lastly, though many would consider it minor, with this configuration, if the firewall itself is defeated, the entire network is open to attack. Of course the solution to such paranoid thinking is costly.

True DMZ

The true DMZ is generally considered the most secure of firewall architectures. With this design, there is an external and internal firewall. Between the two is sandwiched any Internet accessible devices (see Figure 2.3).

Figure 2.3 True DMZ



Internet traffic is only permitted to a server in the DMZ, and only on the port that server is listening on. For example, if you had a Web server in the DMZ and an FTP server in the DMZ, traffic with a destination port of 80 would only be permitted to the Web server. For users accessing the same servers, the same rules would apply. Internal users would have to have permission through both firewalls to access the Internet. Obviously, this type of design costs more, typically double, but that cost buys you increased security. In a true DMZ, if the Web server is compromised the hacker is still trapped between two firewalls. For those who want to go the extra mile, the inside and outside firewalls can be of different types (e.g., Cisco Private Internet Exchange [PIX] and Linux netfilter). In this way, a hacker that finds a security hole in one firewall is unlikely to be able to apply the same techniques to the other firewall.

With all of the basics out of the way, you will be in a better position to make informed decisions when it comes time to propose and implement a firewall solution for your network. Bear in mind, while this chapter covers the basics of firewalls, there are entire volumes (such as *Designing and Building Enterprise DMZs* by Syngress Publishing, 2006) that explore the topic of firewall architectures, DMZ design, and implementation.

Tools & Traps...

Accidents Happen

I saw a corporate firewall/DMZ with a connection that allowed traffic to completely bypass their Internet firewall. I do not know why this happened, because the organization was not lacking properly trained networking professionals. These types of errors could occur because someone didn't analyze the implications of the changes adequately. Perhaps it was a "rush" to install some connectivity, or an emergency repair, or even a "temporary" fix. All of these things would indicate poor change control procedures. It is also possible that someone didn't realize the complete layout of the network when they made the connection in question, which could indicate inadequate network documentation among other things. In any case, these were trained professionals who should have known better, but accidents happen to the best of us.

Implementing Firewalls

When it comes to selecting a firewall there are a host of factors to consider. For commercial offerings there is the up front cost in addition to ongoing maintenance costs, which in some cases can be considerable. For free offerings, however, one of the first considerations is what OS you want to run the firewall on. This will impact how it is managed, and while the capabilities of the firewalls are likely similar, the implementation details will be very different. Most firewalls (commercial and free) run on either Windows or Linux. Some commercial offerings run on their own base system (e.g., Cisco PIX). With some firewalls the underlying Linux system has been so heavily modified it is now considered proprietary. In the case of a Linux firewall, you also have the option of installing the firewall software on a CD-ROM or pen drive. These steps are discussed in more detail in the following sections, along with specific configuration examples for setting up a free firewall on both Linux and Windows.

Hardware versus Software Firewalls

Another consideration is whether the firewall decision-making logic is run as software that sits on top of another functional system, or if the firewall is a dedicated piece of hardware. In the case of a Cisco PIX firewall, the smallest models are the size of a small cigar box and there is no OS other than the PIX software. This is a dedicated hardware device used to perform the firewall function, also called a *firewall appliance*. The other alternative is that the firewall is not a dedicated box, but a software component. Many popular firewalls take this approach as well, such as a *checkpoint firewall* that can be installed on top of a Windows system. Of these two approaches, if you want a free solution the choice is made for you. I know of no free hardware-based firewalls, so you will be using a software firewall.

Configuring netfilter

When it comes to Linux-based firewalls, there is only one choice, which is netfilter. This is partially because it was the best option available for the longest time. Since version 2.4, however, netfilter has been built into the Linux kernel. Even many commercial firewalls are running a modified Linux OS with netfilter inside their own custom cases. Netfilter is the underlying software that makes up the built-in firewall on Linux systems. The netfilter component reads the contents of the network packets and decides to permit or deny network traffic. Many times people incorrectly refer to the firewall as iptables, or prior to that, ipchains. In fact, iptables is the software command that is used to configure the rules that netfilter uses to make decisions to permit or deny traffic, and ipchains is the previous version of iptables. Even after you have settled on using Linux as your base OS for your firewall, there are some additional choices to make before you start any configuring.

Choosing a Linux Version

While all versions of Linux share some common characteristics, there will be differences. Depending on the specific Linux distribution, the differences could be significant and each distribution will likely offer some different sets of software packages. An excellent source of information on the different distributions is www.distrowatch.com. This site includes a brief summary of

what the distribution is trying to accomplish, and includes links to the home page and download locations. Because there are so many free versions of Linux available, it doesn't cost anything but the time to download and install several different versions and see which one you like. In the following examples I use a base system of Fedora core 5, which is the free version of the Red Hat Enterprise Linux that many companies use. I chose this distribution because it is one of the oldest and most well-established Linux distributions, and therefore extensive support documentation is available if you need it. If you just want to see if Linux is something you want to work with, try a live CD such as SLAX. When it comes to choosing the specific version of Linux you want to use, this decision must be made in parallel with choosing an installation media, because not all versions are supported on all media.

Choosing Installation Media

One of the more interesting features that Linux has over Windows is that it can be run from a variety of media. While windows is notoriously difficult to configure to run from a CD-ROM, there are Linux distributions that are capable of running off of a traditional hard disk install, CD-ROM, a Universal Serial Bus (USB) drive, or even a floppy disk. Each media type offers some security pros and cons, and not every distribution will be available on every media type. If you need the features of a specific distribution that doesn't come on the media you prefer, you may need to make a compromise. You will need to research the different media options and choose one that fits in your environment. We will review some of the pros and cons of each.

Full Install

The *full install* is the traditional install to a system's hard disk. Much like Windows, you boot up an install CD and walk through a guided install process. Most of the Linux distributions installed on the hard disk offer graphical user interface (GUI) install programs that walk you through the installation steps. There is no great advantage to using this type of distribution other than that the size of the hard disk allows you to install a lot of extra software. For a firewall, you generally want to keep the software running to a minimum to enhance security, so this shouldn't be a very big consideration. This type of installation also has the advantage that it will be easy to modify and alter the configuration if needed.

On the down side, this type of installation has all of the same disadvantages of a Windows bastion host. Namely that the entire system is sitting on the hard drive and if a hacker manages to compromise the root account, they will be able to install a virus or Trojan on the system that can survive future reboots. This type of install isn't any better or worse than if you were using Windows for your bastion host OS. Despite these concerns, this is the most common type of Linux firewall installation and most versions of Linux install the firewall components by default. This means if you download a version of Linux you like and install it to a hard disk, you will have a firewall waiting to be configured when you're done.

**TIP**

In the event that you discover your firewall has been compromised, it is considered best practice to wipe the system clean and rebuild it from scratch. Unfortunately, unless you have some means of isolating *all* changes that were made, you cannot ensure that it is safe to leave the system operational. One of a hacker's first steps is often to install a back door so that they can easily gain access to the device in the future. These backdoors include techniques such as modifying various systems commands so that detecting the back door is difficult. For this reason, rather than risk leaving a system operational that may be compromised, a complete format and reinstall is recommended.

CD-ROM

While you can get windows running off of a bootable CD-ROM or live CD, it takes a lot more work than it does with Linux. There are many versions of Linux designed specifically to run from a CD-ROM, allowing you to turn virtually any machine into a firewall, router, or general-purpose PC. There is an obvious security advantage to having all of your configuration information on read-only media. Even if a hacker manages to compromise the system, all it takes is a reboot and it can be restored to its previous condition. The system can still fall victim to a virus or Trojan, but only until it is rebooted. Further, if the firewall system has a hardware failure such as a failed central processing

unit (CPU), all you would need to do to restore your firewall would be to move the CD to a new system and reboot.

The primary advantage to a CD-ROM-based installation is also the primary disadvantage. If you burn the entire OS and configuration settings to a CD, any time you need to make adjustments you would need to burn a new CD-ROM. The cost of the CD media probably isn't an issue, but such a configuration may hinder your ability to remotely administer the system, which would be limited to making changes to the running configuration. Changes that remained after a reboot would require someone local to insert the CD-ROM containing the new configuration. If you needed to implement and test changes that required a reboot to take effect, this type of the setup would make things more difficult. Finally, due to simple space limitations on a CD-ROM, you may not be able to fit all of the needed software or functionality on a CD-ROM. That being said, if the firewall rules are relatively static and don't require frequent adjustment, a live CD could be a very attractive option.

USB Drive

If the space limitations are acceptable, a Linux-based firewall booting from a USB disk may offer the best compromise in security and flexibility. Having the operating systems and firewall software on a pen drive offers the same type of flexibility that a CD-ROM-based system provides, with increased storage capacity over that of a CD-ROM. If you purchase a USB disk that includes a physical write protect switch, you can make changes on the fly, like a live system, and then write protect the disk against modification when you are done. As the storage capacity of USB drive increases, you will be able to use a USB-based distribution that includes increasingly greater functionality. One key consideration with this type of media is that not all systems will support booting from a USB disk. While almost all newer systems support this option, many of the older systems that you may wish to install a free firewall on do not.

Floppy Disk

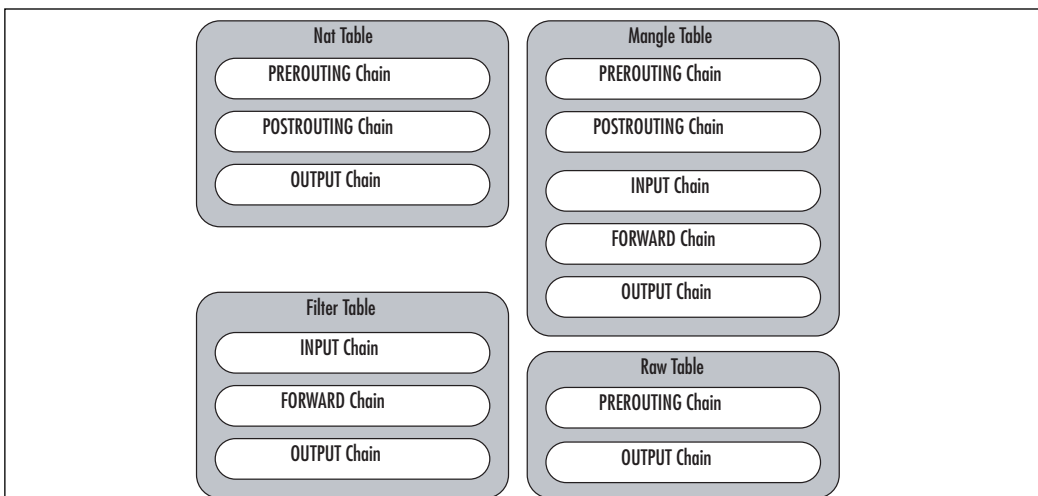
Although the functionality is typically very limited, there are many versions of Linux that can fit on a 3.5" floppy disk. The primary advantage of these distributions is their low resource requirements. Often, these systems only require 8

or 16 megabytes of memory and a 486 processor to function. The ability to toggle the write protect switch on the floppy can also provide a high degree of configuration flexibility and security. Considering the unreliable nature of floppy disks, it probably wouldn't be appropriate for use if an outage cannot be tolerated. At the very least you should have duplicate floppy disks available in the event of a failure. Another disadvantage to these is functionality. Generally, these floppy-based distributions are single-purpose devices and lack much in the way of functionality. Another consideration is that due to the space restrictions on a floppy disk, these floppy-based distributions are almost always command line only, with no GUI for configuration or management.

Linux Firewall Operation

Before discussing the specific commands used to configure the Linux firewall, we will cover some basic Linux firewall vocabulary and how the firewall operates. Netfilter contains the firewall logic, and iptables is the program that is used to modify the rules that the firewall uses. (See the netfilter home page at www.netfilter.org/.) These rules (or ACLs) define the rules used to permit or deny packets and how to react to denied packets. The current iptables use both tables and chains. *Tables* are the blocks of processing where various actions are performed on the packets. Different tables process different chains. *Chains* are a set of rules (or ACLs). There are four built-in tables: *nat*, *mangle*, *filter*, and *raw*, each of which processes different chains (see Figure 2.4).

Figure 2.4 Netfilter Tables and Chains



The following tables and chains are not listed in any particular order, as a given packet may be impacted by multiple tables and chains as it is processed. The primary built-in chains are INPUT, OUTPUT, and FORWARD. In addition to these, you can create your own user-defined chains. Capitalizing the names of the chains is a common convention, but is not required.

A brief summary of the roles the tables and chains play is included for reference.

- **Nat Table** This table is referenced with a packet that is used to create a new connection.
 - **PREROUTING** This chain is processed as soon as a packet is received and before any routing decisions are made.
 - **POSTROUTING** This chain is processed before a packet is sent to an interface but after any routing decisions have been made.
 - **OUTPUT** This chain is processed for packets generated locally.
- **Filter Table** This is the default table that is used when the *iptables* command is used to modify the rules, and the command does not specify an alternate table. This is where the bulk of a firewall's processing is consumed.
 - **INPUT** This chain is processed for packets destined for the local system.
 - **FORWARD** This chain is processed for packets passing through the local system.
 - **OUTPUT** This chain is processed for packets generated by the local system.
- **Mangle Table** This table is used for any specialized packet alterations that are needed. Examples are performing Network Address Translation (NAT) or manipulating various bits within the packet.
 - **PREROUTING** This chain is processed on incoming packets before a routing decision is made.
 - **POSTROUTING** This chain is processed last before a packet is sent to an interface.

- **OUTPUT** This chain is processed before a routing decision is made for packets generated locally.
- **INPUT** This chain is processed for packets destined for the local system.
- **FORWARD** This chain is processed for packets passing through the local system.
- **Raw Table** This table is primarily used for packets that are exempt from connection tracking, and if required, are called before any other netfilter table.
- **PREROUTING** This chain is processed as soon as a packet is received.
- **OUTPUT** This chain is processed for packets generated locally.

After you have reviewed all the various tables and chains, it's worth discussing the overall packet flow. The key to remember is that not all packets traverse all chains. To further muddy the waters, packets will traverse different chains depending on whether they are sourced from the netfilter host, destined for the netfilter host, or just passing through the netfilter host. Remembering this will save you time when troubleshooting your firewall rules in the future. Refer to Figure 2.5 for a diagram depicting the packet flow through netfilter.

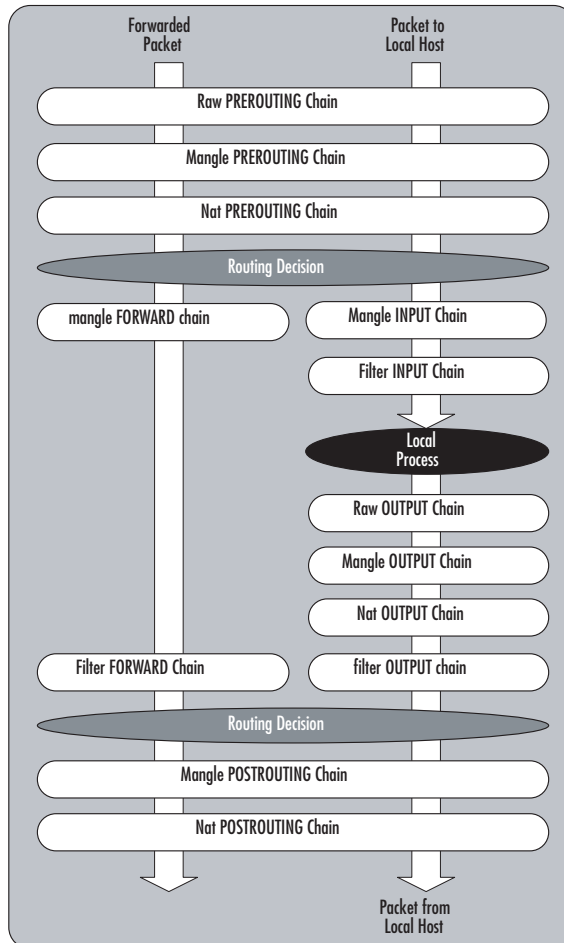
Targets are the actions that should be taken when a packet matches a given rule. A target is specified using the `-j <target>` syntax (for jump). The primary targets used for a firewall are **ACCEPT** and **DROP**.

- **ACCEPT** The packet is accepted and processed by the rest of the TCP/IP stack.
- **DROP** The packet is dropped, and no notice is given to the sender. While this does not honor the TCP/IP protocol specifications, it is considered the most secure option, because it denies a hacker useful information about the firewall. This behavior also has a negative side effect, which is if a system is trying to initiate a connection to a port that is blocked by a firewall, the connection attempt must time out before the initiating host gives up. If you use **REJECT**, the Internet

Control Message Protocol (ICMP) port will allow the initiating system to abort the connection attempt immediately.

- **LOG** This allows you to perform kernel logging, which appears in the syslog log. Further options allow you to specify the log level and a descriptive prefix for the log entry.
- **RETURN** Processing continues in the previous chain at the rule just after the last rule processed in that chain.
- **QUEUE** This is a special target that will hold (or queue) a packet for processing by a userspace process.

Figure 2.5 Netfilter Packet Flow



Unlike some firewalls, netfilter allows you to apply multiple rulesets (chains) to the same interface. Although it may seem minor, this option creates a lot of powerful possibilities. For example, suppose you have an ACL and you want to permit all packets originating on the 192.168.1.0 network except those from 192.168.1.11, which is a host that a third-party uses and is not a completely trusted system. You want packets sourced from 192.168.1.11 with a destination port of 22, 25, 53, 80, and 443 to be permitted, while all other packets are blocked (see Figure 2.6).

Figure 2.6 Cisco ACL

```
1 somerule
2 access-list 100 permit tcp host 192.168.1.11 any eq 22
3 access-list 100 permit tcp host 192.168.1.11 any eq 25
4 access-list 100 permit tcp host 192.168.1.11 any eq 53
5 access-list 100 permit tcp host 192.168.1.11 any eq 80
6 access-list 100 permit tcp host 192.168.1.11 any eq 443
7 access-list 100 deny ip host 192.168.1.11 any any
8 access-list 100 permit ip 192.168.1.0 255.255.255.0 any
9 somerule
```

In Figure 2.5, each line of the ACL is numbered for easy reference. The order of the rules is critical for proper operation of the firewall. Cisco processes each line in the ACL and compares the rule with the packet in question. If it finds a match, it performs the indicated action and then stops any further processing of the ACL. This means if you reversed the order of rules 7 and 8, all packets from 192.168.1.11 would be permitted. This type of arrangement also means that a packet with a source IP address of 192.168.1.22 has to be compared against rules 2–7 before being accepted by rule # 8. With seven rules this will happen quickly, but if the ACL is lengthy this extra overhead could be CPU-intensive.

Netfilter's ability to move through multiple chains for the same packet allows you to design your chains for greater efficiency (see Figure 2.7).

Figure 2.7 Netfilter Chains

```

FORWARD Chain
CUSTOM Chain

1 somerule
2 iptables -A FORWARD -p tcp -s 192.168.1.11 -j CUSTOM
   2.1 iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 22 -j ACCEPT
   2.2 iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 25 -j ACCEPT
   2.3 iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 53 -j ACCEPT
   2.4 iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 80 -j ACCEPT
   2.5 iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 443 -j ACCEPT
   2.6 iptables -A CUSTOM -p ip -s 192.168.1.11 -j DROP
   2.7 iptables -A CUSTOM -j RETURN
3 iptables -A FORWARD -p ip -s 192.168.1.0/24 -j ACCEPT
4 somerule

```

Using netfilter and iptables, you created rule # 2, which says that the source address is 192.168.1.11 for processing the CUSTOM chain. You can create the CUSTOM chain with the *iptables -N CUSTOM* command. Within the CUSTOM chain, you check for the five permitted destination ports (rules 2.1–2.5) and then reject everything else (rule 2.6). Rule # 2.7 has no matching criteria and will therefore match on any packet and instruct the packet to return to the FORWARD chain where processing can continue. FORWARD chain rule # 3 permits all other packets from the 192.168.1.0/24 network. This means that packets not sourced from 192.168.1.11 only have to be checked against rule # 2 and can then move through the chain(s) instead of being checked against all the rules. The actual rules as they would appear in iptables can be seen with the *iptables -L* command.



```

# iptables -L
Chain INPUT (policy DROP)
target    prot opt source                destination

Chain FORWARD (policy DROP)
target    prot opt source                destination
CUSTOM    tcp  --  192.168.1.11          anywhere
ACCEPT    tcp  --  192.168.1.0/24       anywhere

Chain OUTPUT (policy DROP)
target    prot opt source                destination

```

Chain CUSTOM (1 references)

target	prot	opt	source	destination	
ACCEPT	tcp	--	192.168.1.11	anywhere	tcp dpt:ssh
ACCEPT	tcp	--	192.168.1.11	anywhere	tcp dpt:smtp
ACCEPT	tcp	--	192.168.1.11	anywhere	tcp dpt:domain
ACCEPT	tcp	--	192.168.1.11	anywhere	tcp dpt:http
ACCEPT	tcp	--	192.168.1.11	anywhere	tcp dpt:https
DROP	all	--	192.168.1.11	anywhere	
RETURN	all	--	anywhere	anywhere	

Another advantage is that because rule # 2 sent you to another chain, you can make certain assumptions that you wouldn't otherwise be able to. For example, in the CUSTOM chain you could replace

```
iptables -A CUSTOM -p tcp -s 192.168.1.11 --dport 22 -j ACCEPT
```

with

```
iptables -A CUSTOM --dport 22 -j ACCEPT.
```

This is because the packet would not be in the CUSTOM chain without matching the `-p tcp` and `-s 192.168.1.11` (source IP address). If you want to tweak the CUSTOM chain even more, the RETURN target in rule # 2.7 isn't strictly required. If the packet reaches the end of a user-defined chain without having a match, it will RETURN to the previous chain by default. If a packet reaches the end of a built-in chain without a match, it will use the policy target (typically DROP). Now that you have a feel for the flexibility and power of iptables and netfilter, let's look at some practical configuration examples.

Configuration Examples

The next step is to demonstrate how to configure the netfilter firewall. This is a critical step, and the firewall should only be installed and configured after the underlying OS has been installed, updated, and hardened. These instructions assume you are working with an otherwise secure system and now need to configure the firewall functionality.

To make sure the firewall is enabled, you can run `chkconfig --list`, which lists all of the services and the run levels they are configured to start in. For example, you get the following output:

```
chkconfig --list | grep iptables
iptables    0:off    1:off    2:on     3:on     4:on     5:on     6:off
```

This output tells you that iptables will start in run levels 2–5. You can set it to run in run levels 2–5 by using the **`chkconfig --level 2345 iptables on`** command. If you are using a GUI window manager, you probably have another graphical application to see this information. For example, in Fedora Core 5, you can navigate to **System | Administration | Security Level and Firewall**, which opens the screen shown in Figure 2.8.

Figure 2.8 Fedora Core Firewall GUI



You can enable or disable the firewall by going to the **Firewall Options** tab and selecting **Enabled** or **Disabled**. This particular interface in Fedora Core 5 also allows you to perform limited configurations of the firewall rules (e.g., by checking the Trusted Service SSH, a rule would be added to allow inbound connections on TCP port 22). Because any graphical interface provided will likely vary from one distribution to another, we use the command line to configure the firewall.

Deleting Rules and Chains

With many Linux distributions, the netfilter firewall will come enabled, but with an empty ruleset. In others, it might come with the firewall enabled and a very liberal ruleset in place. Let's start configuring a Linux firewall by deleting any default rules that are present. You can use *iptables -L* (or *--list*) to list the current rules. An empty default ruleset should look like this:

```
SYNGRESS
syngress.com
iptables -L
Chain INPUT (policy ACCEPT)
Target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
Target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
Target     prot opt source                destination
```

If there are any default rules present, they can be deleted using the *iptables -F* command. The *-F* option means to flush, which is equivalent to using *--flush*. This will clear all rules out of any existing chains. If distribution has any additional chains created beyond the default, you can delete a custom chain by using the *iptables -X customchain* command. Creating your own user-defined chain is accomplished using the *iptables -N customchain* command. In addition to the individual rules within a chain, the built-in chains have a default policy associated with them. This policy tells netfilter what to do if a packet reaches the end of the chain without finding a match. While the default policy is to ACCEPT, it is better to change this to DROP by using the *-P* option, which sets the default policy for that chain, as follows:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Permitting Traffic to and from the Firewall

Now that you have a clean slate and a default policy of DROP, the first thing you will want to do is make sure that management traffic is permitted to the firewall itself. This is done first, because once you have enabled the firewall

with a default policy of DROP, you will not be able to manage the firewall remotely until you have configured the firewall rules to permit the management traffic. This traffic is processed against the INPUT chain, because the destination is the netfilter host itself. To allow secure shell (SSH) connections to the firewall, use the following command:

```
iptables -A INPUT -p tcp -s 192.168.99.0/24 --dport 22 -j ACCEPT
```

In this example, you are appending (*-A*) a rule to the INPUT chain to allow traffic from the 192.168.99.0/24 network to a destination port of TCP 22. With no other configurations, all other traffic through or to the firewall would be dropped. This will show up in the rule listing as follows:



```
iptables -L INPUT
Chain INPUT (policy DROP)
Target      prot opt source                destination
ACCEPT      tcp  --  192.168.99.0/24        anywhere             tcp dpt:ssh
```

Although the aforementioned rules will permit the inbound SSH session, there is currently no rule to permit the reply traffic for the SSH session. If you were to change the default policy for the OUTPUT chain to ACCEPT, this would permit the reply packet, but we will instead address this more securely in the next few examples.

If you also wanted to allow 192.168.99.99 access to the firewall with a destination of TCP port 80, you could use the same syntax with *-A* to append the rule, which would put the new rule for port 80 *after* the rule for port 22. You could also use *-I* for *insert*, as in the `iptables -I INPUT 1 -p tcp -s 192.168.99.99 --dport 80 -j ACCEPT` command. This would insert the new rule in the INPUT chain as rule # 1, meaning the rule for port 80 would come *before* the rule for port 22. Remember, this is still permitting only half of the conversation; you still need to permit the outbound reply packets. It is sometimes useful to list the chains with rule numbers using the `iptables -L --line-numbers` command.

For outbound traffic (i.e., traffic generated by the firewall), you need to create rules in the OUTPUT chain. To enable syslog traffic from the firewall to a remote syslog server (192.168.1.99), you would enter the following:

```
iptables -A OUTPUT -p udp -d 192.168.1.99 --dport 514
```

This assumes you are using the default UDP syslog port of 514. Because syslog over UDP is a one-way conversation, you will not need to permit any inbound replies to the syslog traffic. The OUTPUT chain is where you need to permit replies for permitted traffic that you allowed inbound in the preceding examples. You could create rules to permit SSH and HTTP specifically, but there is also a way to permit all traffic that is a reply to a permitted session. You can enter

```
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

This will instruct netfilter to permit any outbound traffic that is part of an established session (ESTABLISHED). The RELATED keyword is similar, but it is for traffic that is part of a different session, but where the session is related to an established session. Some protocols will open additional ports (such as FTP) as part of their normal behavior. For those that netfilter understands, it can see the request for the additional port and permit that new session.

**TIP**

iptables commands that manipulate the chains or rules themselves use uppercase letters:

-A append, -D delete rule, -I insert, -R replace, -L list, -F flush, -N new, -X delete chain

Lowercase options are used for specifying rule parameters:

-s source address, -p protocol, -d destination address, -j jump, -i in-interface, -o out-interface

Simulating the Windows Firewall

Now let's configure the firewall. The built-in firewall on Windows XP is enabled by default with service pack 2 or better. The standard configuration is to allow outbound connections from the host system, and deny inbound connections unless they are explicitly configured. The Windows firewall also allows any traffic that is a reply to traffic that the host originally generated outbound. After you execute the **iptables -F** command to flush out all of the previously configured rules, the following commands would configure the Linux host similarly:

```
iptables -P OUTPUT ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

The `--state` extensions track the current status of the connections. By specifying `ESTABLISHED` or `RELATED`, the firewall allows packets that are part of a currently established session, or packets that are starting a new session, but where the session is related to an existing session (such as an FTP data session). If you were hosting a service on this system, such as a Web server, you would need to configure the `INPUT` chain appropriately. This configuration would afford any Linux system a minimum level of firewall security with virtually no impact to its overall functionality.

Simulating a Home Network Router

With the basics of `iptables` configuration out of the way, let's tackle a more practical example. For a typical firewall, there is very little traffic destined *to* or *from* the firewall itself. In general, the only traffic that would fit this profile would be administrative sessions to configure the firewall itself. The vast majority of a firewall's traffic is passing through the firewall, and will thus be checked against the `FORWARD` chain. The following examples would configure the Linux firewall with the same access controls as a typical home network router such as a Linksys or Netgear router/firewall. This example assumes that `192.168.1.0/24` is the internal network on interface `eth0` and the external interface is `eth1`.



```
iptables -P OUTPUT ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -A INPUT -p tcp -s 192.168.1.0/24 -i eth0 --dport 80 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -o eth1 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

NOTE

Always remember that if you have configured the default policy for a chain to DROP (for example, `iptables -P FORWARD DROP`) that you will need to include an explicit rule to permit the return traffic. This can be done by using the following command:

```
iptables -A <CHAIN> -m state --state ESTABLISHED,RELATED -j ACCEPT
```

So if you wanted to permit the return traffic for a FORWARD chain, you would enter

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

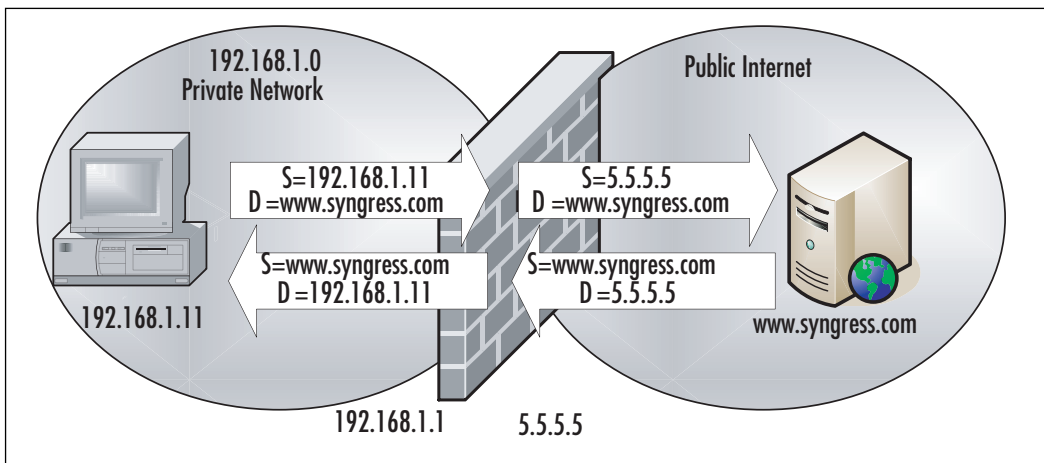
Many hours of troubleshooting Linux firewalls have been spent by overlooking a rule that permits the return traffic.

The INPUT chain allows port 80 to go to the firewall itself from the internal network. Many of the home routers have a Web interface for configuring them, and while your configuration may not need this port open to the firewall, it is included here to help emphasize how the different chains are used. It is important to specify the input interface (using `-i`) so that the source IP cannot be spoofed by an external attacker. In this way, you ensure that even if a packet was generated with the proper source IP, if it came in on the outside interface (*eth1*) it would not match the rule and would thus not be permitted. The FORWARD rule allows any outbound traffic from the internal network to the external network. This configuration is simple to implement; however, the 192.168.1.0 IP range is a private IP range and is not routable on the Internet. Thus, this range wouldn't allow traffic from the internal network to the Internet quite yet. To make this Linux firewall a useful replacement for a home network router, you need to enable NAT, which allows all of the systems on your internal network to appear as a single IP address when communicating on the Internet.

Let's review NAT in its various incarnations. In principle NAT is simple, but in a complex environment, it can get confusing. As always, good documentation can help keep things straight. Basically, NAT means that the NAT device (in this case the Linux netfilter firewall) will change the IP address in a packet and retransmit that packet. Depending on your needs, you can alter the

source IP address (source NAT [SNAT]), the destination IP address (destination NAT [DNAT]), or both (double NAT). For example, take a home router. The objective behind the NAT capability is to allow all of the internal hosts to communicate on the Internet using the single public IP provided by your Internet Service Provider (ISP). (In this case, SNAT is being used.) As each of the hosts on your private network make a connection to an Internet server, the firewall is altering the source address to look like the public IP from your ISP. By doing this, the return traffic can find its way back to the firewall and be retranslated and sent to the originating host (see Figure 2.9).

Figure 2.9 SNAT



In Figure 2.9, the internal host has a private IP address of 192.168.1.11. The public address of the firewall is 5.5.5.5, which is provided by the ISP. If a host on the private network wants to make a connection to www.syngress.com using a Web browser, the connection is sent with source address 192.168.1.1 to a destination address of www.syngress.com. The firewall alters the source address to its own public IP address of 5.5.5.5 and sends the packet on its way. When the server replies to destination 5.5.5.5, the firewall again edits the packet, this time inserting a new destination of 192.168.1.11. All of this takes place and is transparent to the 192.168.1.11 host and the www.syngress.com server. When multiple hosts are using SNAT, the firewall tracks which connections belong to which private hosts using the port numbers. While the destination port of the Web server remains static

(typically port 80 for the Web), the source port is usually a random port above 1024. By tracking the source port, the firewall knows which address belongs to which session. In the event that two hosts attempt to use the same source port, the NAT device edits the source port of one of the connections and replaces it with another random source port. When the return traffic is received, it translates the source port back, just like it did for the IP address. Because this method of NAT relies heavily on using the source port number, it is sometimes referred to as port NAT (PNAT).

To add the SNAT functionality to the example firewall, use the following command:

```
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to-source 5.5.5.5
```

The `-t` option is used to specify the table you want to modify, and the `-A` option specifies that you are going to append this rule to the `POSTROUTING` chain. By specifying the outbound interface, you are ensuring that the SNAT only occurs as traffic leaves the private network, meaning only in the proper direction.

NOTE

SNAT can only be performed in the `nat` table. However, the rules for SNAT can only go in the `POSTROUTING` chain of the `nat` table. This means that any time you use SNAT, your rule will contain `-t nat -A POSTROUTING`.

The jump target SNAT is self explanatory. The `--to-source` option specifies what IP address we want to use as the new source address. SNAT assumes we have a static IP address to SNAT the outgoing packets to. While this is likely the case in a corporate environment, a more appropriate solution to more closely mimic the configuration of a home router would be to use the `MASQUERADE` command:

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

The masquerade command does not require an IP specification, and will use the IP address of the firewall interface. You might be wondering why you

wouldn't use the masquerade target all of the time instead of the SNAT target. Because the source IP is static, the SNAT target will cause the NAT calculations to be performed once for a given session. Subsequent packets belonging to that session are handled the same way as the first. With the masquerade target, each packet is checked for the source IP to use, which requires more overhead than with SNAT. This is why SNAT is preferable if you have a static source IP address, and masquerade is your only option if you do not have a static source IP address to use.

Additional Commands

By this point, you should have a relatively solid grasp of how to configure a Linux firewall. So far we have covered all of the core commands to permit and deny the traffic. Another useful command for your Linux firewall deals with logging packets. If you want to log everything passing through the firewall, use the **iptables -A FORWARD -j LOG** command. While simple, this would likely generate an excessive amount of logging traffic. You also might want some additional control of how the logging occurs. There are some additional options to provide this functionality. Of particular note are the *--log-level* and *--log-prefix* options.

The *--log-level* option allows you to specify what logging level is used for the LOG rule. The effect this log level has depends on how you have your kernel logging configured (via *syslog* or *syslog-ng*). When you combine the custom logging level of *iptables* with the *syslog* configuration, you can have *syslog* act in any manner of ways based on the firewall logs, including sending e-mails for certain events. The *--log-prefix* option allows you to insert up to a 29-letter string in front of the log entry. This can be useful for troubleshooting purposes. Some examples of information you could place in log prefix would be the name of the chain that generated the log entry such as *iptables -A FORWARD -j LOG --log-prefix "from FORWARD chain."* (For more information on event logging, refer to Chapter 7, "Network Reporting and Troubleshooting.")

NOTE

While a packet that matches an ACCEPT or DROP rule will stop traversing any other chains, this is not true of packets that match a log rule. After matching the log rule, the packets continue through any appropriate chains to be processed. Keep this in mind, so that you can configure an additional rule and action for the packet if desired.

Now that you can create a working ruleset for netfilter, you will want to save it. There are two commands of note: one for saving the configurations and one for loading a saved configuration. You can use the *iptables-save* command to generate output that is the current active ruleset. By default, it will generate the output only to the stdout, meaning it will display in the console. To save this output, redirect it to a file. To redirect the current ruleset to a file called */etc/ruleset*, you would type **iptables-save > /etc/ruleset**. If you want to save the current packet counts and rule counts, use the *iptables-save -c > /etc/ruleset* command. Individual tables can be saved separately by specifying the *-t* option using the *iptables-save -t mangle > /etc/ruleset* command.

Restoring a ruleset is accomplished using the *iptables-restore* command. Like *iptables-save*, the restore function takes only two optional arguments. The *-c* option will cause iptables to load the saved packet and byte counts, overwriting the current count values. The default behavior when using *iptables-restore* is to flush the ruleset before loading the saved ruleset, thus all previous rules are lost. If you wish to override this behavior, you can use the **-n** option, in which case the rules will be added to the existing ruleset, and will only overwrite if there is a duplicate rule. You can use the *iptables-restore < /etc/ruleset* command to pipe the saved configuration to iptables-restore.

Command Summary

The following is a brief summary of the most useful iptables commands for easy reference, along with some examples to make the command usage more clear. Bear in mind this is not an exhaustive list of commands; it only represents the most important commands for configuring your firewall. For a complete list, refer to the iptables man page.

- `-A` appends a rule to a chain. `iptables -A INPUT -p icmp -j ACCEPT` will add the rule to permit ICMP at the bottom of the `INPUT` chain in the `FILTER` table.
- `-D` deletes a rule from a chain. `iptables -D INPUT -p icmp -j ACCEPT` will delete the matching rule from the `INPUT` chain. `iptables -D INPUT 3` will delete the third rule from the top in the `INPUT` chain.
- `-I` inserts a rule in a chain. `iptables -I INPUT 5 -p icmp -j ACCEPT` will insert this rule as the fifth rule in the `INPUT` chain
- `-R` replaces a rule in a chain. `iptables -R INPUT 4 -p icmp -j ACCEPT` will replace the fourth rule in the `INPUT` chain with this new rule.
- `-L` lists the rules. `iptables -L` will list all rules and `iptables -L INPUT` will list all rules in the `INPUT` chain only.
`iptables -t nat -L` will list all the rules in the `nat` table only.
- `-F` will flush (delete) the rules. `iptables -F` will delete all rules in all chains. It will not delete chains, only the rules inside the chains.
- `-Z` will zero the packet and byte counters. `iptables -Z` will delete all of the counters. `iptables -Z FORWARD` will delete all of the counters in the `FORWARD` chain only.
- `-N` will create a new chain. `iptables -N CUSTOMCHAIN1` will create a new chain named `CUSTOMCHAIN1`.
- `-X` will delete a chain. `iptables -X CUSTOMCHAIN1` will delete the custom chain named `CUSTOMCHAIN1`.
- `-P` will change the policy for a chain. `iptables -P INPUT ACCEPT` will change the policy for the `INPUT` chain to `ACCEPT`.

The policy for a chain does not need to be limited to `ACCEPT` or `DROP`; it could use a custom chain for a target, if desired.

Option Summary

- `-p` specifies the protocol to match (works with “!”). `iptables -A FORWARD -p tcp` will add a rule to match any TCP packet to the FORWARD chain. `iptables -A FORWARD -p ! tcp` will match any packet that was not TCP.
- `-s` specifies the source address to match (works with !). `iptables -A FORWARD -s 192.168.1.99` will match any packet with a source address of 192.168.1.99. `iptables -A FORWARD -s ! 192.168.1.99` will match any packet that did not have a source address of 192.168.1.99.
- `-d` specifies the destination address to match (works with !). `iptables -A FORWARD -d 192.168.1.99` will match any packet with a destination address of 192.168.1.99.
- `-i` specifies the network interface that the traffic was received on (works with !). `iptables -A FORWARD -i eth0` will match any packet entering the eth0 interface.
- `-j` specifies the target. `iptables -A FORWARD -p tcp -j DROP` would create a rule at the bottom of the FORWARD chain that will DROP any TCP packet.
- `-o` specifies the network interface that the traffic was sent out of (works with “!”). `iptables -A FORWARD -o eth1` would match any packet leaving on the eth1 interface.
- `-t` specifies the table to manipulate. `iptables -t nat -A POSTROUTING -p tcp -j DROP` will add a rule to the bottom of the POSTROUTING chain in the NAT table, to DROP any TCP packet.

If you don't specify the `-t` option, iptables assumes you are working with the filter table.

- `-v` specifies to be verbose. `iptables -L -v` lists all of the rules and includes packet counts per chain and per rule.
- `--line-numbers` specifies that the rule list should be numbered:

```
iptables -L --line-numbers
```

This option makes it easier to know what number to use for the commands that take a rule number as an argument, such as *insert*, *delete*, *replace*, and so on.

- *-m* will match packets based on certain protocol-specific criteria. Because the match options are protocol specific, *-p* (*tcp/udp/icmp*) must be used with *-m*. Some common examples include:
 - *-m --sport* allows you to match packets based on the TCP or User Datagram Protocol (UDP) source port.
 - *-m --dport* allows you to match packets based on the TCP or UDP destination port.
 - *-m multiport* allows you to match packets based on multiple port numbers within the same rule. *iptables -A FORWARD -p tcp -m multiport --dport 22,25,53 -j DROP* would *DROP* any TCP packet with a destination port of 22, 25, or 53.
 - *-m state --state* will allow you to match packets based on the state of the connection. *iptables -A FORWARD -p tcp -m state --state NEW -j LOG* would *LOG* any TCP packets that were being used to initiate a new connection.

There are four recognized states: *NEW*, *ESTABLISHED*, *RELATED*, and *INVALID*.

Netfilter and iptables give you powerful packet-filtering and manipulation capabilities for free. With Linux distributions available for free download, a firewall is within any company's reach. Because of this, deploying firewalls internally to protect highly sensitive systems or data is becoming increasingly viable. If you want to obtain a Linux firewall without having to install Linux, try any of the many live CDs that are available. Some excellent choices are be Knoppix or Slax.

GUIs

While the console commands that are used to manipulate and configure netfilter are not terribly complicated, they can sometimes get very lengthy. As the

length of the command line grows, the chances of an accidental error increase. Alternatively, you may not like working on the command line, in which case there are a wide variety of GUI and menu-driven interfaces available for netfilter. In most cases, these menu-driven interfaces use your input to create the appropriate iptables commands, and alleviate you having to know the various switches and options to use. There are a large number of GUIs available to configure your netfilter firewall. These GUIs are listed in the following section in approximate order of ease of use. All else being equal, we have demonstrated the GUIs that are available on a wide variety of platforms over an equal quality choice that only works with one distribution. In general, simpler also means less full featured, so be aware that if you are trying to create a complex ruleset, some GUIs may not have the needed functionality.

Security Level Configuration

You can start the iptables GUI provided with Red Hat-based Linux distributions by navigating to **System | Administration | Security Level and Firewall**. You can also call the program directly by running `system-config-securitylevel` from a terminal window. While the interface looks nice, it is limited in what it can configure. Basically, all you can do with this GUI is permit or deny certain ports. Fedora Core 5 configures the *INPUT* and *FORWARD* chains to jump to a custom chain named *RH-Firewall-1-INPUT*. There is no ability to differentiate between ports permitted in the *INPUT* chain or the *FORWARD* chain, because all rules configured through the GUI are applied to this custom chain.

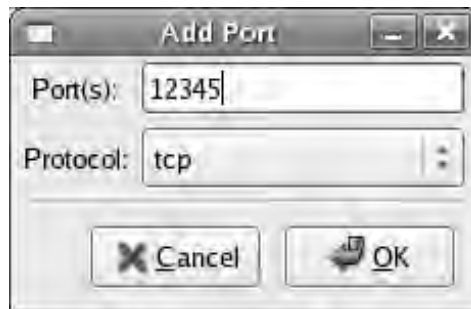
Some services are predefined for you. Placing a check next to **SSH** and clicking **OK** and then **Yes** to commit the changes will create the following rule in the *RH-Firewall-1-INPUT* chain:

```
iptables -A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

By expanding **Other ports** on the **Firewall Options** tab, you can enter a custom port number (see Figure 2.10.)

Figure 2.10 Custom Ports

Click **Add**, and enter the desired port number in the dialog box. Use the drop-down menu to select **TCP** or **UDP** for the protocol and click **OK** (see Figure 2.11).

Figure 2.11 Custom Port Dialog

This creates a rule identical to the SSH rule. There are no other configuration options. While this interface is adequate for a home PC that isn't running any services, it probably will not be adequate for a corporate firewall. If you need to configure access based on the interface in use or need to configure any NAT rules, you will need to use a different GUI. While you probably won't need this particular GUI as a corporate firewall, it is still useful to be familiar with it if you are running any Linux systems as workstations.

Lokkit

Lokkit is an ncurses-based menu for configuring your netfilter firewall. Lokkit is available for most major distributions and can be installed by default on some (such as Fedora Core 5). To start Lokkit, type **lokkit** in a terminal window. The first lokkit screen is shown in Figure 2.12.

Figure 2.12 Lokkit Main Screen



You can navigate the menus using the Tab key and the space bar to toggle the equivalent of radio buttons, such as the **Enable** and **Disabled** options shown here. If you select **Enabled** on this screen, the default ruleset is applied. To edit any custom settings, press **Tab** until the **Customize** button is highlighted and then press **Enter**. The customization screen is shown in Figure 2.13.

Lokkit does provide a little more flexibility than the Security Level Configuration GUI discussed previously; however, it is still limited. By selecting an interface in Trusted Devices, all traffic from that interface will be permitted. This would typically be used to select the inside interface and designate it as trusted. You do have the option of enabling *MASQUERADE*. The interface you select is the one that will NAT outbound traffic; therefore, you would generally select your *external* interface. Some predefined services are

available, and you can enter your own service information in the “Other ports” section. Once you are satisfied with your choices, press **OK** and then **Enter**. This will take you back to the main screen, where you press **OK** and then **Enter** to apply the changes.

Figure 2.13 Lokkit Customization Screen



If you attempt to configure an interface for *MASQUERADE*, it must also be marked as trusted, or Lokkit will generate an error. Bear in mind that although *MASQUERADE* is limited, it has enough flexibility to configure a firewall similar to a typical home firewall/router device. This makes Lokkit a handy little utility to have in your repertoire should you need to configure a simple firewall quickly. The value of this utility is also increased, because it is available for a wide number of Linux distributions.

Firestarter

Firestarter is a GUI front end for netfilter and iptables that is designed to make it simple for average users to configure their firewalls and protect themselves. Firestarter runs on many Linux distributions, and the installation is supported by many automated package management systems (such as *yum*, *apt-get*, and *portage*). Firestarter is an excellent choice if your needs are relatively simple for

your firewall configuration. To install it manually, downloaded it from www.fs-security.com/download.php. Once it is installed, the first time you start the GUI interface you will need to perform some initial configuration. Follow these steps to configure firestarter:

1. Start the Firestarter GUI. In Fedora Core 5 this is done by navigating to **Applications | System Tools | Firestarter**. This will start the Firewall wizard. Click **Forward** on the **Welcome to Firestarter** screen.
2. On the next screen, select your Internet-connected (i.e. external) network device from the “Detected device(s):” dropdown box (see Figure 2.14), and place a checkbox in the “IP address is assigned via DHCP” box. This is similar to the way a home router/firewall would be configured. When satisfied, click **Forward**.

Figure 2.14 Firestarter Network Device Setup



3. The next screen is the “Internet connection sharing setup” screen (see Figure 2.15), which is basically where you enable NAT. If you want to NAT all of the outbound packets to the external IP address, place a check in the “enable internet connection sharing” checkbox. When this checkbox is enabled, you can select the local area network device (i.e. the inside interface). If you only have two interfaces, it should be selected by default. When finished, press **Forward**.

Figure 2.15 Firestarter Internet Connection Sharing Setup

4. On the final screen, leave the “Start firewall now” box checked and click **Save**. This will install a service to start Firestarter each time the system boots up. Firestarter will also change the default action for the chains to *DENY*; therefore, you must explicitly configure any ports you want to permit through the firewall.

The main Firestarter GUI is shown in Figure 2.16. As you can see, it has a straightforward interface. The **Status** tab gives you high-level information such as sent and received data counters per interface, and a list of active connections. When you click the **Stop Firewall** button, all of the iptables chains are flushed and the default action is changed to *ACCEPT*. This can be useful for troubleshooting issues to see if they are related to your firewall configuration.

The “Events” tab lists recent blocked connection attempts. The “Policy” tab is where you configure certain rules to permit desired traffic (see Figure 2.17).

Figure 2.16 Firestarter GUI



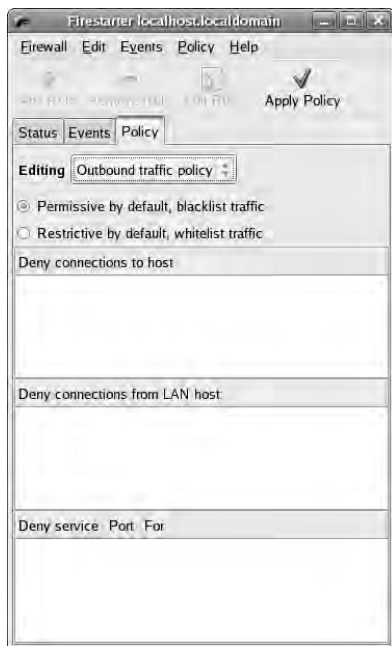
Figure 2.17 Firestarter Inbound Policy



For example, if there was a Web server running on the Linux host, you could use the “Policy” tab to permit inbound connections to TCP port 80. The “Editing” dropdown box allows you to choose between inbound and outbound rules to edit. For the Web server example, we selected “Inbound traffic policy.” The policy group you select when you click **Add Rule** determines where the policy is placed. The function of the various policy groups is outlined below.

- **Allow Connections From Host** This is used to configure a given IP address, hostname, or network. When you enter the IP information and create a rule in this policy group, all traffic from the configured source is permitted.
- **Allow Service** The allow service policy group is used to permit individual services. You can configure the source to be anyone including a specific IP, or network, or all local area network (LAN) clients. The LAN clients option permits the service through the firewall with a source address that is on the same subnet as the inside network adapter.
- **Forward Service** This option is used only when you are NATing. This allows the firewall to forward a specific port or range of ports, so that a service hosted on an internal NAT’ed device can receive inbound connections from the external network.

The “Outbound traffic policy” window shows a different set of policy groups (see Figure 2.18). There are also the additional radio buttons to select “Permissive by default,” “blacklist traffic,” or “Restrictive by default, whitelist traffic.” If you select the permissive option (the default), all outbound connections will be allowed and any rules you create will be *DENY* rules. This is the same default behavior of most home firewalls. If you select the restrictive configuration, the default target for the table is *DENY*, and any rules you create will be *PERMIT* rules.

Figure 2.18 Firestarter Outbound Policy

The function of the different policy groups toggle between “allow” and “deny,” based on whether you select restrictive or permissive mode. The policy groups are outlined here:

- **Allow/Deny Connections To Host** This policy group is used to globally permit or deny outbound access to a given host, IP address, or network range. This policy uses the destination to match the rule. You can use this policy group in permissive mode to list certain Web sites you do not want anyone to have access to.
- **Allow/Deny Connection from LAN Host** This policy group is used to permit or deny all access from a particular host, IP address, or network range. This policy uses the source to match the rule.
- **Allow/Deny Service** This policy group permits or denies traffic based on its destination port and source. When you are using permissive mode, this policy group can be used to block all access to the BitTorrent ports. The traffic source can be anyone; the firewall itself, LAN clients, or an arbitrary IP, hostname, or network range.

Configuring the policies will satisfy the bulk of what you need to accomplish, but there are some additional configuration options available by navigating to **Edit | Preferences**. Selecting **Interface | Events** allows you to configure some useful options. The “Skip redundant entries” checkbox only makes one event entry for sequential event entries. This helps prevent the event windows from being flooded by repetitive alerts. You also have the option of entering certain hosts or ports as being exempt from triggering the event log. After making your selections, click **Accept**.

Another preferences setting of note is under **Firewall | Network Settings**. This allows you to enable Internet connection sharing (the same as during the initial wizard), and enable the firewall host as a Dynamic Host Configuration Protocol (DHCP) server. This allows you to configure the Linux host similarly to a home firewall, which generally acts as a DHCP server in addition to performing NAT and acting as a firewall. The ICMP filtering window also allows you to filter ICMP packets. By default, the permit and deny rules configured by Firestarter apply to TCP and UDP, but not ICMP. This screen allows you to permit the desired types of ICMP traffic. Generally speaking, it is better not to allow any ICMP from the Internet to your firewall or internal network unless absolutely necessary.

One final setting you want to configure is under **Firewall | Advanced Options**. In the broadcast traffic section, check both options under **Broadcast traffic**. In general, you should not permit broadcast traffic to go through your firewall, as doing so poses a security risk. You also want to check the option to “Block traffic from reserved addresses on public interfaces,” which is a common filtering tactic. Because the “private” addresses outlined in RFC1918 should not be routed through the Internet, there is never a reason to receive traffic sourced from any of those addresses on your outside interface. If you do, it is almost always a hacker attempting to bypass a poorly configured firewall.

Short of any advanced packet mangling, there isn't much you can't accomplish using Firestarter as your configuration tool. If you need to implement a more advanced configuration, use an alternate tool, or generate the configuration using Firestarter and use those chains as a starting point to add your own more advanced options.

Easy Firewall Generator

Easy Firewall Generator is not a GUI per se, but it does help simplify your netfilter configuration and avoid the need to be familiar with the iptables syntax. By using the Web page at <http://easyfwgen.morizot.net/gen/index.php>, you can enter the relevant information and click the **Generate Firewall** button. As you select options, if additional information is needed click the **Generate Firewall** button and the page will refresh and provide the additional input fields. When all of the required information has been entered, the page will change to a text page that can be copied and pasted for iptables to read as a saved configuration. On Fedora Core 5 the iptables configuration is stored in `/etc/sysconfig/iptables`. Although this method requires you to replace the default iptables configuration file used by your distribution, it is fairly painless, and it supports all of the same basic functionality as Firestarter.

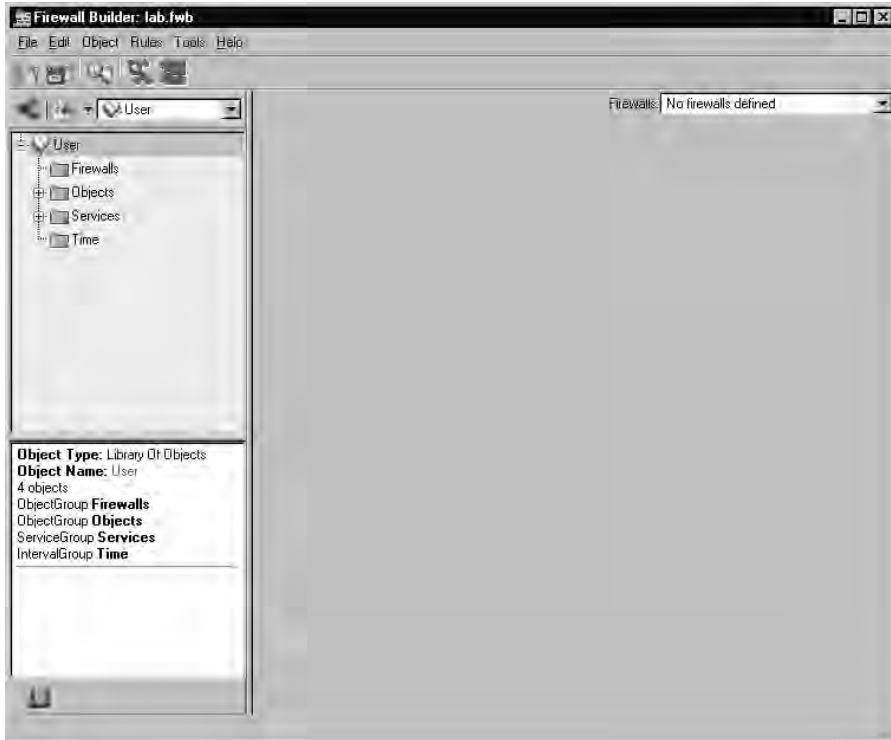
Firewall Builder

Firewall Builder is the most complete GUI offering for managing netfilter firewalls with features and capabilities comparable to some commercial firewall products. As is almost always the case, this functionality and capability come at a price: as far as netfilter GUIs are concerned, Firewall Builder is not the easiest to configure and use. If you want or need its superior management capabilities, however, the extra effort is well worth it. (Download firewall builder from www.fwbuilder.org/.) Firewall Builder manages netfilter firewalls as well as ipfilter, OpenBSD PF, and (commercially) Cisco PIX firewalls. Firewall Builder runs on many popular operating systems including Red Hat, Mandrake, SUSE, FreeBSD, Mac OS X, and Windows XP.

Firewall Builder operates differently than all of the GUIs covered so far. It uses an object-based approach. Essentially, you must define an object to represent any entity that you want to use in the firewall rules. In most cases this means a source, a destination, and a service (port) at a minimum. Both the configuration and the GUI bear a strong resemblance to that of the Checkpoint Firewall GUI. Once the objects are defined, you can drag and drop them into the rules in order to permit or deny communications between the two. For this example we use a Windows XP host to run Firewall Builder and configure a Linux netfilter firewall.

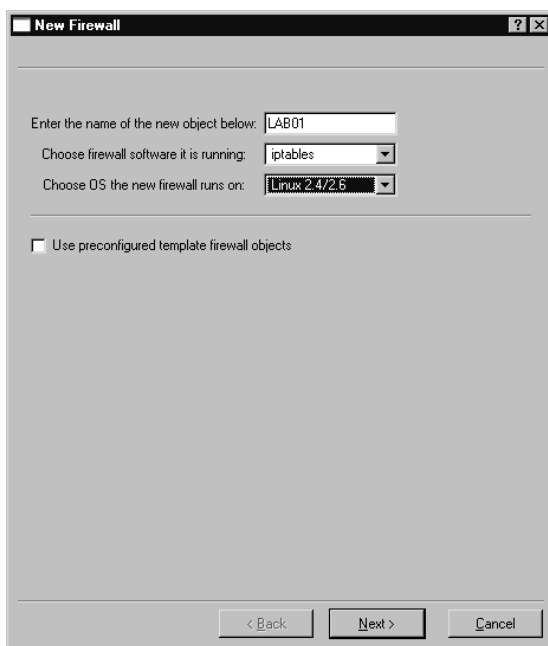
1. Install Firewall Builder.
2. Start the GUI by navigating to **Start | Programs | Firewall Builder 2.1 | FWBuilder**, which opens the main Firewall Builder window (see Figure 2.19). It is divided up into an objects tree (the left pane) and the dialog area (the right pane).

Figure 2.19 Firewall Builder



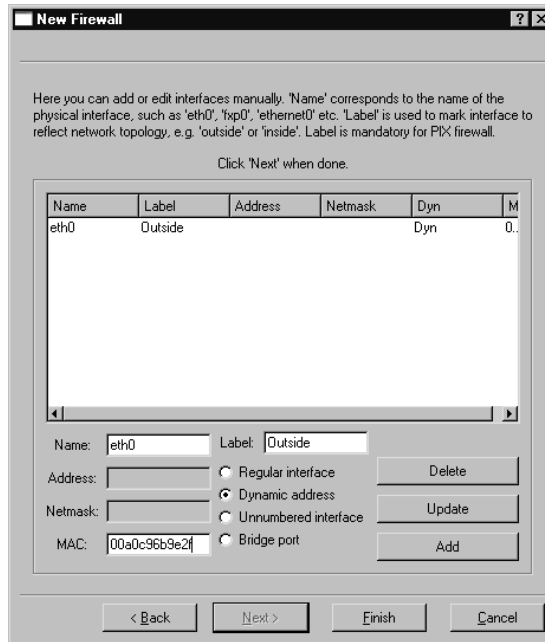
3. Initially, the dialog area will be empty. In order to add the first firewall (in this case a netfilter firewall) on the same host as you are running Firewall Builder, select **Firewalls** in the object tree.
4. Right-click and select **New Firewall**, which will open the New Firewall dialog box (see Figure 2.20).

Figure 2.20 FWBuilder New Firewall Wizard



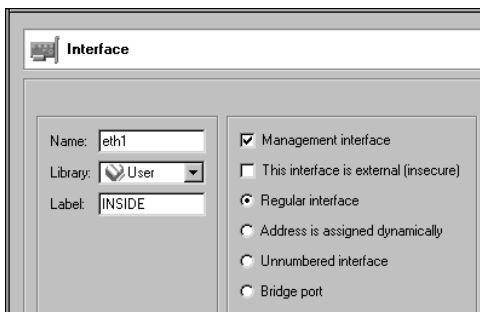
5. Enter the name for the new firewall (in this case LAB01).
6. For the firewall software, select **iptables**.
7. Choose **Linux 2.4/2.6** for the OS and click **Next**.
8. The next window allows you to configure the interfaces on the firewall. You can do so manually, or if the firewall is running SNMP, you can discover them via SNMP. You select **Configure interfaces manually** and click **Next**.
9. The manual interface configuration window is shown in Figure 2.21. Enter the relevant information for each network interface. The **name** must correspond to the actual interface name (same as if you entered ifconfig on the Linux host), such as *eth0*. The **Label** is a human friendly name for easy reference such as *OUTSIDE*. When you are done entering the information for a given interface click **Add**.

Figure 2.21 FWBuilder Manual Interface Configuration



10. When you have entered the information for all interfaces (typically an *INSIDE* and *OUTSIDE*), click **Finish**.
11. You must designate one of the interfaces on the firewall as the management interface, typically the *INSIDE* interface. Do this by navigating to the firewall in the object tree. As you select each interface in the object tree, there is a “Management interface” checkbox in the dialog area. Check this box for the interface you want to use. This will be the interface that FWBuilder uses to connect and upload the firewall rules to. The interface properties are shown in Figure 2.22.

Now that you have the basic firewall defined, you need to define something for it to talk to. In this case, let's assume that 192.168.1.0/24 is your internal network, and you want to allow outbound Web browsing and access to an internal Web server (WEB1). For starters, you need to create an object to represent the internal network. Follow these steps to create the network object.

Figure 2.22 Management Interface

13. Navigate to **Objects | Networks** in the object tree.
14. Right-click **Networks** and select **New Network**.
15. Enter **INTERNAL** for the name of the network, and use 192.168.1.0 for the Address field. Enter 255.255.255.0 for the Netmask and click **Apply**.
16. Let's go ahead and next create an internal Web server at 192.168.1.12. Right-click **Objects | Hosts** in the objects tree and select **New Host**.
17. Enter **WEB1** for the name of the object. Click the **Use preconfigured template host objects** check box and click **Next**.
18. Select **PC with one interface** and click **Finish**.
19. Expand the object tree to **User | Objects | Hosts | WEB1 | eth0 | WEB1**. Edit the IP address to be 192.168.1.12 and click **Apply**.
20. Next, define the appropriate services to allow Web browsing. Right-click **Services | TCP** and select **New Service**.
21. Enter **HTTP** for the name. Leave the source port ranges at zero, but change the destination port range to start and end at 80 and click **Apply**.
22. Repeat steps 20 and 21 for HTTPS on port 443 for secure Web pages.

This can be a lot of trouble; however, the real strength of an object-oriented approach is seen when it comes time to configure the rules. With all of

the appropriate objects in place, let's define the rules to permit the inbound HTTP traffic.

23. In the top panel of the dialog area right-click and select **Insert Rule**.
24. Allow inbound HTTP to WEB1. Click on **WEB1** in the object tree and drag it to the destination cell for rule 0.
25. Now drag the **HTTP** and **HTTPS** service from the object pane to the Service cell in rule 0.
26. Right-click the big red dot in the **Action** column and select **Accept**. This allows the inbound Web traffic to access WEB1.
27. To allow outbound Internet access, create another rule by right-clicking on rule zero and selecting **Add Rule**.
28. Drag and drop **HTTP** and **HTTPS** from the object tree into the Service column of rule one.
29. Drag the Network object **INTERNAL** from the object tree to the **Source** column of the new rule.
30. Right-click on the **Action** column for rule 1 and change the action to **ACCEPT**. Your policy should look like the one shown in Figure 2.23.

Figure 2.23 Sample FWBuilder Policy

Policy	NAT	Routing	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0			Any	WEB1	TCP HTTPS TCP HTTP	All			Any		Inbound Web
1			INTERNAL	Any	TCP HTTP TCP HTTPS	All			Any		Outbound Web

31. Although our rules seem simple at the moment, let's apply them to see how things work. First, save your work by navigating to **File | Save** or **File | Save As**.
32. Next, right-click the **LAB01 Firewall** and select **Compile**.

33. When the “Select Firewalls for compilation” window comes up, **LAB01** should be checked. When satisfied with your selection, click **Next**. When the compilation is complete you should see “Success” in the “Progress” column. After verifying that the compilation was successful, click **Finish**.

Tools & Traps...

Don't Block Yourself

Anyone who has spent any time configuring firewalls has learned the hard way to be very careful when configuring the rules. It is always a good idea to create the rules to PERMIT administrative access before any others. This is because as soon as you configure the default policies to DROP, your SSH connection will no longer be permitted unless you have it added to the access list. If you forget to do this, you could find that you no longer have remote access to your firewall after applying the policy. If that happens, you won't even be able to remotely connect to update the policy and change the ACLs.

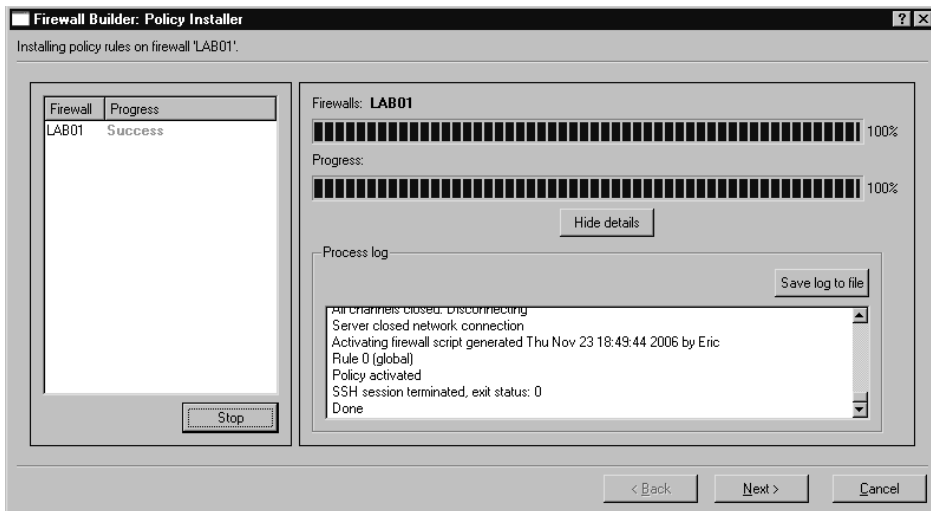
The next step is to tell FWBuilder where to find the SSH executables, because this is how FWBuilder uploads the configuration to the firewalls. You need to have SSH working on both the firewall and the FWBuilder console (assuming they are on different systems). (Detailed steps on using and configuring SSH are included in Chapter 2, “Securing Your Perimeter.”)

34. Select Edit | Preferences from the menu.
35. Select the **SSH** tab and click the **Browse** button.
36. Navigate to the location of your desired SSH utility (e.g., *plink.exe*) and click **Open**. Note that if you are using Windows for the FWBuilder host, you cannot select *PuTTY.exe*; you must use the command-line PuTTY program, *plink.exe*.
37. After selecting the SSH executable, click **OK**.
38. Right-click the **LAB01** firewall in the object tree and select **Install**.

39. Select the Firewalls you wish to install to and click **Next**.
40. Enter the username and password for the SSH connection.
41. All other fields are optional; however, it is recommended that you check “Store a copy of the fw on the firewall.” When satisfied with your choices, click **Ok**.

After the upload completes, you will get a status of “Success” (see Figure 2.24). Checking your firewall (*iptables -L*) shows you the new rules that are listed.

Figure 2.24 Policy Install Success



As you can probably see, once you have completed the up-front work of defining your objects, adding or modifying rules is simple. Additionally, unlike the other free GUI solutions, FWBuilder allows you to centrally and securely administer all of your (supported) firewalls from one location. When you use the aforementioned policy, Figure 2.25 shows a sample of the *iptables* rules that were generated.

Notice that the default chains have rules matching the rule you configured in FWBuilder, with a target of *RULE_<RULE NUMBER>*. These additional chains are used to configure the logging. There is also a rule at the beginning of all chains to *ACCEPT* traffic related to an established session. This is generally desirable but is still configurable. To remove this automatically generated rule,

select the firewall in the object tree and click on **Firewall Settings** in the dialog area. There is a checkbox that is selected by default called “Accept ESTABLISHED and RELATED packets before the first rule.” Although the FWBuilder policies you’ve configured can handle any basic rules you might need, there are still a few more bases to cover. If you need to NAT with your Linux firewall, configuring it with FWBuilder is easy. Follow these steps so that your Firewall will NAT all the traffic from the internal network to the DHCP address used on the outside interface. This configuration is also known as *source nat* because it is the source address that is being changed.

Figure 2.25 FWBuilder Generated Chains

Chain FORWARD (policy DROP)					
target	prot	opt	source	destination	
ACCEPT	all	--	anywhere	anywhere	state RELATED,ESTABLISHED
RULE_0	tcp	--	anywhere	192.168.1.12	tcp multiport dports https,http state NEW
RULE_1	tcp	--	192.168.1.0/24	anywhere	tcp multiport dports http,https state NEW
Chain RULE_0 (2 references)					
target	prot	opt	source	destination	
LOG	all	--	anywhere	anywhere	LOG level info prefix 'RULE 0 -- ACCEPT '
ACCEPT	all	--	anywhere	anywhere	
Chain RULE_1 (3 references)					
target	prot	opt	source	destination	
LOG	all	--	anywhere	anywhere	LOG level info prefix 'RULE 1 -- ACCEPT '
ACCEPT	all	--	anywhere	anywhere	

1. In the dialog area select the **NAT** tab.
2. Right-click and select **Insert Rule**. This will add a NAT rule number zero.
3. Drag your INTERNAL network object from the object tree to the **Original Src** column in the new NAT policy.
4. Drag the external interface on the firewall from the object tree to the “Translated Source” column in the NAT policy.

That’s all there is to it. Save, compile, and install the new policy. Now traffic originating from the internal network will be NAT’ed to the IP on the external interface. Although this source NAT configuration will allow all your internal users to reach the internet, you will need to use *destination NAT* if Internet users need to reach an internal server. Because the internal server is using a private IP address (which is not routable on the Internet), you need to translate this destination to an IP address that the external users can reach. To configure packets destined for the firewall’s single public IP address to an inside resource using destination NAT, follow these steps.

1. In the dialog select the NAT tab
2. Right click on the rule number zero of the existing NAT rule and select **Add Rule Below**.
3. Drag the firewall OUTSIDE interface into the Original Destination column of the new rule.
4. Drag the appropriate services (i.e. HTTP for web access) into the Original Service column of the new rule.
5. Drag the internal server into the translated destination column of the new rule.

Another nice feature is being able to create a time policy (e.g., if you only want the internal systems to be able to surf the Internet from noon to 1:00 P.M., you can easily make that adjustment).

1. In the object tree, right-click **Time**, and select **New Time Interval**.
2. In the “Name” field we’ll call this rule **LUNCH**.
3. In the two time fields provided, enter a time for the rule to START and a time for the rule to STOP. In this case we will enter 12:00 and 13:00 and leave the date field as zeros. The day of the week can stay at -1, which means all days. When done, click **Apply**.
4. Drag the **LUNCH** time interval, from the object tree to the **Time** column of rule # 1.

Now, rule # 1 (which permits outbound Web surfing) will only be active from noon to 1:00 P.M. The ability to configure the rules to be active based on the time of day is a very powerful feature. If the organization is a strictly 8:00 A.M to 5:00 P.M type of place, you could configure the firewall to disable all access during non-business hours. Alternatively, certain non-business-related protocols (e.g., instant messenger, file sharing, and so on) could be enabled after the normal business day ends. While not the easiest GUI to use, FWBuilder is definitely the most full featured, and the only one offering features you would expect to find in a commercial product.

Other GUIs

Although there are too many netfilter GUIs to cover them all extensively, we have tried to cover some of the best ones available. If none of the ones covered strike your fancy, or if you just like to experiment and see what else is out there, you might want to investigate some additional offerings. If you are running KDE look into Guarddog from www.simonzone.com/software/guarddog/#introduction, which is aimed at novice to intermediate users and offers the ability to define security policies based on logical groupings called network “zones.” The Turtle Firewall Project (www.turtlefirewall.com/) allows you to administer your firewall host via a Web interface. While there is no substitute for a good understanding of the command-line configuration of iptables, for an uncomplicated firewall configuration many of these GUIs allow you to get your firewall up and running quickly and without having to read the iptables man page.

Smoothwall

Smoothwall (<http://smoothwall.org/>) is a firewall in its own category. First, let's clarify some basic nomenclature. Smoothwall.org is the site for SmoothWall Express. SmoothWall Express is a free open-source firewall solution. *Smoothwall.net* is the home of SmoothWall Limited, which produces several commercial security products, including a version of the SmoothWall firewall. SmoothWall differs from the other solutions covered here in that it is a dedicated firewall device. Other solutions using netfilter and optional GUIs to configure the firewall can be run on a workstation. You can still use the firewall system as a normal workstation, but it's not recommended. If you want to harden the firewall (as you should), you need to remove unneeded services and software from the system, and update all of the remaining software. SmoothWall takes a different approach in that all of this is done for you. When you install SmoothWall, it wipes out the filesystem and installs a secured version of Linux on the hard disk, along with the SmoothWall software. The SmoothWall firewall has no GUI on the system, only command-line access and administration via the Web management interface. SmoothWall is meant to be a firewall and nothing more.

With that in mind, there are several advantages to this approach. Foremost, you don't have to learn how to harden your Linux distribution so that it will be

secure enough to use. Further, unlike installing Linux and then learning iptables syntax, with SmoothWall you don't need to know Linux. The installation menu walks you through configuring the minimum settings so that you can then use the Web interface to configure the firewall functionality. You don't need to know anything about Linux to get SmoothWall up and running (though it never hurts). The fact that the SmoothWall firewall is already stripped down and unneeded software and services are removed means that you can get the maximum performance out of an old computer without having to spend a lot of time trying to tweak a full (normal) Linux distribution.

Installing Smoothwall

The simplest way to install SmoothWall Express is by downloading the *.iso* image from <http://smoothwall.org/get/>. It is advisable to read along with the manuals located at <http://smoothwall.org/docs/>. The documentation provided with SmoothWall Express is exceptional among free products, and all of the installation screens are shown in the PDF installation guide. This installation method is used as we walk through installing SmoothWall Express.

1. After burning this image to a CD-ROM, boot the prospective firewall with the CD-ROM in the drive.
2. The boot screen will look typical of many Linux distributions. It will warn you that installing SmoothWall Express will delete all data on the hard drive. To continue with the installation, press **ENTER**.
3. The installation then shifts into a DOS-like GUI interface. Navigation is accomplished using the TAB, arrow, and ENTER keys. You will be prompted to insert the installation CD and press **OK**. This is done in case your system cannot boot from a CD-ROM and you used a boot floppy to begin the installation. Either way, ensure that the CD-ROM is (still) in the drive, highlight **OK**, and press **ENTER**.
4. You have to select **OK** twice before the hard disk will be repartitioned and all data lost.
5. When prompted, select **Probe** to allow the installation routine to see what network cards it can detect.

SmoothWall uses a concept of interface colors to denote their trust level, and you will begin seeing them referred to in that fashion during the installation process. For example, your inside interface is assumed to be the trusted traffic and is designated as the GREEN interface. Various dial-up, Integrated Services Digital Network (ISDN), or DMZ interfaces are designated as ORANGE interfaces. You can also have a combination of colors indicating your interface configuration. If you use an additional Ethernet interface as the untrusted (*OUTSIDE*) interface, in SmoothWall parlance that would be GREEN + RED.

After the files are installed successfully, you are given the opportunity to restore your configuration from floppy disks. This is useful if you are upgrading or migrating to new hardware. In this case, we select **OK**.

1. Select your keyboard layout (in my case “US,” and select **OK**).
2. Select a hostname for the firewall (e.g., smoothwall) and select **OK**.
3. The next screen allows you to enter proxy server information in case you need to go through a proxy for the firewall to retrieve Web updates. If you are using a proxy, enter the appropriate information here; if not, select **OK**.
4. The next couple of screens allow you to enter configuration information for an ISDN or ADSL connection. The assumption of the installation process is that your *INSIDE* (trusted) interface will be an Ethernet interface, and the *OUTSIDE* (untrusted) interface will be either an Ethernet, ISDN, or ADSL. If you are using one of these, enter the appropriate information. If your *OUTSIDE* interface is a normal Ethernet interface (e.g., from a cable modem), select **DISABLE** for both the ISDN and Asymmetric Digital Subscriber Line (ADSL) configuration screens.
5. The next screen allows you to review and edit your network configuration. If you are using an Ethernet interface for both, you need to select **GREEN + RED** for the network configuration type. Check each menu option here and ensure that both interfaces have been recognized, have a driver installed, and have IP address settings. Commonly, the RED interface uses DHCP and the GREEN uses a static IP address, so that internal hosts can configure the firewall as their default gateway out to the Internet.

6. When you are satisfied with all the settings, select **DONE**.
7. You will be asked if you want to enable the SmoothWall firewall to serve as a DHCP server. This is the same configuration as most home firewalls, acting as firewall, gateway, and DHCP server. If you do not already have a DHCP server in your network, enable it. Fill in the desired values for the various fields. Most of the settings are not mission critical, but one setting to take note of is the lease duration. If your lease duration is too long, you will slowly lose IP addresses from systems that did not get the chance to release the address properly prior to going offline (such as from a crash or power outage). If the lease time is too long, this IP address attrition can exhaust the DHCP scope and leave no address available for other users. A 24-hour lease is not uncommon, and generally the larger the network the shorter the lease duration you will want. If you are unsure about DHCP, you can leave it disabled. The DHCP settings are easily configured later from the Web interface.
8. The next several screens allow you to enter the password for various accounts used by the firewall. Here are summaries of these accounts.
 - **Administrator** This is used for administering the firewall via the Web interface. This account is only for accessing the Web interface and cannot be used to login to the Linux OS on the firewall directly.
 - **root** This is a local Linux account that is used for command-line access on the firewall itself.
 - **setup** This is a local Linux account that is used to run the setup program, which automatically starts when you login as setup. The setup program allows you to configure some of the network settings if they need to be changed after the initial installation.
9. After you configure the final password, the CD-ROM will eject and the system will reboot. When the system comes back up, you can log in directly via the console using the root account, or the preferred method is to log in to the Web interface. The default Web interface is found at <http://smoothwall:81> and the secure HTTP is found at

https://smoothwall:441. Both the root and the setup account can also login via SSH, which is configured by default on port 222.

Configuring SmoothWall

When you first log in to the SmoothWall Web interface, the screen will look like the one shown in Figure 2.26.

Figure 2.26 SmoothWall Web Interface



There is some information available before logging in, such as the number of users and average load on the firewall. As soon as you click on a menu item at the top you are prompted to authenticate with the Web admin user. By default, the account name is “admin.” One of the first things you should do is enable SSH access, which is disabled by default. This allows you an additional way to manage the firewall if something goes wrong with the Web server or the firewall filters. You can enable SSH by clicking on the **Services** tab, and then selecting **Remote Access**. Next, place a check in the box next to SSH and click **Save**. You can verify what services are running by clicking the “About your smoothie” tab. There are three screens available under this tab. The *status* screen shows which services are running. The *advanced* screen

shows more detailed information regarding memory usage, hard disk usage, network interface settings, and uptime. The *traffic graphs* screen shows input and output rates for all interfaces.

After enabling SSH, you should be able to connect on port 222. An example using *openssh* would be:

```
ssh <IP> -l root -p 222
```

Now that you have a backup way to get into the firewall, the next priority is to update the firewall. Although you don't have much of a configuration to warrant making a backup before applying the patches, it is still a good habit to get into. By selecting the **maintenance** tab and then the **backup** screen, you have a couple of options. The "Create backup floppy disk" button will write the configuration information directly to a floppy disk. Given the relatively unreliable nature of floppy disks, you should choose the "Create backup floppy image file" option. This creates and downloads an *.img* file to the system you are using for Web administration. You can store this file on a more reliable media, and then write the image to a physical floppy disk at a later date using a utility such as *rawwrite*. Once you have made a backup, you can safely apply the firewall updates.

Firewall updates are another area where the SmoothWall designers have made things as painless as possible. Click on the **maintenance** tab and you will see two sections on the **updates** screen. The top section shows *installed* updates and the bottom one shows *available* updates. To update the firewall, go to <http://smoothwall.org/get/>. In the "Latest Updates and Patches" section there is a small link called **updates archive**. Click that link and on the following page, download all the available updates to your local system.

NOTE

All of the updates must be applied sequentially; the SmoothWall updates are not cumulative updates. Apply update 1 first, then update 2, and so on, until you have applied all of the updates currently available for your firewall. At the time of this writing there were seven updates available for download.

The bottom of the **Maintenance | Updates** page has a box to upload an update. Click **Browse** and select the first update, and then click **upload**. The firewall automatically installs the patch as it is uploaded and, when finished, the page will refresh and show the updated listed in the “Installed updates” section. Continue this process until all available updates have been completed. A partial list of the successfully installed updates can be seen in Figure 2.27.

Figure 2.27 SmoothWall Installed Updates

Updates
See the latest updates and fixes available for your SmoothWall, and an installation history of updates previously applied.

Installed updates:				
ID	Title	Description	Released	Installed
001	fixes1 update	This update contains an updated kernel to version 2.4.24 to correct recently discovered, locally exploitable, vulnerabilities. It also corrects known issues and a problem with dynamic DNS.	2004-01-12	2006-11-26
002	fixes2 update	This update contains an updated kernel to version 2.4.25 to correct recently discovered, locally exploitable vulnerabilities.	2004-02-26	2006-11-26
003	fixes3 update	This update contains an updated kernel to version 2.4.26 to correct recently discovered, locally exploitable vulnerabilities. It also updates Apache and OpenSSL to correct several recently discovered vulnerabilities. In addition, it adds support for the latest SpeedTouch modem (revision 4). It also corrects issues with custom dyndns.org accounts.	2004-05-26	2006-11-26

One final configuration option that should be a part of any firewall setup is providing the firewall with a good way to synchronize its clock. Accurate time is important for many reasons, one of which is to make sure your logs have accurate time stamps. Without accurate time stamps, it will be more difficult, if not impossible, to reconstruct events later if there is an intrusion. You can configure the time source on the “Services” tab using the “time” screen. You should use the drop-down box to select the appropriate time zone. While SmoothWall does not give you the option to configure network time protocol (NTP) security, it does give you the option of using random public servers. In this fashion, even if it pulled time from one that was too far off

(either accidentally or maliciously), the next time it is checked (a different server) would likely correct itself. To enable SmoothWall to retrieve the time from a public time server, check the **Enabled:** checkbox and then click **Save**.

With all of the basic administrative configuration out of the way, the actual rule configuration is next. SmoothWall relies heavily on the security level of the interfaces for access permissions. By default, all traffic will be blocked that enters on a RED interface and is destined for an address via a GREEN interface, unless it is specifically configured to be permitted. Similarly, traffic is blocked by default from an ORANGE interface to a GREEN interface. Traffic from a more secure interface to a less secure interface is permitted by default. This behavior is similar to several other commercial firewalls including the Cisco PIX/ASA. What all this means is that for your users to access the Internet, you don't need to configure anything at all.

On the other hand, suppose you wanted to permit inbound access to a Web server (GREEN interface) with an IP address of 192.168.1.63 from any host on the Internet (RED interface). You would configure this by selecting the **Networking** tab and the **Port Forwarding** screen (see Figure 2.28).

Figure 2.28 SmoothWall Port Forwarding

Port Forwarding
Forward ports from your external IP address to ports on machines inside your LAN or DMZ.

Add a new rule:
External source IP, or network (blank for "ALL") Source port or range: Destination IP: Destination port*:
 Enabled:


* If blank, then the source port will be used as the destination port

Current rules:

Proto	External source IP	Source port	Destination IP	Destination port	Enabled	Mark
TCP	ALL	80	192.168.1.63	80	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Leave the source IP blank (for ALL) and enter a source port of 80. For the destination IP, enter the internal server's IP of 192.168.1.63, and for the destination port, enter 80 for HTTP. When finished, click **Add**. If you need to

permit HTTPS, you need to repeat the process with 443 as the destination port. By using SSH to connect to the firewall directly (SmoothWall uses port 222 for SSH), you can list the netfilter rules using *iptables -L* and see where the HTTP and HTTPS rules were added.



```
Chain portfwf (1 references)
target      prot opt source                destination
ACCEPT     tcp  --  anywhere              192.168.1.63          state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere              192.168.1.63          state NEW tcp dpt:https
```

If you have three interfaces in a one-legged DMZ design, the DMZ interface is labeled as ORANGE. If you need to permit access from the DMZ into the trusted (GREEN) network, the process is a little different. You would then navigate to the **networking | DMZ pinholes** screen. The interface is very similar to the port forwarding with the exception that there is no field to specify the source port.

At times, an internal system’s permissions may allow it to communicate with a device outside the firewall (on the RED interface); however, you may wish to block the communications completely. If you do this, any access by the blocked site will fail, even attempts to respond to an internal trusted system’s request. You can configure this on the “Networking” tab, using the “ip block” screen. Enter the source IP address or name to block, and click **Add** to save the rule. You also have the option to enable logging for the blocked attempts.

With the basic firewall rules and maintenance configured, there are a few “extras” that are nice to see in a free product. One of these is the built-in Intrusion Detection System (IDS). Because it uses Linux as its base operating system, it conveniently includes Snort IDS; all you have to do is enable it. Enable Snort by selecting the “Services” tab, and then the “Intrusion Detection System” screen. Place a check next to Snort and click **Save**. The Snort alerts and other logs can be viewed on the “logs” tab. There are several subscreens that include a drop-down box to select what subset of logs you want to see, such as SSH, SmoothWall (which will show your recently applied patches), and several more. The “Web proxy” screen is only useful if you are using the Web proxy feature of the firewall. The “firewall” page shows all blocked connections and allows you to filter by month and day. Lastly, the

IDS screen shows events logged by the Snort IDS. Unfortunately, SmoothWall Express does not support remote logging natively, while the commercial offering does. It does, however, allow you to export the log files to a text file.

Another nice option is the dynamic DNS support. There are various dynamic DNS services available that will allow you to use a consistent DNS name to refer to a system whose IP address is dynamic via DHCP. In order to do this, you typically must install a small program on the host system that will periodically contact the dynamic DNS server and alert them to your current IP address. The service then uses this information to update their DNS records so that people can locate the system via DNS. The SmoothWall firewall has the capability to perform these updates for you, to the major dynamic DNS providers. You can configure dynamic DNS support by selecting navigating to the **services | dynamic dns** page. Use the drop-down menu to select the dynamic DNS service you are using, fill in the rest of the information, and click **Add**. The firewall will then make the updates to the service and all of the hosts to IP mappings can be maintained in one place rather than having to install an agent on all of the systems that need dynamic DNS functionality.

SmoothWall Express is a very well-built firewall package. The documentation is very good, and the setup and management are straightforward and understandable. You don't have to know anything about the underlying operating system. With all of the advanced features such as traffic graphs, intrusion detection, and respectable logging, it deserves a top spot on the list of contenders for "best free firewalls." If you want the efficiency of running your firewall on Linux without having to learn how to secure your Linux installation, give SmoothWall a try.

Configuring Windows Firewall

Although there is a plethora of commercial firewalls available to run on Windows, the field is a lot smaller when it comes to free offerings. Additionally, while there are several quality offerings for Windows as personal firewalls, there are not any free ones that are appropriate to protect your network perimeter. The built in **Windows Firewall** included with Windows XP and 2003 is very limited in its configuration options and is only appropriate as the personal firewall it was intended to be. The Windows Firewall included

with Windows Vista is supposed to incorporate increased flexibility and control over the filtering rules, so that might be something to keep an eye on when it is released. Given this, configuring the Windows Firewall is covered in Chapter 3, along with content on other personal firewalls.

Providing Secure Remote Access

Sooner or later odds are good that you will either want or need the ability to work remotely. Providing remote access must be undertaken very cautiously, because, as soon as you allow employees to connect to the corporate network, you have to some degree, extended your network boundary to their workstations. This means your network's security is now only as good as the security of the remote user's system or network. In many cases this borders on no security at all. That is why remote access must only be granted after careful consideration and planning. While the different types of remote access pose different levels of security risk, there are some planning and configuration steps that are common to all of them.

The first task is to determine what type of remote access is appropriate. With a virtual tunnel network (VPN), it is as if the remote workstation is on the corporate network. This generally provides the greatest level of functionality, but also poses the greatest risk. If the remote system is compromised, an attacker is effectively inside your corporate network. While there are steps you can take to mitigate these risks, they may be time- and effort-intensive. To plan, configure, and properly secure a VPN solution is the most involved choice of the various remote access solutions you could provide.

Another option is to provide remote desktop functionality. This would allow a remote user to see and use the desktop of a system at work. A remote desktop acts as if the user is at work, while a VPN acts as if the user's computer is at work. This type of solution is slightly easier to implement, because you can typically isolate the traffic that needs to be permitted through the firewall to a single TCP port. Many of the same risks exist, however, in that if an attacker manages to gain access to an internal desktop remotely, it is usually easy for them to move information out of the network or otherwise cause mischief. Another key consideration with this type of solution is that you need to have a computer at home and a computer at work. With the

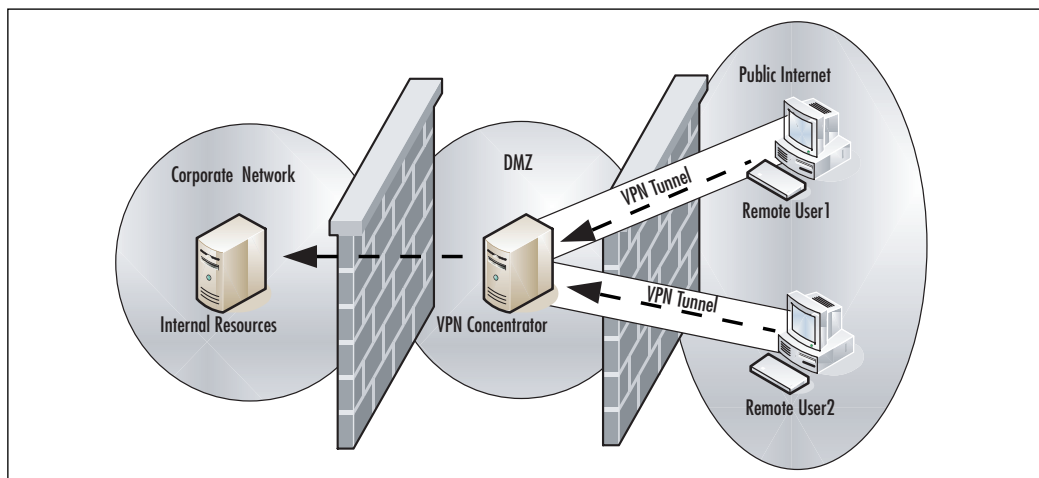
VPN option, you only need to use one system, so if the user has a laptop, it can be used while they work remotely.

The last and least functional option is that of a remote shell. Because most average users don't operate extensively (if at all) in a console (i.e., text only) environment, this type of remote access is generally most viable for network administration personnel. While it may be impossible for accountants to operate their accounting program without a GUI, many network tasks and most firewall administration tasks can be performed with only terminal access. Because the widely used Telnet protocol sends all data unencrypted, any sensitive tasks should only be performed using a secured protocol such as secure shell (SSH), or Telnet over a Secure Internet Protocol (IPsec) tunnel.

Providing VPN Access

A virtual private network (VPN) is exactly what it sounds like, the network connection you create is virtual, because you can use it over an otherwise public network. Basically, you take two endpoints for the VPN tunnel, and all traffic between these two endpoints will be encrypted so that the data being transmitted is private and unreadable to the systems in between. Different VPN solutions use different protocols and encryption algorithms to accomplish this level of privacy. VPNs tend to be protocol independent, at least to some degree, in that the VPN configuration is not on a per-port basis. Rather, once you have established the VPN tunnel, all applicable traffic will be routed across the tunnel, effectively extending the boundaries of your internal network to include the remote host.

One of your first considerations when planning to implement a VPN solution is the network design. Because the VPN tunnel needs two endpoints, one will be the remote workstation. The other will be a specially configured device for that purpose. This is generally called a VPN concentrator, because it acts as a common endpoint for multiple VPN tunnels. The remote systems will effectively be using the concentrator as a gateway into the internal network, so the placement of the concentrator is important. In a highly secured environment, the concentrator is placed in a DMZ sandwiched between two firewalls—one firewall facing the Internet and the other facing internally (see Figure 2.29). While this type of arrangement is the most secure, it takes more hardware to implement.

Figure 2.29 VPN Concentrator Design

Another way to place the VPN concentrator inside a DMZ is to use an additional interface on the firewall as the DMZ in a “one-legged” configuration. This saves you having to implement an additional firewall, but still provides some isolation between the concentrator and the rest of the internal network. If an attacker compromised a remote host who was VPN’d into the concentrator or compromised the concentrator itself, they would still have a firewall between them and the internal network. The least preferable option is to place the concentrator inside the internal network. With this type of design, if the concentrator is compromised, the attacker would have full access to the internal network, with no firewalls to inhibit their activities. With any of these designs, you will have to permit the required ports through the firewall and forward them to your VPN concentrator.

Another consideration is the type of VPN protocol you want to use. IPsec is still the most widely deployed VPN technology for good reason. One is interoperability. As a widely used and tested standard, IPsec will work with virtually any modern firewall and operating system. The disadvantage of IPsec is that it can sometimes be difficult to configure properly, and there is zero margin for error on the configuration. Both ends have to use the same parameters for encryptions, hashing, and so forth, or the tunnel cannot be established. Secure Sockets Layer (SSL) is an increasingly popular choice for VPNs, largely because of its simplicity to implement.

Once you have chosen a design and VPN technology, you need to consider the administrative ramifications of offering remote access. Some level of training will be required, at the very least so that they can sue the VPN software. You should educate the users on good security habits as well. A determination will also need to be made as to whether remote users are allowed to use their own personal computers, or if they must use a company-provided computer for remote access. The former option carries with it many risks. When remote users connect their personal computers to the corporate network (via a VPN), they may have spyware, a virus, or any number of potentially damaging conditions present on their systems. Because you probably don't have any administrative access to their systems, you may have no way to secure the personal systems even if you wanted to. This is why most companies require that only corporate resources be allowed to connect to the company network. In the case of remote users, this typically means a company-provided laptop, but I have also seen instances of older desktops being sent home for remote access.

A final consideration is one of hardware selection. Normal desktop productivity applications typically place very little strain on an even remotely modern processor. The same is not true when it comes to VPN connections. A single VPN connection requires little overhead and rarely impacts the remote user's system unless it is especially underpowered. For the VPN concentrator, however, it will handle the encryption and decryption of multiple connections, in addition to managing the volume of network data that will be accessed through it. For this reason, if you anticipate more than just a couple of VPN connections to be used simultaneously, you will want to test and evaluate your hardware needs.

Using Windows as a VPN Concentrator

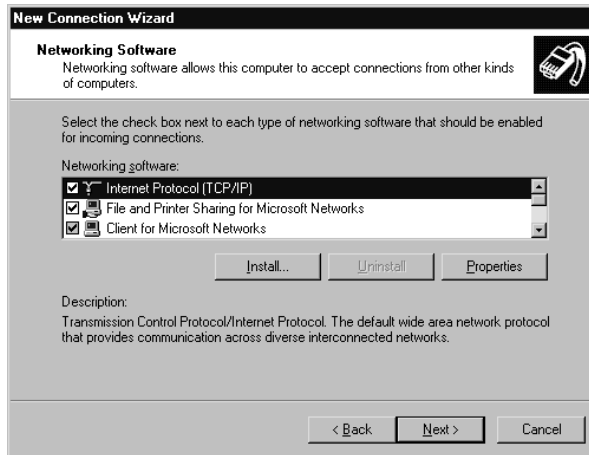
For a simple VPN solution servicing a small number of users you can use a Windows 2000, XP, or 2003 system using native software. This has the advantage that you are not using any third-party software, so installation and support may be easier. Not only is the configuration reasonably simple, but it may be easier to sell to upper management, because it doesn't involve any non-Microsoft software being installed or relied on. The Microsoft VPN connection uses the point-to-point tunneling protocol (PPTP), which is not

compatible with other types of VPNs such as IPsec-based or SSL-based VPNs. PPTP is a widely supported and relatively lightweight protocol. PPTP support can be found on Linux, MAC OS X, and Palm Personal Digital Assistants (PDAs).

To configure a Windows host as a VPN endpoint using Windows 2000, follow these steps.

1. Navigate to **Start | Control Panel | Network Connections**.
2. Click **Create New Connection**
3. On the welcome screen, click **Next**.
4. In the New Connection Type windows, select **Set up an advanced connection** and click **Next**.
5. In the Advanced Connections Options window, leave the default **Accept incoming connections** checked and click **Next**.
6. On the “Devices for incoming Connections” window, click **Next**. Any modems you have installed will be listed; however, for a network connection, you can leave them unchecked.
7. On the next screen, select **Allow virtual private connections** and then click **Next**.
8. On the “User Permissions” window, place a check next to the user accounts you wish to be able to connect via VPN and then click **Next**.
9. On the networking software screen, highlight **Internet Protocol (TCP/IP)** and click **Properties** (see Figure 2.30).
10. The Incoming TCP/IP Properties window is where you configure the most important settings for the VPN connection. The **Allow callers to access my local network** *must* be checked or the VPN connection won't work. For TCP/IP address assignment, you need to plan accordingly. If you leave the default selected, the remote systems will be assigned an IP address via DHCP as they connect.

Figure 2.30 Network Software



11. When finished configuring the TCP/IP properties, click **OK**.
12. On the Networking Software window, click **Next**.
13. Click **Finish**.

After completing these steps the server should be ready to accept an incoming VPN connection. The next step is the client side of the configuration. We will walk through this configuration using a Windows XP system, as the client, in order to make a VPN connection to the Windows XP VPN server.

1. Navigate to **Start | Control Panel | Network Connections**.
2. Click **Create New Connection**
3. On the welcome screen, click **Next**.
4. Select **Connect to the network at my workplace** and click **Next**.
5. Select **Virtual Private Network Connection** and click **Next**.
6. On the Connection Name window, choose a descriptive name for the connection and then click **Next**.
7. The next window is the VPN Server Selection screen. Enter an IP address or host name and click **Next**.

8. On the final screen you have the option of adding a shortcut for the connection to your desktop. Select the checkbox if you want to create the shortcut and then click **Finish**.

The shortcut that is created can be opened to initiate the VPN connection. You will be prompted to enter the login credentials to use for the VPN connection (see Figure 2.31).

Figure 2.31 Windows XP VPN Login



After entering your username and password, click **Connect**. If the connection is successful, you should see a pop-up in the system tray indicating that you are connected. Once connected, you can route traffic through the VPN server.

NOTE

In order for the connection to work, you must have a password for the account you are using to connect via VPN. If the account has no password, you will not be able to connect.

One final thing you may need to configure is the routing table on the client system. When you make the PPTP connection, a *default route* is added to the client's routing table after their existing default route. You can view the

routing table on Windows using the **route print** command. The default route has 0.0.0.0 for the network destination. This means that any traffic destined for an IP address that the client doesn't know where to send it, will continue to go out the interface it was using before the PPTP connection was made. If you need to change this behavior so that all non-local traffic goes through the VPN tunnel, you can alter the routing tables with a simple batch file.

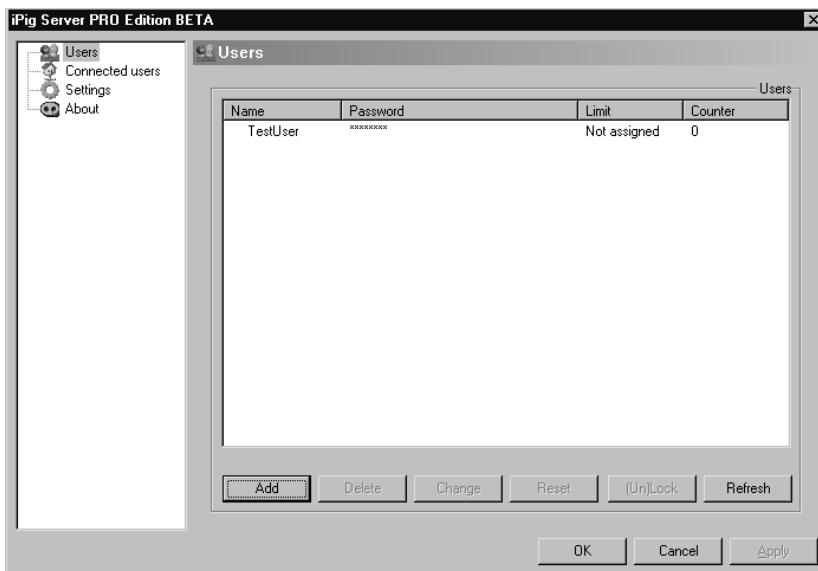
iPig

iPig is a VPN solution provided by iOpus Software at www.iopus.com/ipig/. The client is freeware, and the server portion (the VPN concentrator) is offered as an unlimited commercial product or as a five-user iPig Server Express Edition for free. The five-user limit is for simultaneous connections to the VPN server. You can create more than five user accounts to use the VPN, but they cannot all use the server at the same time. Both the commercial and the free versions use AES256 for their encryption and run on Windows 2000, XP, and 2003. If you download the iPig client and do not install the iPig server, you can still use the client. In this configuration the client will connect to an iOpus-controlled server on the Internet. You are limited to 10MB of “free” bandwidth before you must pay for additional bandwidth. Instead, you should install your own server side component, which is the iPig Server Express Edition product. We walk through setting up the iPig server component first, and then the client software.

Installing the iPig Server Express Edition

Download and install the iPig server software. There are no unusual options during the installation process. In typical fashion, the install begins with a welcome screen and then asks you to accept the License agreement. The next screens lets you choose the installation directory and start menu folders. When the installation is complete, you will get a window informing you that the server started successfully. You can configure the iPig server options by navigating to **Start | All Programs | iPig Server | iPig Server**. The main configuration screen is shown in Figure 2.32.

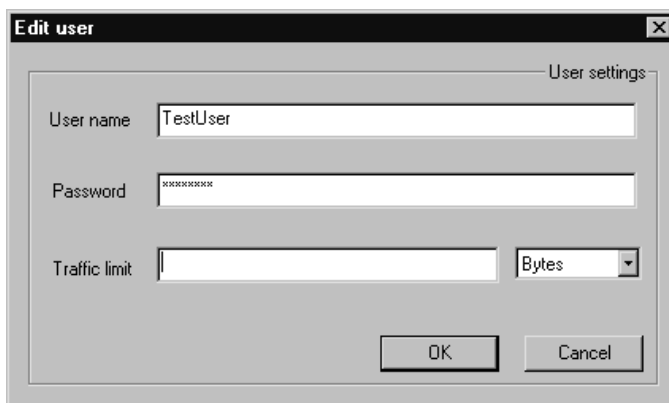
Figure 2.32 iPig Server Configuration



The first step is to define the VPN users.

1. Select **Users** in the left pane and then click **Add** in the right pane.
2. The Edit user window allows you to enter a username, a password, and a traffic limit if desired (see Figure 2.33). You can artificially throttle back the VPN users to make sure they do not consume too much of your Internet bandwidth and negatively impact Internet access for the local network users.

Figure 2.33 iPig Edit User



3. After entering the user information, click **OK**. Back at the main configuration window there are a few additional settings to configure. One option of note is under the “Settings” section in the left pane. If you select “Settings” you can then edit the Server port that is used to listen for incoming connections. Generally speaking, a non-default port is almost always better than using the defaults. You should also change the Log type from **None** to **Small** or, even better, **Full**, depending on how much VPN traffic you expect to see. The server log is located in the `\iPig\server\vpn_log.csv` file. The logging that is offered even in “Full” mode is pretty minimal, but it’s better than nothing.

Installing the iPig VPN Client

Installing the client is equally as painless.

1. Download the client installer and run the installation for the client setup.
2. On the initial welcome screen, click **Next**, select the radio button to **accept the license agreement**, and then click **Next**.
3. Choose the installation folder and click **Next**.
4. Choose the start menu folder and click **Next**, and then click **Install** on the next screen. You will need to reboot the system when the installation completes.
5. Start the iPig client program by navigating to **Start | All Programs | iPig WLAN Security | iPig Client**.

Once the client is started, there are a couple of settings you must configure.

1. First, select **Advanced Settings** in the left pane (see Figure 2.34).
2. Select the radio button next to “Use your own iPig server,” and enter the IP address and server port (11888 is the default port) and click **Apply**. Click **Connect** and select **User** in the left pane (see Figure 2.35).

Figure 2.34 iPig Client Advanced Settings

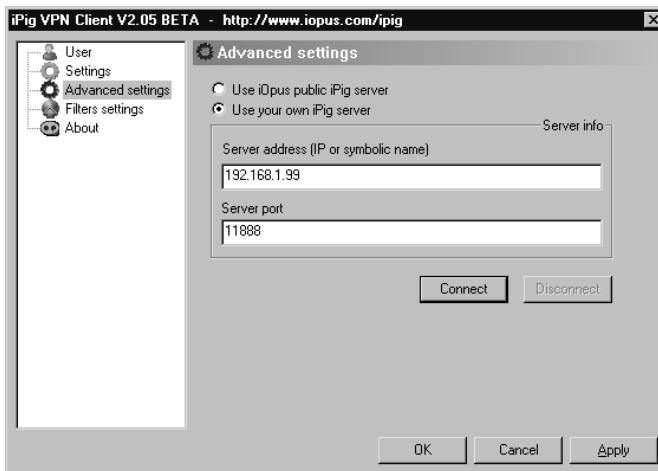
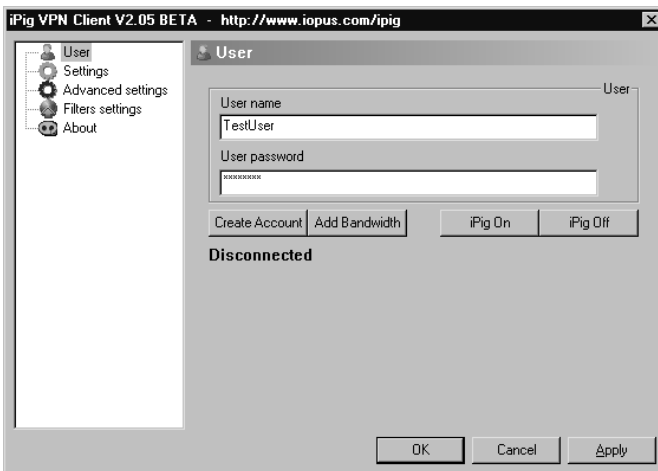


Figure 2.35 iPig Client User Settings



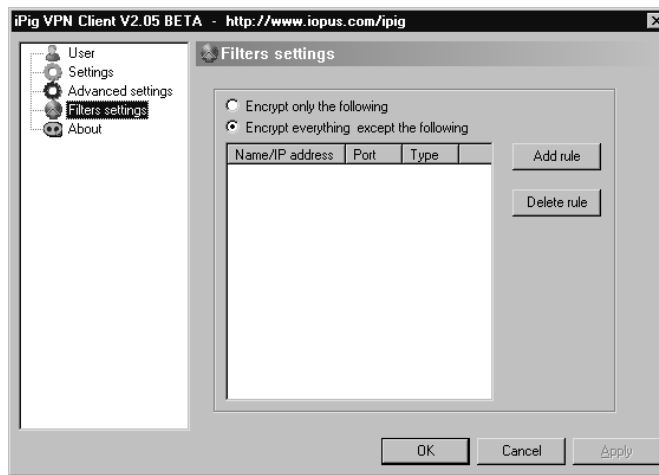
3. Enter the user name and password that matches the ones you defined in the iPig server.
4. When finished, click **iPig On**.

This is all that is needed to have the VPN tunnel up and working. However, there are some additional configuration options that would be advisable to configure.

In the Settings page, make sure you have **Log all Internet Access** checked. There is also a checkbox to encrypt UDP traffic. If you leave this unchecked, only TCP traffic will be encrypted. Depending on your needs, this may or may not be significant. One thing to consider is that if you encrypt UDP traffic, it will include DNS requests. Therefore, when the client requests an IP address to match a host name (e.g., www.syngress.com), the request is encrypted and sent to the iPig server, which then decrypts the request and sends it to its DNS server. In most cases, this shouldn't be an issue, but sometimes ISP's and others will configure their DNS servers to only answer queries when they come from their internal network.

Another option of note is the ability to configure encryption filters. The Filter settings window is shown in Figure 2.36.

Figure 2.36 iPig Client Filter Settings



The two options at the top determine if this filter will act as an *inclusion list* or an *exclusion list*. If you select **Encrypt only the following**, only traffic matching the rules will be encrypted and everything else will be sent unencrypted as if the iPig VPN were not there. Alternatively, you can configure it to encrypt all traffic *except* those that match the filter rules. The latter option is probably more desirable, because it will allow you to send all data over the VPN except any applications you specify in the filters. The filters allow you to specify the name or IP address to match, as well as port number and protocol (UDP, TCP, or both).

The iPig VPN solution is extremely easy to set up, offering a quick and painless way to provide VPN connectivity to remote users. The limit of five connections to the iPig server is very reasonable and should be able to accommodate a small office. If you decide that the product meets your needs and you want to upgrade to the full, unlimited user version, the registration cost of \$99 is very reasonable. Another offering in the “zero configuration” VPN space is LogMeIn Hamachi, available from www.hamachi.cc/. LogMeIn Hamachi’s structure for the free version limits you to 16 systems in the VPN network and the documentation says that the product’s “mediation” server is used to help clients find each other. It’s not entirely clear if the initial authentication passes through Hamachi-owned devices or not. As always, do your research. While iPig’s limitations seem easier to work with, Hamachi might be worth investigating to see if it suits your needs.

OpenSSL VPN

There are many commercial VPN solutions using SSL to provide encryption. SSL is the same encryption protocol that is used for secure Web pages (*HTTPS://*) and as such it is a very well tested and widely understood protocol. There are not very many offerings for free SSL based VPNs, but OpenVPN is a very robust and active package. You can read about it and download it from <http://openvpn.net>. OpenVPN can be installed on Linux, Windows (2000 or newer), several versions of BSD, MAX OS X, and Solaris. We will be using Windows 2000 for the server and Windows XP for the client, although the differences between using Windows and Linux should be minimal.

OpenVPN uses a single executable to serve as both the client and server components. Download the latest stable version from <http://openvpn.net/download.html>. You can download a *.ZIP* file or a Windows installation program (*.EXE*). The executable is the easiest to use, so that will be the installation method used in the examples. OpenVPN is natively a command-line program; however, there is a GUI available for download from <http://openvpn.se/>. Follow these steps to get OpenVPN installed and configured.

1. Start the installation program. Click **Next** on the welcome screen.
2. On the license agreement screen you must click **I Agree** to continue with the installation.

3. The next screen is the “Choose Components screen.” Leave all components checked and click **Next**.
4. On the next screen, select your installation directory and click **Install**. During the installation you will receive a notice that the TAP driver has not been Windows certified. Click **Yes** (Windows 2000) or **Continue Anyway** to continue with the installation.
5. After the installation completes, click **Next** and then click **Finish**.

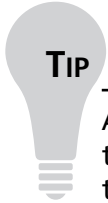
Configuring the OpenVPN Server

After installation, the next step is to edit the client configuration file and the server configuration file. You will first configure the server side. Pay strict attention when following the instructions as to whether you are working with the server configuration file or the client configuration file. Also note that your operating system has to be able to find any files it needs, so your configuration files either need to use the complete path for all file references, or you must have the appropriate directory in your system’s search path. Sample configuration files can be found in the `\OpenVPN\sample-config\` directory.

1. Copy the `server.ovpn` file to the `\OpenVPN\config` directory and rename it to something meaningful. For this example, we used `\config\LAB_SERVER.ovpn`.
2. Open the server configuration file for editing with an American Standard Code for Information Interchange (ASCII) editor, such as Notepad.
3. There is a directive named `port` that specifies the port the server will listen on for inbound connections. The default port for the server is 1194. If you want to change the port and use something non-standard, edit the port number here.
4. Scroll down to the line containing `;dev-node`. You need to remove the semicolon and enter the name of the virtual adapter that OpenVPN installed in place of the default “MyTap.” You can find this by going to a command prompt and entering `ipconfig /all`. One of the adapters will list “TAP-Win32 Adapter V8” as its description. The name of that adapter is often **Local Area Connection 2**, or something similar.

Edit the line in the configuration file with the adapter name such as **dev-node Local Area Connection 2**.

5. Save your changes and then close the configuration file.



As a general rule, I always rename the network interfaces on my systems to make it easier to administer. When you are troubleshooting a connection or program it is far easier to understand which interface is which with a more meaningful name than Local Area Connection 1, Local Area Connection 2, and so on. To do this, navigate to your Network Connection, right-click any of the connections, and select **Rename**. You can use `ipconfig` to see the description, which usually tells you which one is which. In the preceding example, I renamed the OpenVPN virtual adapter to OpenVPN, so my configuration file would read **dev-node OpenVPN**.

At this point you can choose which type of authentication you want to implement. OpenVPN can support a wide variety of methods, including public key infrastructure (PKI), user/password, and even two-part authentication with the proper plug-ins. For simplicity, we use a simple user and password. You will need to create a server certificate and Certificate of Authenticity (CA) certificate. This must be done as part of the SSL encryption, and is required whether you use PKI or user/password authentication.

6. Open a command prompt on the server and CD to the `\OpenVPN\easy-rsa\` directory.
7. Enter the **init-config** command and press **Enter**. This creates the `vars.bat` file and the `openssl.cnf` files.
8. Edit the `vars.bat` file using a text editor such as `notepad.exe` or `write.exe`.
9. Edit the `HOME` variable to match your directory structure if you installed OpenVPN to a non-default directory location.

10. Edit the *KEY_COUNTRY*, *KEY_PROVINCE*, *KEY_CITY*, *KEY_ORG*, and *KEY_EMAIL* variables with their appropriate values. These are used to generate the key file.
11. Enter the following commands at the command prompt.

```
vars
clean-all
build-ca
```

When you run *build-ca* it will prompt you for some localized information. It should read the defaults from what you entered in the *vars.bat* file for any required fields. Additional fields that were not in the *vars.bat* are optional. Pressing ENTER should accept each default value and fill in all of the required information. The only exception is the server's "common name," which you must enter. When the process completes there will be no special indication as to whether it was successful or not. You can verify the creation of the CA certificate by checking in the `\OpenVPN\easy-rsa\keys\` directory for a newly created *ca.crt* file.

- 12 Generate the required Diffie-Hellman parameters by running `\OpenVPN\easy-rsa\build-dh`.
- 13 Generate the server certificate by running `\OpenVPN\easy-rsa\build-key-server server`. As with *build-ca*, there will be a series of questions you must answer. The questions that must be answered should pull information in from the *vars.bat* file and use it as the defaults. Once again you will need to enter a common name in a series of prompts.
- 14 Add the following directives to the server configuration file (`\OpenVPN\config\LAB_SERVER.ovpn`).
 - ***client-cert-not-required*** Tells the server not to expect the clients to present their own certificate.
 - ***username-as-common-name*** Tells the server to use the username the client provides as the unique identifier for the client, rather than the common name found in the client's certificate.
 - ***Auth-user-pass-verify* <script> <via-file | via-env>** tells OpenVPN how to authenticate users. This directive is required

when not using certificates. The script is a file or program that authenticates the users. *Vie-file* or *via-env* tells OpenVPN to pass the username and password to the script as an environment variable or as a two-line file.

The default assumption for OpenVPN is that you will use PKI to authenticate all parties. If you want to use a user/password authentication mechanism, you are expected to configure OpenVPN so that it can pass the credentials out to a third-party process for verification. This modularity allows OpenVPN to support many different types of authentication. If you are running on Linux, there is a PAM module you can use. If you are running on Windows, there is no built-in way to verify the authentication. You can use any script you like. If the script produces an error code of zero, the authentication was successful; a one means it was not successful. This means there are many ways to verify the users, with only your imagination as a limitation.

To elaborate further, let's look at a simple example. Suppose all you want to do for verification is see if a directory is present on the VPN server with the same name as the user's name (not secure, but this is just an example of how the process works). You could create an "authentication" script called *C:\check.bat*. The line in the server configuration file would be *auth-user-pass-verify C:\check.bat via-env*. This will cause the OpenVPN server to call the batch file any time a user logs in. The batch file will have environment variables of *username* and *password*. The authentication script would only need the following line to check for the appropriate user directory:

```
IF EXIST C:\%username% EXIT
```

If the directory is present, an error level of zero would be returned to the OpenVPN Server and the user would be authenticated. For something more practical, we used a *psexec* utility, which is part of the *pstools* package. These are free tools available from Microsoft at www.microsoft.com/technet/sysinternals/Security/PsTools.msp. The purpose of the tool is to allow you to remotely execute commands on Windows hosts. The key component we use allows you to authenticate, so if you try and run the command using the credentials the VPN client supplied, it will tell you if they are a legitimate user or not, based on their local Windows account. The *check.bat* file assumes all relevant files and utilities are in the system's path. *Check.bat* contains the following lines:

```
IF "%1"=="test" exit
psexec -l -u %username% -p %password% C:\check.bat test
```

The first line is checking the command line for an argument of test. The first time the OpenVPN Server calls check.bat, this argument will not be present, so it will go to the next line. The next line will use psexec to run this same check.bat again. This will fail if the supplied credentials are not correct. If they are correct, check.bat (# 2) will be opened, this time with “test” as the first argument. When check.bat (# 2) sees “test” as the first argument, check.bat (# 2) will close. At this point, check.bat (# 1) has completed its assigned task and exits with an error code of zero, thus authenticating the VPN client. The psexec utility does allow you to specify the system to run the command on using `\\<computername>` format. When you omit the computername, as in this example, psexec will assume the account is local. In this case, the VPN server would need a local account to authenticate against.

1. Find the line with *Server 10.8.0.0 255.255.255.0*. This line tells the VPN server to give out addresses to the clients from that network range. You will need to edit this to provide a range of IP addresses that fits your network topology and is not in use or conflicting with your own internal DHCP servers. If your DHCP assigned addresses from 192.168.1.100–192.168.1.200, you could use 192.168.1.64–255.255.255.224, which would assign clients to addresses 192.68.1.66–192.168.1.94, with one of the available IP addresses going to the VPN Server itself.
2. If there are any non-local subnets that the VPN clients need to access, you must update the clients routing table accordingly. There are two ways to accomplish this, and both use the *push* directive. One is by sending them a specific route in the format *push route 192.168.111.0 255.255.255.0*. This will modify the client’s routing table and add a route to 192.168.111.0 with the VPN server as the next hop to reach that network. The other is using the *push “redirect-gateway”* directive. This will create a new default route in the client, so all traffic without a more specific route defined will go through the VPN server. The latter method is generally preferable as it is more secure. The increased security is because you will effectively disable

normal Internet access from the remote client via its own Internet link, thereby increasing the isolation between the corporate network and the rest of the Internet. The disadvantage is that when doing so, other non-work-related traffic will also traverse the corporate network, potentially consuming bandwidth.

3. When you are finished making your changes save them and close the configuration file.

Configuring the OpenVPN Client

The next step is to configure the client-side configuration file.

1. Copy the *client.ovpn* from the `\OpenVPN\sample-config\` directory to the `\OpenVPN\config\` directory and rename it. In this example it is named *LAB_CLIENT.ovpn*.
2. Open the new configuration file in notepad for editing.
3. Once again, edit the *dev-node* directive with the appropriate adapter name.
4. Edit the directive *remote my-server-1 1194*. Replace *my-server-1* with the hostname or IP address of the VPN server. The 1194 is the default port, which you can edit it if you want to use a different port number, which must match the port number you configured on the server “port” directive.
5. In the client configuration file, you can comment out the lines *cert client.crt* and *key client.key*.
6. Add the line *auth-user-pass* to the client configuration file. This instructs the client to prompt for a username and password.
7. You need to copy the *ca.crt* from the server to each client, and edit the directive to include the appropriate path. The server and all of the clients must have a copy of this file.
8. When you are finished making your changes save them and close the configuration file.

Once everything is configured, you can start the VPN server with the following command: **openvpn --config C:\openvpn\config**

lab_server.ovpn. Obviously, you will need to make sure the path and name of the configuration files match your environment. If you do not have the file directories in the path, you will need to place an explicit path into the configuration files, using a double backslash for directories. For example, you would use `C:\openvpn\easy-rsa\keys\ca.crt` for the CA certificate path. Start the client with `openvpn --config C:\openvpn\config\lab_client.ovpn` or the equivalent for your directory path and file names.

Using PKI Certificates of Authentication

With this much complete, using CAs instead of a user/password is easy. Follow these steps to change the authentication mechanism from user/password to CAs.

1. In the server configuration file, comment out the following lines:

```
client-cert-not-required
username-as-common-name
auth-user-pass-verify C:\\check.bat via-env.
```

2. On the server, generate a key pair for each client that will connect to the VPN. In the `\OpenVPN\easy-rsa` directory run `build-key client1`, where `client1` is the name you want to use for the first client to connect. This is often the same as the users logon ID. When prompted, the common name should be unique for each client; and again, the login ID might be a good choice for the common name. When prompted, it is recommended that you create a password for the client certificate. If you do not, anyone who obtains the certificate files can access the VPN.
3. Move or copy the `client1.key` and `client1.crt` to the appropriate client host. The server does not need a copy of these files, but generally you would leave a copy on the server as a backup.
4. In the client configuration file comment out the `auth-user-pass` line.
5. Edit the `.cert` and `.key` directives that were commented out previously for password authentication. Configure both of these lines with the appropriate paths to the `.cert` and `.key` files you copied from the server. Remember to use double backslashes for the path.

After the VPN connection is established, there are a few helpful shortcuts you can use from within the terminal window.

- **F1 Conditional Restart** This is similar to a warm reboot of the VPN tunnel. This will reset the tunnel, but will not reset the virtual Network Interface Card (NIC) (TAP adapter).
- **F2 Show Connection Statistics** This will give you some basic input and output statistics. Sample output is shown with the time and date removed from each line to conserve space.



```
OpenVPN STATISTICS
Updated, Sun Nov 26 10:20:20 2006
TUN/TAP read bytes, 240
TUN/TAP write bytes, 240
TCP/UDP read bytes, 4366
TCP/UDP write bytes, 4109
Auth read bytes, 272
pre-compress bytes, 0
post-compress bytes, 0
pre-decompress bytes, 0
post-decompress bytes, 0
TAP-WIN32 driver status, "State=AT?c Err=[(null)/0] #O=5 Tx=[4,0,0]
Rx=[4,0,0] IrpQ=[1,1,16] PktQ=[0,1,64]"
```

- **F3 Hard Restart** This reset will reset the virtual adapter.
- **F4 Exit** This will close the tunnel completely.

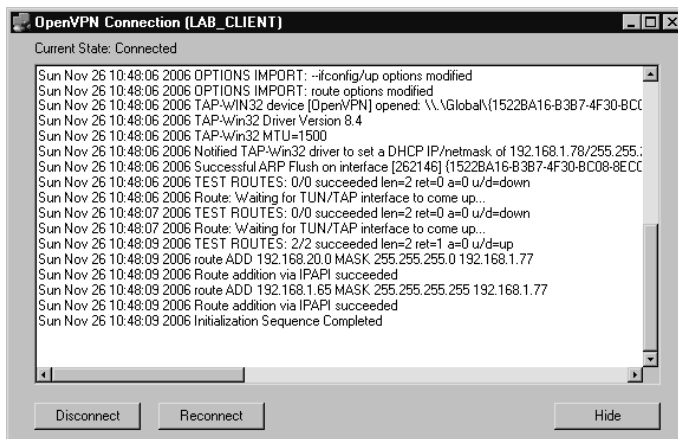
Configuring the OpenVPN GUI

Once the hard part is done (i.e., configuring the client-side and server-side OpenVPN configuration files and generating all the required keys), initiating the VPN tunnel is very simple. You could place the command line into a batch file or shortcut for your user's desktop. There is also a GUI interface available, though again, once you have everything running, the GUI may not be needed. This GUI interface can be downloaded from <http://openvpn.se/index.html>. The installation program is very straightforward, prompting for all of the normal parameters. The installer will install the base OpenVPN package in addition to the GUI components; therefore, you

don't need to install both. You will be prompted to confirm the installation, because the virtual NIC driver is not Microsoft certified. You should remove the previous OpenVPN program (via Add/Remove programs) before installing the GUI version to the same directory. The uninstall will not remove your certificates, keys, or configuration files, and the install will not overwrite them.

After installing the GUI version, the virtual network adapter name will be reinstalled with a standard name (e.g., Local Network Adapter 2). If your configuration file refers to a more meaningful name, you will need to rename the adapter (again). There should be a new icon in the system tray. Right-click this and select **Connect**. If you have multiple configuration files present, you will be presented with a menu folder for each, and options to connect or edit each individually. You should see a progress window with the same messages as you saw in the console window. If it scrolls by too quickly and you want a second look, right-click the icon in the system tray and select **Show Status** (see Figure 2.37). After negotiation completes, a pop-up window will indicate a successful connection.

Figure 2.37 OpenVPN GUI Status Window



While OpenVPN is not a “zero configuration” VPN like iFig, it is very powerful and flexible. The capability to support such a wide variety of authentication methods is not offered by any other free VPN software at this time. What really sets OpenVPN apart is its enterprise-class options and management

features. As an example, add the line *management localhost 7505*. This directive will work in both the server and client configurations and will instruct OpenVPN to listen to the indicated port (7505 in this example) for management connections. This will allow you to remotely query the OpenVPN instance and execute some limited commands.

An additional feature that lends itself to corporate functionality is the *remote* (*remote <IP> <PORT>*) and *remote-random* directives on the OpenVPN client. While we used the *remote* directive to specify the OpenVPN server to connect to, you can define multiple servers in the configuration file and the client will attempt each one in turn. This will allow you to configure a backup VPN server for redundancy. The *remote-random* directive instructs the clients to randomly select from defined remotes. In this way, you can load balance across two or more OpenVPN servers in an active-active configuration.

There are many more features available for OpenVPN. If you are not discouraged by the configuration that is required, you will be hard pressed to find a more robust free VPN solution. A quickstart guide is available from the OpenVPN Web site at <http://openvpn.net/howto.html#quick>.

Providing a Remote Desktop

Some of the considerations for placing your remote desktop are similar to those of a VPN. The primary consideration (i.e., physical location for the desktop host) has very little flexibility. Because the remote users will be accessing the desktop virtually, the desktop needs access to all of the same things it normally has. Unless you have some systems you can dedicate to this task and place in a DMZ, this probably means the users will be coming in through the firewall and accessing their desktops that are sitting on the internal network. Because of this, you want to secure the connection as much as possible. This desktop will have a login prompt exposed to the Internet unless you take steps to prevent it. A personal firewall (covered in more detail in Chapter 3) can help mitigate this, as well as firewall rules on your Internet connection. Most home users will be using a dynamic IP address, so you will probably not be able to restrict the connection to the users' specific IP addresses. You can, however, restrict it to the block of IP addresses corresponding to the local ISP's dynamic range. It is better to only let your local

geographic area be able to initiate a connection to the remote desktop than to the entire world.

There are people who scan the Internet looking for systems that are listening on ports commonly used for remote access. This is the primary reason you may want to consider using a non-standard port for your remote desktop solution. In the case of terminal services, the port is configured in the registry, and Microsoft does not recommend changing it; however, it can be changed. For third-party products such as Virtual Network Computing (VNC), changing the listening port is typically much simpler and advisable. Using a non-standard port does not guarantee the system will not be discovered and attacked. In fact, you can bet it will be, just less often than with a standard port. For this reason you must require and enforce a policy requiring very strong passwords for accounts with remote access privileges. If you're going to be exposing your internal network to the outside world, you should also implement an IDS if possible (see Chapter 4).

Windows Terminal Services

Windows terminal services are a handy way to provide access to a remote desktop across a single TCP port using remote desktop protocol (RDP). The fact that terminal services only uses a single port (TCP 3389; however, you can configure the port to be any port), makes the firewall rules simple to configure. With Windows XP or Server 2003, the terminal services client and server will be installed by default. For Windows 2000, the terminal services component can be installed via the add/remove programs applet in the control panel. Another advantage of using the Windows terminal services is that the client portion can be installed on older systems, allowing them to use the newer hosts and software. The remote desktop client can be installed on Windows 95, 98, ME, and NT as well as all of the newer Windows operating systems.

To enable the remote desktop functionality on a Windows XP Professional or Windows 2003 system, follow these steps.

1. Right-click on the **My Computer** icon and select **Properties**
2. Select the **Remote** tab.
3. Check the box that says **Allow users to connect remotely to this computer**.

4. Click the **Select Remote Users** button (see Figure 2.38).

Figure 2.38 Enabling Windows Terminal Services



5. On the “Remote Desktop Users” screen, click **Add**
6. On the “Select Users” window, enter the name of any user accounts that should be able to connect remotely and click **OK**.
7. Click **OK** to close the “Remote Desktop Users” window.
8. Click **OK** to close the “System Properties” window.

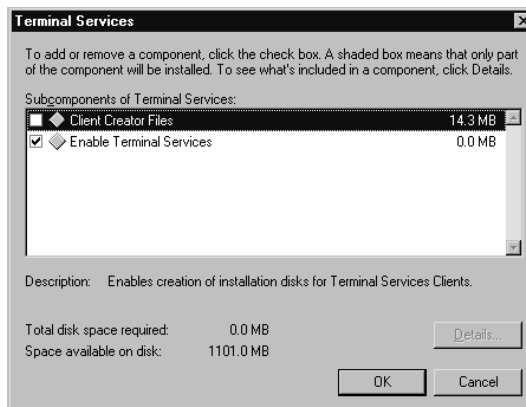
If you are using a client host that has the remote desktop client already installed, navigate to **Start | Accessories | Communications | Remote Desktop Connection**. The “Remote Desktop Connection” window is shown in Figure 2.39.

To connect to a remote desktop, enter the computer name or IP address in the “Computer” field and click **Connect**. If you click **Options** there are a wide variety of configurable parameters you can experiment with. Most of these settings are geared towards increasing the performance of the connection.

Figure 2.39 Remote Desktop Connection

If you want to use terminal services on a Windows 2000 computer, you need to install it following these steps.

1. Navigate to **Start | Settings | Control Panel**.
2. Select **Add/Remove Programs**.
3. Click the **Add/Remove Windows Components** button.
4. Highlight **Terminal Services** and click **Details**.
5. In the “Terminal Services” window place a checkmark next to **Enable Terminal Services** and click **OK** (see Figure 2.40).

Figure 2.40 Installing Terminal Services

6. In the Windows Components Wizard, click **Next**.
7. The Terminal Services Setup window will allow you to choose from **Remote Administration Mode** or **Application Server Mode**. Remote administration mode basically gives you a remote desktop, while application server mode allows you to share a single application. After selecting a mode, click **Next**.

8. When the wizard completes, click **Finish**.

The simplicity of providing a complete desktop for remote users to access can be appealing, and the ability to do so without any third-party software can be a major factor when deciding which solution to use. The flexibility doesn't end there, however. If you have a Windows server but need to access the server remotely from a Linux system, you can install `rdesktop`, which is an open-source client capable of speaking RDP natively. After installing `rdesktop` (via whatever means you choose; `yum install rdesktop` works on Fedora Core 5), all it takes to connect to a terminal server at 192.168.1.90 is to enter the **`rdesktop 192.168.1.90:3389`** command at a console prompt. To make things even easier, you can install terminal server client (`tsclient`), which provides a single GUI for establishing client connections to several common types of terminal servers. You still must install a client program such as `rdesktop`, because `tsclient` is only a front end, it does not include the client software itself. The only function of `tsclient` is to save you the trouble of needing to know the command-line options needed to make the connection. The `tsclient` interface is shown in Figure 2.41.

Figure 2.41 `tsclient`



Given the variety of RDP clients available, you should be able to connect to a Windows terminal server from most any OS. And with the convenience of having a GUI interface to make the connection, it is an easy solution. You should not expose any RDP services to the general Internet because then your only protection will be the password to connect, and automated brute-force password crackers will eventually gain access. If you do need to allow any type of remote desktop functionality to an Internet-based source, you should filter traffic to the respective port so that only the trusted IP address is allowed to connect.

VNC

VNC computing and has been around for a long time. It has gone through a lot of changes and is now available under many names, each with their own focus. This resembles the situation with Linux distributions in that the number of options can sometimes make it difficult to know which one is the best choice. Some of the more prominent are RealVNC, TightVNC, and UltraVNC. TightVNC (www.tightvnc.com/) encrypts the password exchange when you initially logon, but the rest of the session is unencrypted. While you could use an encrypted tunnel to encapsulate the VNC session (see Chapter 5, *Managing Event Logs*, for instructions on encrypting arbitrary sessions over both SSH and SSL), the other alternatives include native encryption support. RealVNC Personal Edition (www.realvnc.com/) includes support for AES128 encryption, but is only available for Windows platforms. This leaves my current top choice as UltraVNC from <http://ultravnc.sourceforge.net/>. In addition to providing encryption plug-ins, the UltraVNC server will run on any Windows system (Windows 95 thru 2003) and allows you to connect to it from any system with a compatible browser. To install UltraVNC follow these steps.

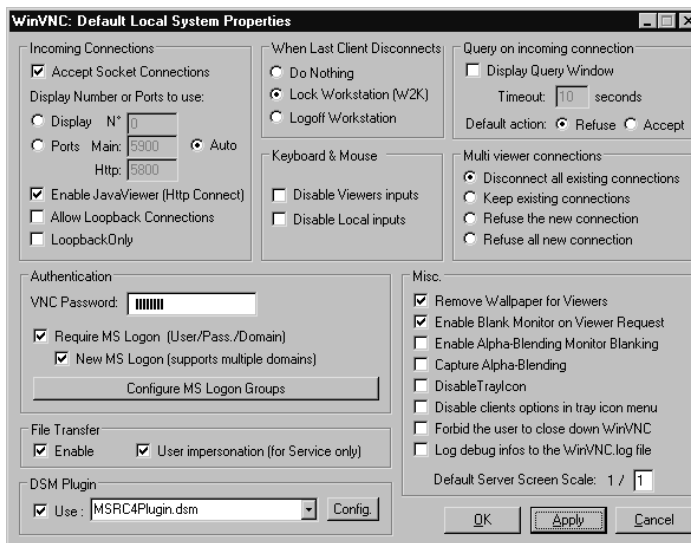
1. Download the UltraVNC setup file. The setup program includes both the client (the viewer) and the server component.
2. Run the setup program.
3. Choose the language and click **OK**.
4. At the welcome screen, click **Next**.
5. Accept the license agreement and click **Next**.

6. Click **Next** on the information screen (after reading all of the text).
7. Choose the installation directory and click **Next**.
8. Select the components to install (the DSM Encryption plug-in should be checked), and click **Next**.
9. Choose a start menu folder and click **Next**.
10. Next is the “Additional Tasks” screen, which has some additional options. The more noteworthy are outlined in this section.
 - **Register UltraVNC Server as System Service** This will cause the server component to be installed as a service. If you install this, the server will be started automatically. If you choose not to install the server as a service, the server will need to be started manually in order for someone to connect to the server. Requiring a manual start is more secure, but installing the server as a service might be a legitimate need in your environment.
 - **Start or Restart UltraVNC Service** This instructs the installation program to start the service during installation, or if it’s already started, to stop and restart the service.
 - **Configure MS-Logon II** This is a relatively new option that allows an ACL to be configured that specifies who can have access to connect to the server, based on the Microsoft account information. If you select this option but do not configure the ACL, only the administrator group will have access. This may be the preferred way to configure it if you only plan on using the VPN connection for support purposes. It is recommended that you enable this for increased security and simplified administration.
 - **Configure Admin Properties** This option prompts you for the location of a file registry file (*.reg*) containing the administrative settings for the UltraVNC server. These settings control things such as what to do with disconnected sessions and what to do upon disconnect. This can make configuration more automated by exporting the *HKEY_LOCAL_MACHINE\SOFTWARE\ORL* key for use during the installation of subsequent hosts.

- **Clean Old VNC Registry Keys** This option enables a house cleaning function to clear out old settings in the registry.
11. After making your selections, click **Next**.
 12. Accept the defaults for any needed configuration files in the next couple of screens (such as the MS-Logon ACL file, if those options were checked) and then click **Install**.
 13. Click **Next** on the information screen, and then click **Finish** on the next screen.

The server component should now be installed, but it still needs to be configured. Start up the UltraVNC server by navigating to **Start | Programs | UltraVNC | UltraVNC Server | Show Default Settings**. The properties page that opens is a little busy (see Figure 2.42).

Figure 2.42 UltraVNC Server

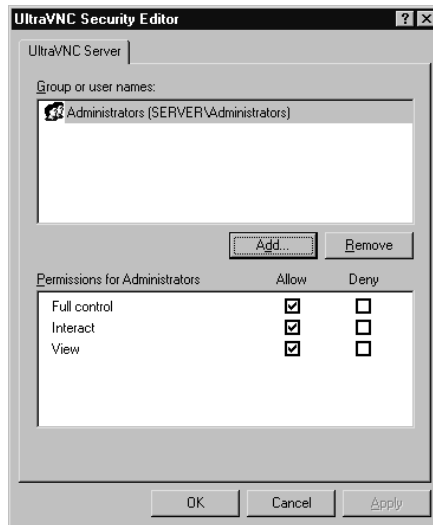


Most of these settings can be left at their defaults, but there are a few that are significant from a security perspective. The section labeled “When Last Client Disconnects” determines what happens after the viewer disconnects. Which setting is best will depend on your intended uses for UltraVNC. If you are going to use it primarily for technical support, there will probably be an authenticated user at the server when you disconnect, in which case you

probably want **Do Nothing**. If, on the other hand, you plan on offering the server as a remote productivity tool, there may be no one present at the server end, in which case you likely want to **Lock Workstation**, or even better, **Logoff Workstation**. The “VNC Password” in the “Authentication” section is a default password to be used with no other password supplied. This must be configured or VNC will not accept any inbound connections.

In order to take advantage of the Microsoft user and groups, place a check next to “Require MS Logon,” and preferably, “New MS Logon.” After doing so, click on **Configure MS Logon Groups** and configure what access different groups will have to the VNC server. The “Security Editor” windows provide a high degree of granularity. The editor is shown in Figure 2.43.

Figure 2.43 UltraVNC Security Editor



You can configure each group or user to have different levels of access, including view only, which can be a good choice for presentations or other applications where you only want the viewer to be able to see what’s occurring without the ability to interact with the server’s desktop.

Perhaps the most significant portion of the configuration is the encryption plug-in, in the “DSM Plug-in” section of the window.

1. Place a checkmark next to **Use:** and select the appropriate plug-in from the drop-down box. More plug-ins are available from

<http://msrc4plugin.home.comcast.net/index.html>. There are currently three different encryption plug-ins available:

- **MSRC4Plugin** This plug-in provides RC4 128-bit encryption. There are two versions provided with the UltraVNC installation: *MSRC4Plugin.dsm* and *MSRC4Plugin_NoReg.dsm*. You should use the *MSRC4Plugin_NoReg.dsm* plug-in.
- **ARC4Plugin** This plug-in provides RC4 128-bit encryption and handles the password slightly differently than the preceding one.
- **AESV2Plugin** This plug-in provides AES 128-bit encryption. The plug-in will not be available for selection in the drop-down box while it resides in the `\UltraVNC\plugin` directory. You must copy the *MSRC4Plugin_NoReg.dsm* plug-in to the main `\UltraVNC\` directory.

NOTE

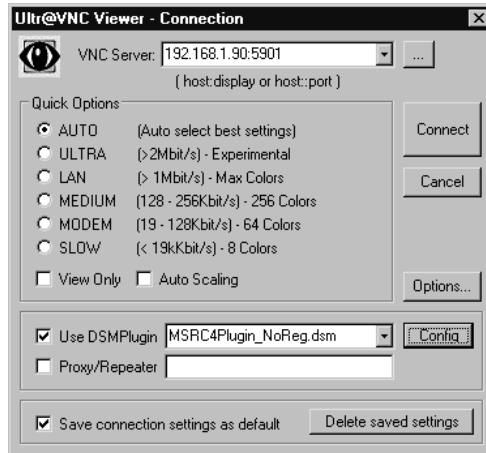
The plug-in that comes with the installation is an older version than the one available from the plug-in site. The version that came with the install did not work for me. Using the latest plug-in resolved the issue. Also note that you need to use the same key on both the server and viewer, a fact that is not clearly spelled out in the documentation. It doesn't matter which system you generate it on, although the server probably makes more sense.

2. Select the *MSRC4Plugin_NoReg.dsm* in the dropdown box and click **Config**.
3. Click on **Gen Key** to create a server key. When finished, click **OK** to close the plug-in configuration window.
4. Click on **Apply** and then **OK** to close the Default properties window.

The VNC Server should now be started and running securely. The next step is starting up the UltraVNC viewer (the client), which is much easier to do.

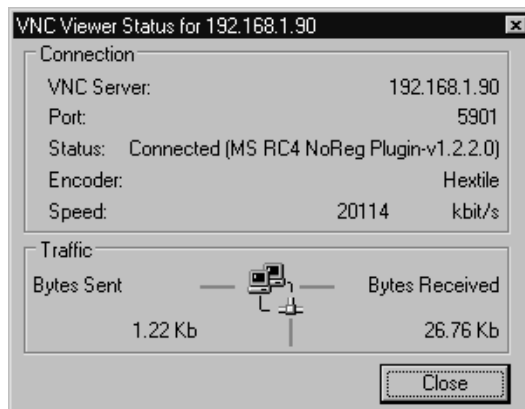
1. Start the UltraVNC Viewer via the desktop icon (if applicable) or by navigating to **Start | Programs | UltraVNC | UltraVNC Viewer**. This will open the connection window (see Figure 2.44).

Figure 2.44 UltraVNC Viewer



2. Enter the host name or IP address in the VNC Server field. Follow this with two colons (::) and port number, or a single colon (:) and the display number. The Default port will be 5901.
3. Place a check in the box next to **Use DMSPlug-in**.
4. Select the same exact plug-in in the drop-down box that you are using on the server. The plug-in files need to be the same version as well. Your key file should be the same one used on the server. The key file is not as “top secret” as you might think. Even if someone obtains the key file, they will still have to provide a username and password, and will be subject to the Windows account access you configured on the server.

Once you are connected, the status window will indicate that you are using encryption (see Figure 2.45). You can open the status window for an established connection by using the “Show Status Window” button at the top of the viewer window (it has a large exclamation point on the button).

Figure 2.45 UltraVNC Viewer Status

The status line will show the plug-in you are using (if any), thus indicating that you are using the MS RC4 plug-in, which in turn is a simple way to verify that you are successfully using encryption for your sessions.

Between using the native Windows Terminal Services and UltraVNC, you have a variety of options to provide you remote desktop access. Although Windows Terminal Services (also referred to by the Windows client/protocol name, Remote Desktop) is included and doesn't require additional software, a VNC solution is likely to offer more flexibility, and the server is supported on more platforms. VNC clients and servers are also available for virtually all Linux distributions. In fact, the default install of many includes one or both components. Fedora Core 5 installs the VNC server by default, while the client is an optional installation package. This widespread availability and support are one of the reasons VNC (in all its variants) is such a widely used solution.

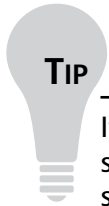
Using the X Window System

X window is the underlying management system for most UNIX and Linux GUIs. It takes an entirely different architectural approach than a Microsoft Windows system, in that the X Window system is set up in a client-server architecture from the beginning, similar to VNC. When reading the X Window documentation, you will find that X Window systems use the terms server and client in the reverse of what would seem intuitive, meaning the server is where the display is being generated, not the remote machine you are connecting to. Most current implementations are based on the work of the *X.Org foundation*

(<http://x.org>), which is the open source implementation of the X11 protocol. A closely related project is the XFree86 Project (www.xfree86.org), which is the open source version of the *X Window system* (which uses the X11 protocol). X11 is the protocol that is used to transfer information about the GUI between the server and the client. The end result of these design decisions is that much like Windows' built-in terminal server support, two Linux systems can remotely access each other via a GUI virtual desktop.

You can configure the X Window System to permit connections from remote systems without any third-party software. While this works, the evolution of desktop Window Managers and common software packages has rendered this method inefficient. A much more robust way to accomplish the same thing is using NX technology developed by *NoMachine*, which is a highly optimized process and protocol to make X sessions available remotely. NX is available for free (client and server) from www.nomachine.com/download.php. Commercial variations are also available. You can see the differences between versions, and thus see what the limitations of the free version are at www.nomachine.com/features.php. The big limitation is that the *NX Free Edition* limits you to only two concurrent connections. In most cases this won't be much of a limitation unless you are trying to use it as a full-blown terminal server solution rather than just a remote access mechanism. An open-source version of the NX server is called FreeNX and is available from <http://freenx.berlios.de/>. FreeNX does not support relaying sounds to the client (the NoMachine server does). This is the recommended server and the one used in the following examples.

To set up the FreeNX server, download and install FreeNX using whatever method is appropriate for your Linux distribution. For Fedora Core 5, I used `yum install freenx`. Yum (the package installer for Fedora Core 5) will automatically check for and install any dependencies. The aforementioned command also installed the core `nx` package, and `expect` as dependencies. In the case of Fedora Core 5, there is no need for any further installation. Depending on the distribution you are using, the installation may be more involved. Most of the major distributions should have packages available that make the installation relatively painless. You will need to have SSHD listening on port 22 in order for NX to work properly.

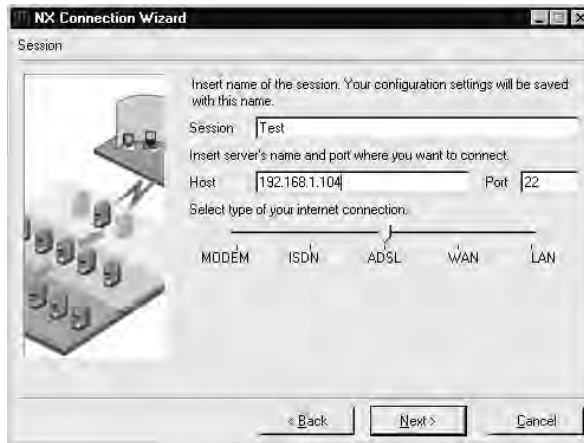


If you check the running services in an attempt to verify that the FreeNX server was installed successfully (via *chkconfig* or the like), you will not see it listed. This is because the server FreeNX server does not sit and listen on a port for an inbound connection like most services. Instead, when you login via SSH (this is why you need *SSHD* running on port 22), it logs you in as a special user (NX). That user's profile is configured such that it executes a process to start up the server and listen for the inbound connection.

With the server configured, the next step is to download and configure the client to make a connection. While there are alternate clients available, including some command-line clients, we will go with the original NoMachine client, which is installed on Windows XP for this exercise. Download the client and follow these steps to make a connection to the Linux FreeNX server:

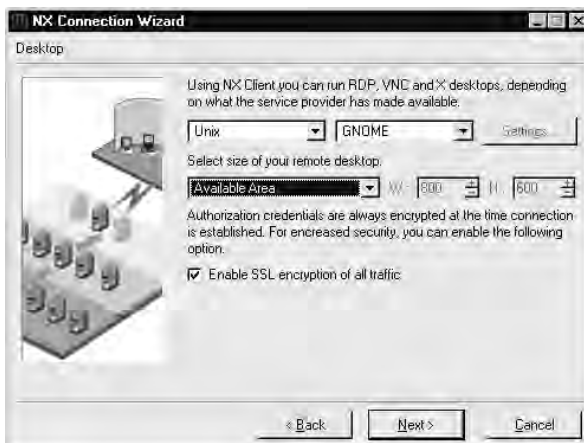
1. Run the installation program. The installation is unremarkable, asking for all the standards prompts such as a license agreement, and choosing an installation directory, the option to create a desktop icon, and so on.
2. When you first run the NX Client for Windows shortcut, it will launch the NX Connection Wizard, because you have no sessions defined. The wizard will walk you through establishing a connection. On the first screen of the wizard click **Next**.
3. The next screen allows you to configure a name for the session. All of the connection settings will be saved under this name for future use. This window also asks for the host and port. Unless you have changed it, leave the port at the default of 22, and configure the appropriate host name or IP address in the host field. This screen is shown in Figure 2.46.

Figure 2.46 NX Client Wizard Session



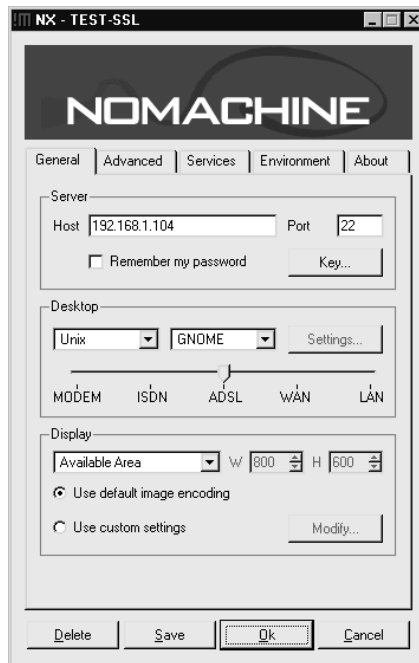
4. After entering the required information, click **Next**.
5. The Desktop setting window is next. Select the OS you will be connecting to, along with the window manager. In this case it was **UNIX** and **GNOME**. This window also gives you the option of enabling SSL encryption of all traffic. Unless you have a reason not to have additional security, you should enable this option. If you do not enable encryption, all of the X11 data will be sent unencrypted, meaning that someone with a sniffer could capture and reconstruct everything sent between the client and the server. This screen is shown in Figure 2.47.

Figure 2.47 NX Client Wizard Desktop

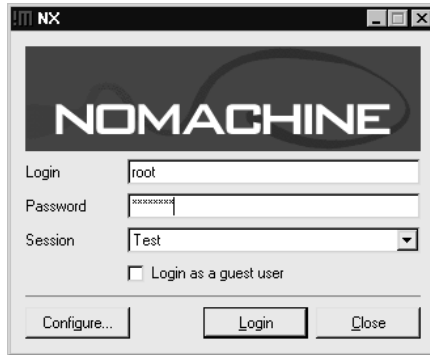


6. When you are satisfied with your settings, click **Next**.
7. The next screen of the wizard is the final one. The options are to **create shortcut on desktop**, and/or **show the advanced configuration dialog**. In order to enable SSL encryption for the connection, you will need to select the **show the advanced configuration dialog** checkbox and then click **Finish**.
8. The advanced configuration dialog is shown in Figure 2.48.

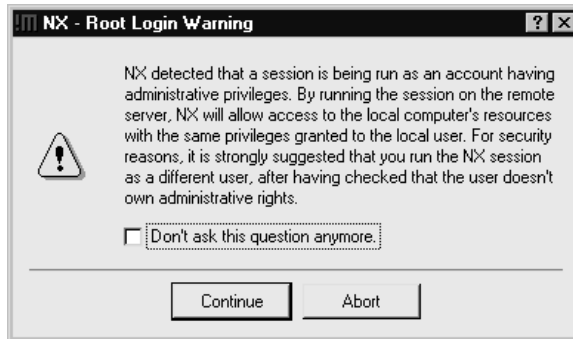
Figure 2.48 NX Client Advanced Dialog



9. On the “General” tab in the “Server” section, click on **Key**. You must have a copy of the *client.id_dsa.key* file, which will be found on the server in either */etc/nxserver/* or */var/lib/nxserver/home/.ssh/*.
10. On the Key Management screen, click **Import**, and select the *client.id_dsa.key* file and then click **OK** followed by **Save**.
11. Finally, click **OK** to commit your settings for this session.
12. After completing the advanced configuration, you will see the client login window shown in Figure 2.49. Enter the appropriate login name and password, and click **Login**.

Figure 2.49 NC Client Login

If you login as a user with administrative privileges, you will be given a warning (see Figure 2.50) that this is not best practices and that it would be more secure to use a non-administrator account. If you get this warning, you can click **Continue** to proceed with the connection anyway.

Figure 2.50 NC Client Login Admin Warning

You will be prompted to verify the key fingerprint of the host you are connecting to, which can be done by clicking **Yes**. After a period of encryption and authentication negotiation, you should see the remote desktop. You will quickly notice that the desktop that is spawned is a “new” fresh desktop just for this login, not a view of an existing logged-in user’s desktop. This default behavior is different than both Terminal Services and VNC, which both connect you to an existing session if one is present. While NX doesn’t behave this way natively, there are some workarounds to try and modify this behavior. Hopefully, future releases will simplify this option and make it easier to configure as the current workarounds are not elegant.

If you are in full screen mode such that you cannot see your real local desktop, there are some keyboard shortcuts available to accomplish some commonly needed tasks. These are outlined here.

- **CTR+ALT+F** Toggles full-screen mode.
- **CTRL+ALT+T** Shows the terminate/suspend dialog. If you terminate, the session is closed, while if you select suspend, you will be able to open the same session next time.
- **CTRL+ALT+M** Maximizes or minimizes the NX client window
- **CTRL+ALT+Mouse** Drags the desktop viewing area, so you can view different portions of the desktop. This is useful if the remote desktop has a higher resolution than your local resolution.
- **CTRL+ALT+Arrows** **CTRL+ALT+Keypad** Will move the viewport by an incremental amount of pixels.
- **CTRL+ALT+S** It will activate “screen-scraping” mode, so all the GetImage originated by the clients will be forwarded to the real display. This will allow you to take a screenshot of the remote desktop, to your local clipboard. If you press the sequence again, nxagent will revert to the usual “fast” mode.
- **CTRL+ALT+E** Enables lazy image encoding for improved speed.

Providing a Remote Shell

Sometimes you need remote access to a system but all you need is command line access. Maybe you could use more, as in GUI access, but connecting via the command line might be faster if you just need to connect and check something quickly. The primary use where a command line only connection is most applicable is when you are doing scripted changes. For example, if you needed to connect to 3 dozen systems and change a value in a configuration file, this would actually take a significant amount of time to do manually via the GUI, and a batch file would not be able to handle the GUI interfaces anyway. In these instances what would be ideal is a secure command line only remote access method. I mention secure expressly because a simple Telnet connection, while fast and efficient, includes no encryption, and all data

during your session (including your login username and password) will be sent in clear text across the network. For this reason you need a form of access that includes encryption such as Secure Shell.

Using Secure Shell

SSH (Secure Shell) requires both an SSH client and an SSH server component. SSH is the industry standard for remote command line access and most systems come with it as part of the default install. Windows systems of course are one of the few that do not. There are a variety of products available to bring SSH functionality to Windows, both commercial and free. One of the better known commercial SSH clients is SecureCRT (www.vandyke.com). Most of the free versions are based on the OpenSSH (www.openssh.com) package. There is also a GUI front end for OpenSSH, called PuTTY. Cygwin (www.cygwin.com) is a port of many UNIX tools for Windows and included in this package is an SSH server. To add even more options, SSHWindows is a free package that installs *only* the minimum components of the Cygwin package to use SSH, SCP, and SFTP. We will walk through setting up SSHWindows (on a Windows XP system). This package includes both the SSH client files and the SSH server files.

1. Download SSHWindows from <http://sshwindows.sourceforge.net/> on the client and server
2. Unzip the file and run the setup utility. Answer the standard prompts and then click **Finish**

At this point the SSH client is ready to be used without the need for any additional configuration. Before you can use the SSH server, however, you *must* create and edit the `\OpenSSH\etc\passwd` and `\group` files.

1. If desired create a separate group on the system to hold users who will have access to SSH, and add the local user accounts to the group for anyone you wish to have access to connect to the SSH server.
2. At the console navigate to the directory where you installed `\OpenSSH\bin\`
3. Enter the following command on the server to specify which groups can connect via SSH: `mkgroup -l >> ..\etc\group`. This will give

all local (-l) groups permission to connect via SSH. You should open the group file and edit out the lines corresponding to any groups you do not wish to have access.

4. Enter the **mkpasswd -l -u <accountname> >>**

..\etc\passwd command on the server to specify a single account that is authorized to connect via SSH. You must perform both steps 3 and 4 for SSH to work. If you do not specify the **-u <accountname>** all local users will be added to the passwd file.

5. Edit the *Banner.txt* file located in `\etc\` to match the banner specified by your IP security policy.

Once this is completed you can start and use the SSH server via the Services applet of the MMC or by entering **net start "openssh server"** at the command prompt. Here is an example of output from a successful SSH connection.



```
I:\OpenSSH\bin>ssh sshuser@192.168.1.101
***** WARNING BANNER HERE *****
sshuser@192.168.1.101's password:
Last login: Sat Jun 24 20:05:22 2006 from 192.168.1.99
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\OpenSSH>ipconfig

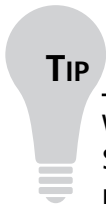
Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : rr.com
    IP Address. . . . . : 192.168.1.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

This is the sample output from sending a file to the bastion host
(192.168.1.101) via SCP.
I:\Internet\OpenSSH\bin>scp sample.txt sshuser@192.168.1.101:/
***** WARNING BANNER HERE *****
```

```
sshuser@192.168.1.101's password:
Could not chdir to home directory /home/SSHuser: No such file or directory
sample.txt                               100% 1735      1.7KB/s   00:00
```

**TIP**

While the SSH port in SSHWindows uses standard CMD.exe syntax, the SCP command and SFTP command both use Unix style paths. Also of note is that unless it is configured differently, the SSH connection will assume that the directory you installed OpenSSH into is the starting root for client connections.

If you get an error of *segid: Invalid Argument*, this typically means that the permissions are incorrect in the passwd file. The logon account on Windows systems should be 544 instead of 514. The latest installation didn't seem to have this issue but it's not uncommon.

Using a Secure Shell GUI Client

In virtually all cases with a command line utility offering many options and configuration parameters, some one will come along and create a GUI front end to make things simpler. This is true with SSH as well. By far the most widely used front-end is PuTTY, which is available from www.chiark.greenend.org.uk/~sgtatham/putty/. Setting it up is fast and simple because all you need to do is download the single PuTTY.exe file and it's "installed." When you run the EXE, the configuration window will look like the one shown in Figure 2.51.

While there are a lot of options, a basic SSH session is easy to configure. Make sure that **SSH** is selected as the protocol, and enter the host name or IP address you wish to connect to in the appropriate field. You can enter a session name and click **Save** if you wish to save your settings for future use. When you are happy with your settings, click **Open** to initiate the session. If it is the first time to connect to that particular host you will be prompted to verify the server key by clicking **Yes** as shown in Figure 2.52.

Figure 2.51 PuTTY Configuration

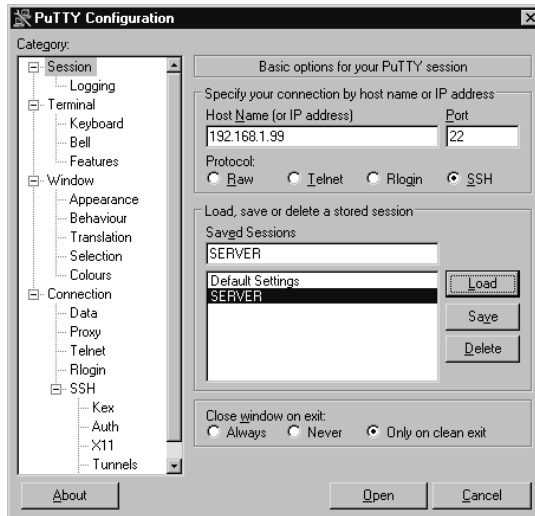


Figure 2.52 PuTTY Server Key Verification



A console window will open and the configured logon banner will be displayed. You should then have access to the command prompt on the remote system. To restore the session settings, simply highlight the session name and click **Load**, followed by **Open** to connect.

There is a wide variety of options for providing free, secure remote access, probably more options than most people realize. Each option has its strengths and weaknesses, and in reality, what you already have in-house may dictate which server you use. Hopefully, the coverage of remote access clients and servers in this chapter, will allow you to make the most out of the resources you have available. Table 2.1 summarizes the capabilities of the various remote access methods.

Table 2.1 Remote Access Methods Feature Summary

	Remote Access Feature Matrix					
	Terminal Services Client	Terminal Services Server	X Window Client *	X Window Server *	VNC Client / Server	OpenSSH
Command Line Support	✓	✓	✓	✓	✓	✓
GUI Interface Support	✓	✓	✓	✓	✓	-
Runs on Linux	✓	-	✓	✓	✓	✓
Runs on Windows 9x/ME	✓	-	✓	-	✓	✓
Runs on Windows 2000	✓	✓	✓	-	✓	✓
Runs on Windows XP/2003	✓	✓	✓	-	✓	✓

* I am using the more common definitions of "client" and "server", which is different than the terminology used by the X Windows System. The Client is the desk you are sitting at, the Server is the system you are connecting to, and the system whos resources you are using.

Summary

In this chapter, we examined a multitude of methods to secure your network perimeter and provide you, as the administrator, the access that is needed to administer the network. The Linux built-in firewall netfilter was covered extensively due to its power and flexibility, not to mention availability, as a free stateful firewall. In addition to *iptables*, we looked at several GUI front ends that allow you to manage the netfilter firewall without knowing the *iptables* command-line syntax. With your perimeter secured, the next step was to establish a secured doorway, so that you could sit at home and take care of the network. With command-line access via SSH, and Windows Terminal Services offering a remote desktop, FreeNX rounded out the offering by offering multiple remote desktop sessions from the same server.

Armed with this knowledge, you have no excuse to not have some type of firewall for protection on any and all unsecured connections. We say unsecured, not Internet intentionally, because any business partner, home user network, or the Internet are all considered untrusted, meaning you have no or incomplete administrative control over the security of the network you are connected to. Ultimately, you have no way to guarantee or enforce the proper security controls of an untrusted network. The sad fact is, if you have an Internet connection and don't have any type of firewall between a computer and the Internet, odds are very high that you have already been compromised. For other types of untrusted connections your odds may be better, but you're still gambling if you don't take steps to protect your network and systems.

Solutions Fast Track

Firewall Types

- ☑ In the networking sense, a firewall is basically any component (software or hardware) that restricts the flow of network traffic.
- ☑ Some firewalls are notoriously limited in capability, and others are extremely easy to use.
- ☑ To permit or deny traffic based on which network device is the sender or recipient and what ports are being used, you would use a packet-filtering firewall.

Firewall Architectures

- ☑ The most securely configured firewall in existence will not provide much protection if the underlying network was not designed properly.
- ☑ A *screened subnet* is the simplest and most common firewall implementation. Most small businesses and homes use this type of firewall.
- ☑ The one-legged demilitarized zone (DMZ) still has the advantage of cost, because you are building a DMZ using only a single firewall.
- ☑ The true DMZ is generally considered the most secure of firewall architectures. With this design, there is an external and internal firewall. Between the two is sandwiched any Internet accessible devices.

Implementing Firewalls

- ☑ Netfilter is the built-in component that performs the firewall logic. iptables is the command-line interface used to configure the netfilter ACLs.
- ☑ Many GUIs exist with widely varying degrees of functionality and complexity. My suggestion here is choose the simplest one that will do what you need it to do. In all likelihood the “right” one will change for you over time.
- ☑ SmoothWall sits in a class of its own, due to the fact that it turns a PC into a dedicated firewall appliance that is completely configurable without ever logging into the underlying Linux operating system.

Providing Secure Remote Access

- ☑ Your remote access options will depend most heavily on the platforms you have available to use for the remote access server. In most cases, the client used for remote access will run on virtually any OS.
- ☑ In conjunction with considering the available resources, you need to evaluate what your remote access needs really are. Is command line good enough? Do you need a remote GUI? Do you need to tap into existing sessions for a tech support type of functionality?
- ☑ For any remote access solution, remember to make sure you are using encryption if you plan on the session traversing an untrusted network, such as the Internet.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: How do I make the NX Client connect to an existing session?

A: Currently, the most off used solution is to have the local user use an NX session (to localhost). That way, all of their activities are always within an NX session, making management (terminate, pause, resume) much simpler. Again, it’s not elegant but it works adequately.

Q: I like the features of a particular flavor of VNC (or any remote access method) but it doesn’t support encryption. What can I do to secure it?

A: One of the most common approaches is to use the SSH port-forwarding functionality. This is how many of the VNC variants are providing encryption behind the scenes anyway. While we discuss gaining administrative access via SSH in this chapter, Chapter 5 (Managing Event Logs) contains detailed instructions on using the SSH port forwarding feature.

Protecting Network Resources

Solutions in this chapter:

- Performing Basic Hardening
 - Hardening Windows Systems
 - Hardening Linux Systems
 - Hardening Infrastructure Devices
 - Patching Systems
 - Personal Firewalls
 - Providing Anti-virus and Anti-spyware Protection
 - Encrypting Sensitive Data
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Chapter 2 focused on protecting the perimeter of your network, which typically means the Internet link, but it could include any link to the outside world, including connections to business partners and affiliates. This chapter focuses on how to secure the network-connected resources, such as servers and workstations. Many times an organization looks only at securing its perimeter, while leaving its interior network wide open and unprotected. This hard-*exterior-soft-squishy-interior* approach is surely better than no security, but it is not the best approach. The best approach is through *defense in depth*, which is the practice of applying security measures at all levels of the network. A solid *defense-in-depth* approach includes defenses at the outer perimeter—typically, firewalls and an intrusion detection system. It also includes defenses within the interior of the network, such as internal firewalls, network segmentation, and port-level access controls. Finally, at the core of the security onion are the actual network resources. You can protect these resources in a variety of ways, including via personal firewalls, antivirus software, antispyware software, data encryption, and automated security policy enforcement. This chapter demonstrates how to accomplish all of these objectives and secure your network resources using only free tools.

Performing Basic Hardening

All general-purpose operating systems will, by their very nature, come with weaker security settings than you might like. This characteristic is unavoidable, largely because the devices are general-purpose. To accommodate the wide variety of uses the system might fulfill, some sacrifices must be made when it comes to securing the system. This isn't necessarily true when it comes to special-purpose systems, which often come with highly secured and specially tailored configurations so that the system can be used only for its intended purpose. You, on the other hand, know exactly what you want systems to do, so you can customize the general-purpose installation to be more secure in your environment. Securing a system is also referred to as *hardening* the system, which means to make it harder for an attacker to compromise the system.

Regardless of what purpose the system serves, there are some common hardening steps that you should apply to all systems. There are high-level tasks, and as such, the specific implementation details are going to vary from system to system. These high-level hardening tasks have been outlined here. Any plan you develop to harden any network resource should address all of these issues in some fashion.

Defining Policy

You cannot possibly harden a system, at least not from an auditing perspective, if you do not have a definition and set of criteria for what constitutes “hardened,” or secure. Because of this, any hardening process actually starts long before you ever configure anything on a device, with defining policies and standards. We touch upon some of the policy-related elements in Chapter 8. Your IT security team will clearly outline the objectives you are trying to meet. The related standards will provide measurable milestones to meet in pursuit of that objective. Your security standards are your yardstick for success and provide an objective measure of your progress. Although the words “make the server secure” imply certain objectives, they do not clearly define measurable tasks; therefore, proving that you have “made the server secure” would be difficult at best. As such, having some well-defined security standards will be to everyone’s advantage.

The IT policies and standards your organization employs should accurately reflect your organization’s specific needs. Many organizations will sell you an IT policy to use, or one to use as a shell for filling in your own specifics. Taking this approach of using a “canned” policy has many challenges. What constitutes an appropriate policy or standard for one organization may not adequately cover the needs of another organization. The business model, type of business, and a host of other factors all contribute to making a good policy fit your organization. You always will want your security policy and security standards to address some elements, as shown in the following list. Note that you need to represent each item in both policy *and* standards:

- **Classifying Data.** You must have some guidelines on the classification of data (public, confidential, secret, etc.) in order to define steps to secure the data. These classifications will drive subsequent stan-

dards, such as “confidential data must be secured on untrusted networks.” Also, such policy needs to define when data needs to be classified, and explain the different levels of data classification.

- **Information Confidentiality.** This policy and standard should define how to keep data private. This will include encryption requirements and methods. You also need to define the requirements concerning authorization and authentication. Any password requirements would fall into this category. This will also include procedures for granting and revoking access to data, and who has the authority to do so.
- **Information Availability.** This covers when and where redundancy mechanisms are required. This should spell out what levels of redundancy are needed under what circumstances, such as redundant hardware, redundant Internet circuits, RAID arrays, server clusters, failover hot sites, spare hardware, and UPS requirements.
- **Physical Security.** These will address requirements concerning granting access, revoking access, monitoring, and types of access. These could include when and where badge readers are required, when keys should and should not be used, office locks, access by service personnel, and so on.

Generally speaking, the larger the organization, the larger the IT security documents will be. In some cases, you may get as specific as to have separate documents for “Physical Security for IT Datacenters” and “Physical Security for Retail Storefronts.” In a very small organization, the entire IT security policy may be one document, and the entire set of IT security standards another document. Your IT policies are an instance of where “size doesn’t matter,” meaning that a large 100-page policy is not necessarily a “better” policy than a short, concise one. The key is that the policy fits your organization and addresses your needs.

Except in an emergency situation (where you have vulnerable devices in need of immediate hardening steps), defining the appropriate policies and standards should be the first step toward securing your network. You will not be able to create a proper IT security policy without understanding the business first. The creation of these policies cannot be done in a vacuum. I have

seen more than one policy or standard that was written without input from the appropriate groups. Although you could point to it and say, “yes, we have a policy defined,” the policy was constantly in need of revision, and there were innumerable exceptions. Remember, from an audit perspective, a high number of exceptions against a very granular policy are likely to look less favorable than very few exceptions against a more liberal policy.

Access Controls

Access to the devices will be one of the first issues to consider, regardless of what type of device it is. This will include hardening both the logical access and the physical access. When it comes to logical access, the simplest control to introduce is the use of firewalls, whether separate firewalls or a built-in “personal” firewalls on the host in question. Some systems have their own mechanisms for implementing logical access controls in addition to simply filtering network packets. Where possible, using these additional methods helps provide defense in depth and increase security. You should address physical access as well. The universal truth is that if you have physical access to a system, you can have full access to the system. This is because if you have physical control of a system, generally the system has mechanisms that allow you to gain complete access to the system.

For example, if you have physical access to a server, you can boot it up under Linux, edit the raw data on the hard disk, and reset the administrator password. You can use encryption mechanisms to encrypt the entire hard disk, which will render this particular attack ineffective. In this scenario encryption is serving as a type of access control as part of your defense-in-depth strategy. These types of requirements are exactly the ones you would need to spell out within your policy and then implement them to secure your systems and your data. Your hardening steps should address all of these concerns.

Authentication

Authentication means to prove your identity. In the most common form, you do this using passwords. Recall that there are several different means to authenticate a user, including something they *have*, *something they are*, *something they know*, or *any combination of the three*. In the case of a password, this would mean

using something the user knows (the password) to prove he is who he claims to be. When you hear the term *two-factor authentication*, this refers to using two out of the three mechanisms for proving someone's identity. Two-factor authentication most often takes the form of a token which randomly generates a key. This key (something you have), combined with a password or PIN (something you know), provides heightened authentication. One of your key goals of your hardening efforts is to strengthen the authentication process as much as possible. Your hardening steps will need to provide authentication as much as possible.

Authorization

Authorization means to define what you have access to do. Obviously, authorization cannot occur securely without authentication happening first. You cannot possibly know what Jill should be able to access until you have positively identified that the person in question is in fact Jill. Usually you control and harden authorization through tighter configuration of file-level security. It can also include access to systems—for instance, a restricted user who is not allowed to install software drivers or applications on his workstation. The objective is, of course, to provide as few privileges as possible, while still enabling the person to perform his assigned tasks. This concept is known as the *principle of least privilege*. It helps to determine the least amount of access a user requires in order to fulfill his assigned duties.

Auditing

Auditing is a part of the hardening process as well. A system with no audit trail is certainly less secure than one with an audit trail. While most of the security hardening you will perform to various network resources comprises preventive controls, an audit trail serves as a detective control. You should not only enable and configure the appropriate level of auditing but also take steps to protect the resultant audit logs. Remember to protect the logging process (by using a secured account to run it) and the logging configuration (typically using file-level access controls). You will need to safeguard the audit logs by implementing mechanisms to ensure log integrity and availability. If you cannot ensure the integrity of a log file, it may still be useful when it comes to troubleshooting a technology issue. The log file will be practically useless,

however, if you need it to reconstruct the actions of a hacker or for use during legal proceedings.

Hardening Windows Systems

Windows systems have a reputation for being insecure out of the box. This reputation is certainly less justified than it once was. Microsoft has made a lot of progress toward a very difficult goal, which is to make its operating systems inherently more secure without diminishing the user's experience. After all, a secure system that is unusable isn't going to be of much value. Much like minimum password requirements, there is a point at which your efforts to increase security will actually backfire and will reduce your overall security level. Because of this, you should always maintain an effort to balance increased security measures with overall system usability and functionality. Again, doing this properly will require an in-depth understanding of the needs and processes of the organization.

The first thing to do in terms of hardening a system is to assess the *current* security posture of the device in question. Only after you know how the system is configured can you determine what you need to do next. Chapter 6 provides some good guidance on how to assess the current security posture of your network, while this chapter will focus on the tools that you can use to configure security settings in an effort to harden your network resources.

General Hardening Steps

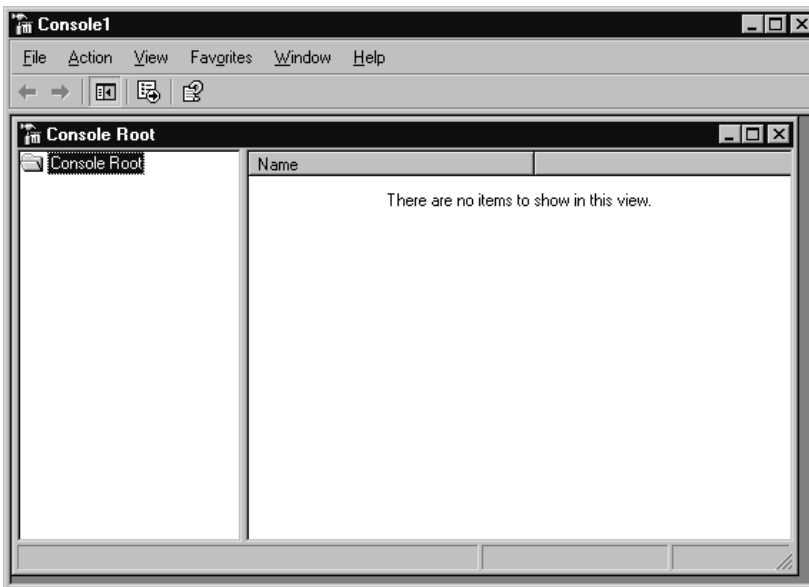
All the possible steps that you can take to secure a Windows host could fill (and have filled) entire volumes. The contents of this chapter should start you on your way with the basic hardening steps, and hopefully will point out a few tools you might not have known you had at your disposal. Be aware that in some cases, certain policies and standards may influence or affect other areas indirectly. For example, if you use biometric scanners for logon on all workstations, a "secure password" policy becomes practically moot. Under those circumstances, the password policy would end up applying only to special-purpose devices that were not able to take advantage of the biometric scanner. The exception, of course, is a requirement to use multifactor authentication, such as biometrics and a password. In this way, all of the hardening

steps are interrelated and interdependent. A policy decision in one area can impact a policy in another area, and we see a similar relationship in the use of standards.

Most configuration on modern Microsoft systems is performed using the Microsoft Management Console, or MMC. The MMC has come standard on Windows systems since Windows 2000. Because we'll be relying on it so heavily in the following sections, let's get familiar with using the MMC here. You can start by opening a new MMC console. Do this by navigating to **Start | Run | mmc.exe**.

Figure 3.1 shows the blank console.

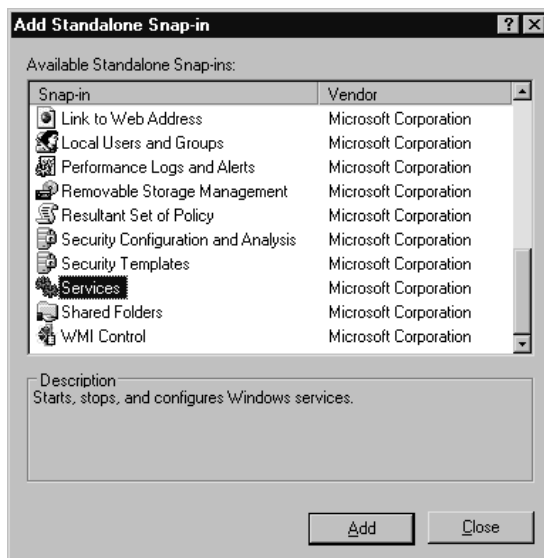
Figure 3.1 Empty MMC Console



The MMC is not very exciting without some additional configuration. The various interfaces are called *snap-ins*. The first time you open the MMC in this fashion it will have no snap-ins loaded. To load the services snap-in, you would open the console and click **File | Add/Remove Snap-in**. Then, in the Add/Remove Snap-in window, click the **Add** button. On the **Add Standalone Snap-in** window (shown in Figure 3.2), select the desired snap-in—in this case, **Services**, and click the **Add** button. Leave the default selection of **Local Computer** and click **Finish**. You can then close the **Add**

Standalone Snap-in window by clicking **Close**. Click **OK** on the **Add/Remove Snap-in** window. If you peruse the fields of the services snap-in you will find that they are the same as what you'd see if you had navigated to the stand-alone services snap-in via **Start | Run | Services.msc** or **Start | Programs | Administrative Tools | Services**. For one-off configuration tasks, the individual snap-ins might be easier to access, but an MMC console that includes all the snap-ins you need in one place is very efficient.

Figure 3.2 Add Standalone Snap-in Screen



When you close the MMC you will be asked whether you want to save your settings. If you click **Yes**, you can choose a name and location for the .msc file. In this way, you can have access to an MMC with your preferred snap-ins for future use. These saved configuration files are also portable, so you could place the file on a pen drive or network share and use it on another system. This can be handy for support personnel to have access to when working on other systems, especially since the start menu shortcuts might not be available on all systems. Now that you understand the basic operation of the MMC, let's move on to some actual hardening steps.

Users and Groups

One of the simplest ways to define access is by configuring the user and group accounts and the rights they have. Typically you can access the users and groups configuration via the **Control Panel** under **User Accounts**. You can also access the same information (in an easier-to-use format, at least in my opinion) within the MMC using the **Local Users and Groups** snap-in. If you try to use the **Local Users and Groups** snap-in on a domain controller, you will receive an error. You should instead use the **Active Directory Users and Computers** snap-in to manage your domain user and group accounts. There are a few common hardening steps you can perform with little effort, which are very effective from a security standpoint.

Most user and group administration will revolve around the Administrator account. On Windows, the Administrator account is a special account. Although the Administrator account is in the Administrator group, the uniqueness of the account goes far beyond that. Placing another user in the Administrator group will not give it the level of access and breadth of power that the Administrator account enjoys. Because of the privilege this account provides (similar to the root account on Linux), the Administrator account is the number-one target for attacks upon the Microsoft operating system. The name of this account is identical on every Microsoft Windows system, so everyone knows which account to attack. But we don't have to leave it that way. You cannot delete the Administrator account, but you can rename it to a more customized (and unknown) username.

A clever security technique is to rename the Administrator account. You can then create a new user account (with a username of *administrator*) with as little access as possible. This new Administrator account can have very thorough logging and auditing enabled for any actions this account performs. Using this method, all of your local technical support personnel will know about the renamed Administrator account and will not attempt to use it. Therefore, the only people attempting to use the Administrator account are likely to be up to no good, or at least such account activity would be highly suspicious. Renaming the Administrator account is as simple as right-clicking the account and selecting **Rename**. You might also want to remove the description for the newly named Administrator account. In some environ-

ments, the accounts are renamed to very innocuous-looking names, such as Printing or Lockout. In other environments, I have seen random-looking account names, such as XHOU923744. You can decide what strategy you want to employ. Just make sure the people who need to know can figure out the appropriate account to use.

Creating a new, fake “administrator” account is simple. Just right-click **Users** in the tree view (the left pane of the MMC) and select **New User**. Enter **Administrator** for the account name. You could even enter **Built-in account for administering the computer/domain** in the description field. You can change the password to random characters, as no one should ever really be logging on using the new Administrator account anyway. If you really want to make life difficult for an attacker, you can configure the login script for the fake Administrator account to execute some type of notification program. If anyone does actually manage to log in as the new administrator, it can send an e-mail to the security team. I have also seen organizations in which the legitimate Administrator account (the renamed one) did the same. They did this because tech support personnel should never be using the local Administrator account. Instead, they should be using their own personal accounts or a separate account that possesses administrative access to the local workstations.

If you wanted, you could take this concept a step further. You could require that any administrative accounts are to be used for administrative activities only and that they are *not* to be used for day-to-day login, even by the administrators. In this scenario, an administrator would have his normal account, which would have the same access as any other user account, and also have a unique account to use when doing administrative tasks. This unique account would have the elevated privileges that are required. Although this approach might be considered extreme, it does provide increased separation, as well as a high-quality audit trail of what high-privilege accounts are being used for. Whenever possible shared accounts should not be used (including the local administrator’s account) because, if something happens, having the audit trail tell you the local administrator account was used won’t really mean much when any number of support staff have access to the administrator password.

You can add a special script to the fake Administrator account by double-clicking the account name and selecting **Properties**. Click on the **Profile** tab and enter the path to an appropriate file in the **Login Script** box. I have seen instances where the true Administrator account was also “trapped.” When anyone logged in with either the “fake” or the real Administrator account, a batch file was run to collect a variety of system information, and then it was all e-mailed to the security team. A *net send* was also used to send an alert message to the entire security team. How elaborate you want to get when it comes to “trapping” the accounts is up to you.

Another simple step is to *disable* the guest account. In an environment where you have a domain controller you want all users to log in using their domain accounts anyway. This provides for much more meaningful auditing. The only common instances where a guest account serves a useful purpose would be when the environment is completely peer-to-peer and there are no domain controllers. Even then it would be preferable to have each person log in with a unique account, although doing so would require creating the account on all the workstations to which the person needed access. You can use *net user* to programmatically add a user account. By entering **net user newuser newpass /add** at a command prompt, you can add an account, called *newuser*, with a password of *newpass*. If you add **/domain** to the end and execute the command on a domain controller, it will add the account to the domain instead of adding it as a local account. If you use the same command without the */add* or */domain*, it will allow you to change the password for the account.

When it comes to account management, there is also an issue of education. It is desirable to use the actual Administrator account as little as possible. It is also desirable to perform day-to-day operations as a standard user instead of as a user with administrative access. This limits the possibility of a virus or other malware from being able to compromise the system. When the software in question attempts to modify the Registry or perform other actions to embed itself it would be met with inadequate privileges. Because spyware, malware, and viruses almost always require some type of elevated privileges to propagate, this mode of operation is much safer. In reality, most people know this is the recommended way to operate, but few people actually do it because of the inconvenience it can cause. Many programs are written poorly

and require access only an administrator has, and still others truly require administrative access by their nature. People can quickly grow tired of not being able to run the software they want to, so they tend to revert back to just logging in as an administrative user for their day-to-day activities.

Some tools, when combined with education, can help make it easier to follow best practices and limit the use of Administrator accounts. The Windows NT 4 Resource Kit included a utility called *su.exe*. This program allows you to execute a command as the super user, even though you are logged in as another user. More modern Windows operating systems accomplish the same functionality using the *runas.exe* tool. Here is the help output for *runas.exe*:



RUNAS USAGE:

```
RUNAS [/profile] [/env] [/netonly] /user:<UserName> program
```

```

/profile      if the user's profile needs to be loaded
/env         to use current environment instead of user's.
/netonly     use if the credentials specified are for remote access
only.
/user       <UserName> should be in form USER@DOMAIN or DOMAIN\USER
program     command line for EXE. See below for examples

```

Examples:

```

> runas /profile /user:mymachine\administrator cmd
> runas /profile /env /user:mydomain\admin "mmc %windir%\system32\dsa.msc"
> runas /env /user:user@domain.microsoft.com "notepad \"my file.txt\""

```

By using *runas*, a normal (nonprivileged) user could operate safely, but still have access to elevate his privileges quickly and conveniently in order to perform a specific task. The following command opens a command prompt as the user named *test*, on the *lab* machine:

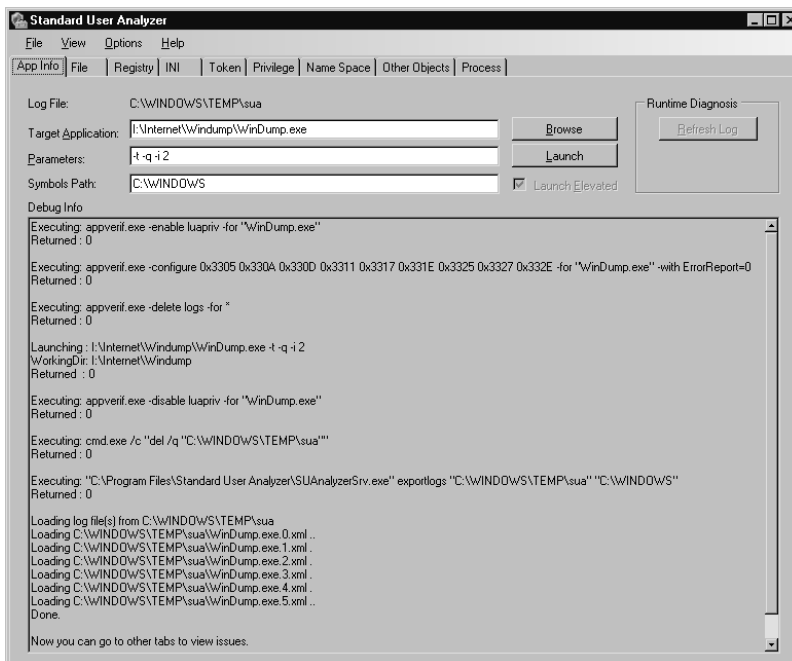
```
runas /user:lab\test cmd
```

After executing the command, you will of course be asked to provide the password for the account you specified. This doesn't keep the administrative staff from knowing the password for an elevated account, but at least they can use it only when needed. You can start most programs in this fashion. The

most common scenario where the *runas* command doesn't work properly is when multiple smaller executables must be started for a single application to function properly. You could even use *runas* to open the MMC with administrative access, to perform detailed administrative steps.

I have seen a normal user with administrative access on many occasions. The justification was that the user used some application which required administrative access. If you've ever tried to figure out what access the program truly needed in order to lock it down, this can be quite a chore. Microsoft provides a diagnostics tool to help you isolate the specific access that is required. Called the Microsoft Standard User Analyzer, the tool unfortunately will run only on Windows XP or newer machines. The program also needs the Microsoft Application Verifier (www.microsoft.com/technet/prodtechnol/windows/appcompatibility/appverifier.mspx), which is an additional download. Still, it can be a very powerful tool for locking down a standard user who happens to use a software package that "requires admin." You run the analyzer, specify the application to run, as well as any parameters the target application needs, and then click **Launch**. Figure 3.3 shows the main output screen.

Figure 3.3 Microsoft Standard User Analyzer



While the program is running, you will want to activate as many of its functions as possible. If you do not use a particular feature or menu function, the program may not fully exercise all the access rights it needs. When you are through testing the program, stop the program manually. This will cause the analyzer to complete its analysis and show the results. Each tab contains the elevated access that the program used. In the example shown in Figure 3.3, I was testing Windump, the Windows command-line sniffer. If you navigate to **View | Detailed Information**, the window will split into multiple panes showing more detailed fields explaining what type of access was requested. With the help of this tool, the odds are very good that you can have user access restricted to that of a standard user and simply grant elevated privileges where needed, rather than making the user account an Administrator account.

File-Level Access Controls

With all your accounts and groups organized and secured, the next step is to get more granular and look at file-level access controls. This is also the next logical step if you have used the Microsoft Standard User Analyzer and now need to modify some access within the file system. The standard way to access the file-level permission for a given directory or file is to right-click on the file or folder in Explorer and select **Properties**. Select the **Security** tab and you will be able to see the currently applied access for users and groups, as shown in Figure 3.4.

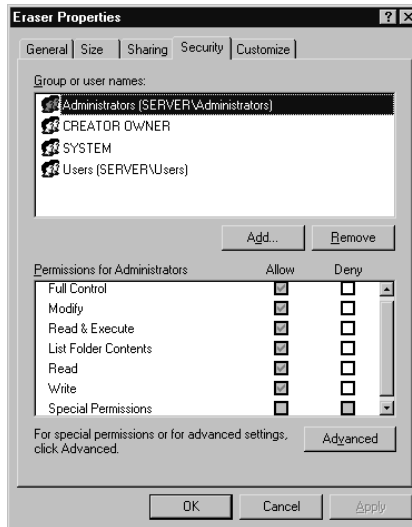
As you highlight each user or group in the top pane, the assigned rights will be displayed in the bottom pane.



WARNING

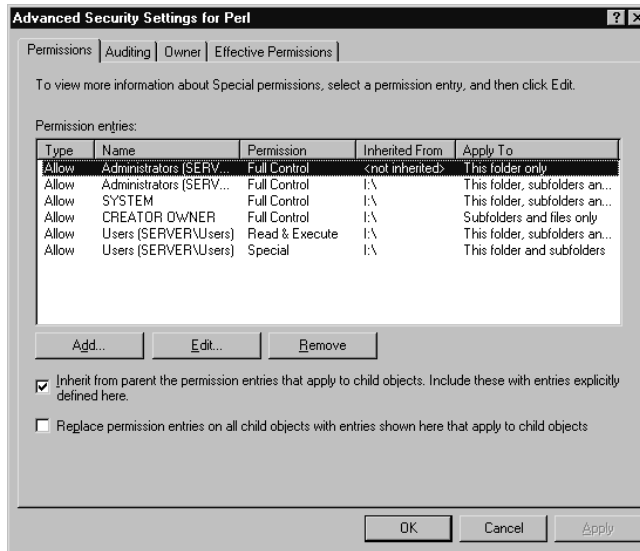
Avoid the common mistake of clicking **Deny** when you do not want a particular user to have access to a file or directory. To remove access, you should instead remove the check in the **Allow** column. If the user is not granted access via the Allow column—either directly or as an inherited permission from a folder higher up—the user will not have any access. The Deny selection overrides any granted access that may be present. This particular error occurs most often when an administrator wants to remove access to a directory for the Everyone group. By clicking Deny, no one will have access, including authorized users.

Figure 3.4 File-Level Security



By clicking the **Advanced** button, you gain access to several tabs with some very powerful functionality. One of those is the **Permissions** tab, shown in Figure 3.5.

Figure 3.5 Windows Advanced Security Settings

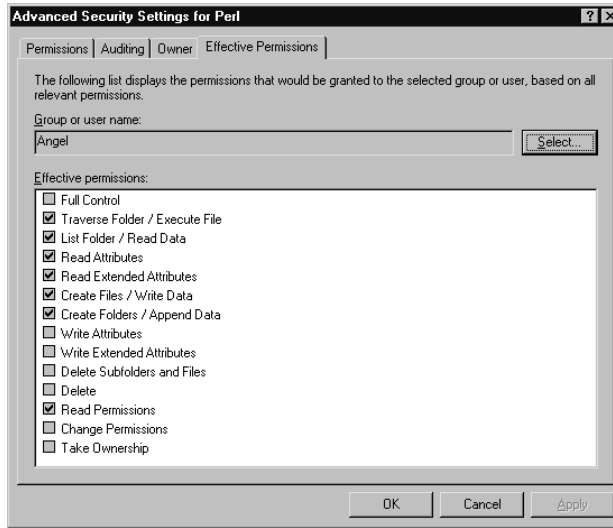


The **Permissions** tab allows you to control inheritance. With inheritance, if you grant access to a folder, it is assumed that you should have the same level of access to newly created files and folders underneath that folder. The new folders effectively inherit the same access control lists. The **Replace permission entries on all child objects with entries shown here that apply to child objects** checkbox allows you to do much the same thing with existing folders and files. The **Auditing** tab allows you to configure auditing on files or folders. Chapter 5 also covers enabling auditing on files and folders, so we won't discuss it here. The **Owner** tab can be very useful to an administrator. In cases where the administrator has been denied access to a file or folder (either directly or via membership in the **Everyone** group), the simple method to regain control over the file or folder is to go to a level above it, where the administrator *does* have access, and take ownership. You can do this by navigating to the **Owner** tab. The administrative user you are logged in as should already be entered in the **Name** field. Select the checkbox that says **Replace owner on subcontainers and objects** and click **OK**. As the new owner, you will always have access. You can then edit the permissions as needed to restore proper operation.

The **Effective Permissions** tab allows you to specify a particular group or account and generate a listing of what access users would have. This takes into account inheritance, group membership, and any explicitly defined permissions that may be present. To see the effective permissions for a user, click **Select**. On the next screen, you can enter the account name, or if you are unsure of the account name, you can click **Advanced**. This will take you to a third window that allows you to search for an account name. You can click the **Object types** button to limit the search to only users, groups, or built-in security principles, for example. The **Locations** button will allow you to choose on which system you want to search. After selecting your search criteria, you can click the **Find Now** button. With no search criteria defined, you will see a list of all accounts, groups, and built-in security principles for the local machine. Clicking one and then clicking **OK** will place the name in the **Select User or Group** window. You can then click **OK** to show the effective permissions. Figure 3.6 shows the **Effective Permissions** window. This is an effective way to know the result of all the various levels of filters and

access controls. This is also one of the first places to check when a program or user action generates an access denied error when you don't want it to.

Figure 3.6 Windows Effective Permissions



Sometimes you may be looking for a security issue and you want to look at all the directories, not specific directories or folders. The built-in security interfaces don't do a very good job of allowing you to do that. Some additional utilities do, however.

AccessEnum (www.microsoft.com/technet/sysinternals/Security/AccessEnum.msp) is a GUI utility that you can use to quickly identify permissions that deviate from the norm. AccessEnum will run on Windows NT, 2000, XP, and 2003. It will scan the folder you specify and all subfolders and files, and list only the ones that deviate from their parent. In this way, you can quickly see which ones have had permissions changed or modified in some fashion. AccessEnum can also scan a portion of the Registry, looking for similar permission deviations. This is a big plus because the provided interfaces for working with Registry permissions are very cumbersome. Unfortunately, AccessEnum abbreviates the access types to just Read, Write, and Deny, so it's probably not an auditor's best friend, but it can help you spot things that look out of place.

Another handy utility is AccessChk (www.microsoft.com/technet/sysinternals/Security/AccessChk.msp). This is a command-line program that will run on Windows 2000, XP, and 2003. AccessChk will also collapse the various permissions into a simplified read, write, and deny, but this program is still a very powerful tool for a security administrator. The following command will list the access for all files and folders at the root of the C: drive:



```
I:\Internet\Firefox>accesschk eric c:\
```

```
AccessChk v2.0 - Check account access of files, registry keys or services
Copyright (C) 2006 Mark Russinovich
Sysinternals - www.sysinternals.com
```

```
RW c:\i386\
RW c:\My Documents\
RW c:\MSDOS.SYS
RW c:\WINDOWS\
RW c:\IO.SYS
RW c:\Program Files\
RW c:\Windows Update Setup Files\
RW c:\boot.ini
RW c:\CONFIG.SYS
RW c:\AUTOEXEC.BAT
RW c:\ntldr
RW c:\ntdetect.com
RW c:\BOOTSECT.DOS
RW c:\Documents and Settings\
RW c:\System Volume Information\
RW c:\Recycled\
```

If you add the `-s` switch, it will recursively list all directories, and the `-v` switch will cause it to list all permissions that are present, instead of just R, W, and Deny. You could easily combine this utility with a little creative programming and create a batch file that could generate some very useful security reports, as shown here:



```
I:\Internet\Firefox>accesschk someuser I:\testfile.txt -v
```

```
AccessChk v2.0 - Check account access of files, registry keys or services
Copyright (C) 2006 Mark Russinovich
```

Sysinternals - www.sysinternals.com

```
R I:\TestFile.txt
    FILE_EXECUTE
    FILE_LIST_DIRECTORY
    FILE_READ_ATTRIBUTES
    FILE_READ_DATA
    FILE_READ_EA
    FILE_TRAVERSE
    SYNCHRONIZE
    READ_CONTROL
```

AccessChk will also scan the Registry, and you can specify a service instead of a username for the reporting. Several other switches not covered in this chapter add to the functionality and flexibility of AccessChk. Running AccessChk without any parameters will induce the program to output its usage text.

Additional Steps

There are, of course, additional steps that you can take to harden a Windows system besides tightening users and groups, and file-level permissions. As you might expect, Microsoft has the inside track on securing Windows systems. You can read some very extensive hardening guides on the Microsoft Web site. Here is a list of some of the more noteworthy security documents available:

- **Windows Server 2003 Security Guide:** www.microsoft.com/technet/security/prodtech/windowsserver2003/w2003hg/sgch00.mspx
- **Windows XP Security Guide:** www.microsoft.com/technet/security/prodtech/windowsxp/secwinxp/default.mspx
- **Microsoft Windows 2000 Security Hardening Guide:** www.microsoft.com/technet/security/prodtech/windows2000/win2khg/default.mspx
- **The Microsoft Windows NT 4.0 and Windows 98 Threat Mitigation Guide:** www.microsoft.com/technet/security/guidance/networksecurity/threatmi.mspx

Although the respective guides will go into great detail on hardening the operating system you are interested in, there are, of course, many more articles on the Internet and books on the subject. Essentially you can summarize all of the hardening steps in a few high-level tasks which are the same on any operating system, and most of them do not require additional software other than what is included with the operating system. The steps are as follows:

1. Remove unnecessary software.
2. Disable unneeded services/daemons.
3. Patch the operating system and any remaining software.
4. Configure user and group accounts to provide only the minimum required access.
5. Tighten operating system parameters (login requirements, timeouts, etc.).
6. Configure network access to permit only the minimum required connectivity (IP/ports).

Using Microsoft Group Policy Objects

If you find yourself wanting to lock down a particular security setting, such as removing user access to a directory, you could connect to each system and edit the security properties individually. While this would work, it's not a very efficient way of doing things. Of course, you can use the MMC to edit and control these settings individually, but you can also configure these settings and then have them applied to computers automatically. This collection of security settings is called a *group policy object (GPO)*. You can apply a GPO to the domain level, to individual organizational units (OUs), or to individual computers. Chapter 5 covers applying a GPO to an individual computer in detail, so we won't cover it here. Unless you have special security needs, such as for a high-risk host, you will configure most of your security policy to apply to all devices within the domain. If this is the case, the logical place to define your GPO is at the domain level. You could then address any high-security hosts by applying a more restrictive local policy (this policy should not include settings that conflict with the domain policy or else they will be

overwritten) or placing the high-security hosts in a particular OU and apply an OU GPO. Here is a step-by-step example of how this would work.

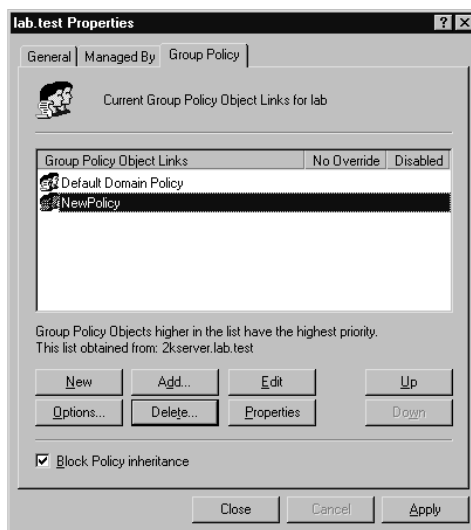
1. Local policies are applied. These could be modified or just left at the default; it doesn't really matter if you will be updating them with the domain GPO anyway.
2. The domain GPO is applied to every device on the network (workstations and servers). This overwrites the local policy with settings you want applied to everyone. In effect, this acts as your security baseline.
3. The GPO for the server OU is applied to the servers (or other high-security hosts). These devices will still have the minimum security settings from the domain GPO, but in cases where you wanted to define a more strict security setting, the OU GPO will allow you to do so.

The nature of GPOs is that you will configure settings with the intent of overriding other settings. GPOs are applied in the following order: local, the site GPO, the Domain GPO, and each OU GPO working downward. Typically you configure the domain settings to intentionally override the local computer security settings. Each successive application of the GPO overrides the previous one, which also has the desired side effect that the local policies are of lowest priority. You can view the *default domain policy* by opening your MMC console. Add the **Active Directory Users and Computers** snap-in. Right-click the domain in question and select **Properties**. Click the **Group Policy** tab. Initially the window will contain only the **Default Domain Policy**. This represents the top level of the domain, and you can configure multiple policies here. Again, all settings you configure in the domain GPO will override any local settings.

If the default domain policy does not have that particular setting configured at all, the previous policy settings will remain in effect. Because the default domain policy contains all of your default settings, you do not want to edit it directly. Instead, click on **New**, give the new policy a name, and then click **Edit**. This will place the newly created policy below the domain policy.

You can use the **Up** and **Down** buttons to shift positions of the highlighted policy. As you can see in Figure 3.7, the highest policy in the list will take priority (remember all of the GPOs in this list are domain GPOs). Because any newly created policies are intended to override the default domain policy, you probably want to click **Up** so that your policy takes precedence over the default policy. Figure 3.7 shows the **Group Policy** tab showing the default domain policy.

Figure 3.7 Domain GPOs



You can follow the same procedure to add a GPO to an OU (right-click **Properties | Group Policy**). In this fashion, you can assign very granular policies to all your resources. If you click **Edit**, you are presented with the same MMC console structure as that found within the **Local Computer Policy** snap-in. If you were to instead edit the Local Computer Policy settings, you would be configuring a local GPO, which would be overwritten in the event of conflicting settings with the domain GPO, or an OU GPO.

You may have noticed that within the Local Computer Policy snap-in there is no mechanism to export and import settings. This would imply that you have to configure any desired policies within Active Directory instead of doing so locally. Indeed, if you plan to apply the settings to all hosts, doing it within Active Directory may be a better way to do it, but there is actually a

way to configure the desired settings and then export them for use on another machine's local GPO. You do this through either the GUI or the command line. The GUI is, of course, the MMC. You will need to add the **Security Configuration and Analysis** and, optionally, the **Security Templates** snap-ins to your console.

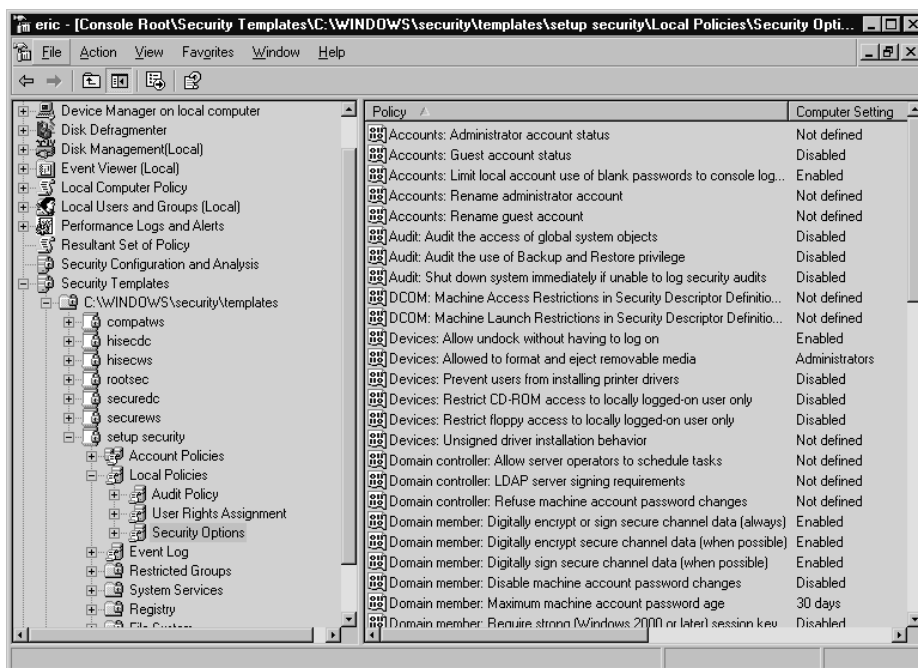
In order to import settings from another system, you have to have configured the desired settings at some point. To do this you use the Security Templates snap-in. A *template* is simply a preconfigured collection of settings. You can, of course, edit the templates to your taste, or create your own. The following is a brief summary of the templates included with Windows XP:

- **compatws.** Relaxes the default file and Registry permissions for the Users group.
- **hisecdc.** Provides further restrictions on LanManager authentication and further requirements for the encryption and signing of secure channel and SMB data above and beyond what is configured within the securedc template.
- **hisecws.** Provides further restrictions on LanManager authentication and further requirements for the encryption and signing of secure channel and SMB data above and beyond what is configured within the securews template.
- **rootsec.** Will reset the default permissions on the drive roots and propagate those permissions to child objects. Use with caution.
- **securedc.** A “secure domain controller” template that will configure tighter restrictions on domain account policies and additional restrictions on anonymous users.
- **securews.** A “secure workstation” template that will configure tighter restrictions on local account policies and additional restrictions on anonymous users.
- **setup security.** Holds the default settings and is usually a good place to start if you're not sure which template to use.

Figure 3.8 shows the Security Templates snap-in. If you want to create your own template, or make any changes to one of the default templates, you

should make a copy and work from that one so that you always have the original if you need it. You can copy the **Setup Security** template by right-clicking and selecting **Save As**. You can edit the settings in the template by drilling down using the Security Templates snap-in. After making changes, be sure to save the template. Because these templates are simply ASCII files, you can open a template to use the settings on different computers. If you can edit the security settings and import and export them using the Security Templates snap-in, you might be wondering why you would need the Security Configuration and Analysis snap-in. You need it because there is no way to *apply* the templates using the Security Templates snap-in.

Figure 3.8 Security Templates MMC Snap-in



In order to apply the settings in a security template, right-click on the **Security Configuration and Analysis snap-in** and select **Open Database**. Since this is the first time you're configuring the settings, you will be creating a new database. Enter a meaningful name ending in **.sdb** and click **Open**. You will be prompted to import a template. This is where you can import the custom template you configured previously. Once you do this, you

can apply your template by right-clicking the **Security Configuration and Analysis snap-in** again, and this time select **Import Template**. The various settings will populate within the snap-in. This provides you with yet another opportunity to edit the security settings. The settings are still not applied, however. The final step to commit the settings is to right-click again and select **Configure Computer Now**. All of the settings configured within the template will be applied. Depending on the number of nondefault settings this step could take a few minutes to complete.

Should the need arise, you also have the ability to create a template based on the current policy settings of the local system. You can do this using the Security Configuration and Analysis snap-in. Right-click the snap-in and select **Analyze Computer Now**. Choose a filename and path for the log file, or just accept the default and click **OK**. This will reveal several expandable items, all under Security Configuration and Analysis. These settings represent what is currently configured. To save these settings as a template, right-click and select **Export Template**. The most common reason to work from the current policy template and change it would be if you wanted to make incremental changes in a very controlled fashion. This would carry less risk of breaking something than applying any of the pregenerated templates if the system has undergone significant policy adjustments since it was originally installed.

Microsoft also provides a command-line utility called *secedit.exe* which you can use to import and export policy settings. In order to export a policy template using *secedit*, you must start by analyzing the current settings against a template. Using `secedit /analyze /db C:\export.sdb /cfg C:\test.inf` would instruct *secedit* to analyze the current local policy, using the `test.inf` template. The results will be stored in a database file called `C:\export.sdb`. To export the settings to an `.inf` file that you can import, you would use `secedit /analyze /db C:\export.sdb /cfg C:\test.inf`.

NOTE

If you try to export the setting and the file ends up being empty, you will be experiencing a known bug with *secedit*. The bug is caused because XP stores the security information in a different location than *secedit* is looking for it (*secedit* was originally developed for Windows

2000). A hotfix is available to fix the issue, and it's basically is just a newer version of `seced.exe`. See the Knowledgebase article ID 897327 at <http://support.microsoft.com/kb/897327> for more information on the bug with `secedit`.

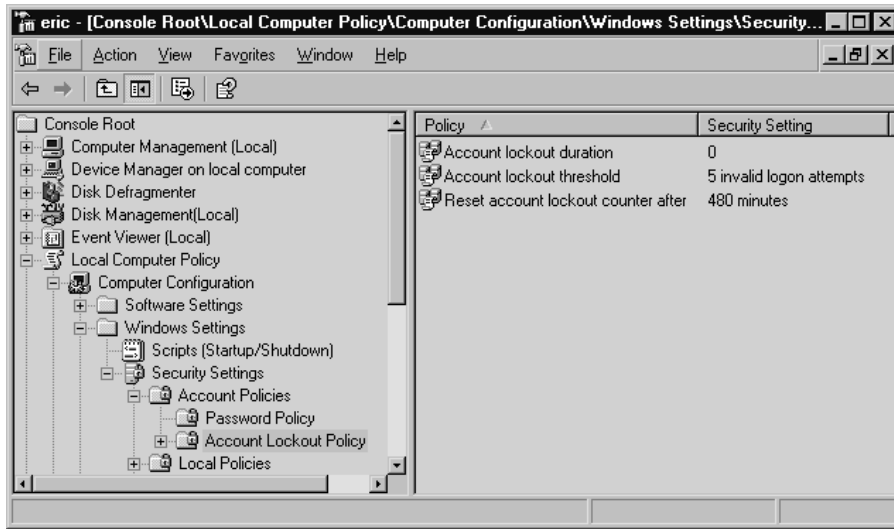
Account Lockout Policy

Now that you are familiar with GPOs and how to apply them, we will discuss a few policy settings that you may want to consider implementing, either at the domain level or with local GPOs. The account lockout policy (`\Computer Configuration\Windows Settings\Security Settings\Account Policy\Account Lockout Policy`) allows you to configure the number of incorrect passwords that a user can enter before being locked out of an account, how long the account stays locked out, and how long before the lockout counter will reset. The following recommended settings will provide the most security in an average environment:

- **Account Lockout Duration** represents how long the account will stay locked out. Setting this to zero means that the account will stay locked out until an administrator manually unlocks it. This is the most secure option. However, even allowing the account to reset after as little as 10 minutes will serve to slow down a hacker who is attempting to brute force the password.
- **Account Lockout Threshold** represents how many invalid passwords a user can attempt before locking out the account. A setting of three invalid logon attempts is usually considered adequate. If the number is too low, a simple typo could result in an account being locked out. If this is set to 0 (insecure), the account will never be locked out.
- **Reset Account Lockout Counter After** determines how long before the invalid attempt counter is reset. The default setting of 30 minutes is usually adequate. A longer setting is considered more secure.

Figure 3.9 shows the account lockout policy setting and MMC console.

Figure 3.9 Account Lockout Policy



Audit Policy

Chapter 5 covers the specific settings for controlling various aspects of your auditing settings. Don't forget that the audit trail you configure can help you catch a hacker, and sometimes help you troubleshoot issues. Having extensive auditing of failed access attempts, for example, can sometimes help you isolate a rights issue that is keeping software from properly running. Refer to Chapter 5 when configuring your policy for auditing.

User Rights Assignment

The list of configurable events under user rights assignments is extensive. These settings (Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignments) allow you to configure what users (including the accounts processes run as) can do. Many of these settings are used only in unusual circumstances to allow particular software to run. Rather than elaborate on all of the specific settings in this category, we will examine only the *most significant* configurable events and recommend the most secure settings for the majority of environments:

- **Access This Computer from the Network.** You can safely set this to Authenticated Users and remove all other access. You should make certain that ANONYMOUS LOGON is not in the allowed list. In some cases it is more secure to remove the ability to log on locally for normal users. For example, a nonadministrator account really has no reason to need to log on to your domain controller.
- **Act As Part of the Operating System.** No accounts should need this privilege. It allows a user to impersonate any other user on the system without authentication. This would pose a huge security risk, and it would render your other auditing events meaningless. If an application needs this type of access to function properly, it should use the LocalSystem account, which includes this access by default.
- **Bypass Traverse Checking.** This setting allows a user to navigate through directory trees even if the user does not have access to a directory. It does not allow the user to list the directory contents of a directory to which he does not have the appropriate rights. You should set this to Administrators Only.
- **Change the System Time.** While setting the time might not seem important at first glance, setting it incorrectly can create a huge security hole. If you do not set the time correctly, certain encryption systems such as IPsec will fail. Further, it becomes impossible to accurately correlate event logs, and critical transactions could fail, causing a denial of service for legitimate traffic. You should set this setting to Administrators Only.
- **Create Token Object.** This setting allows an account to create a token that can be used to gain access to any system resource. You should not need to set this right manually on any account. If you need this right, you should assign it to the LocalSystem account.
- **Debug Programs.** This right will allow a user to attach a debugger to a process which, in turn, will give the user access to many sensitive internal resources. You should assign this right with great care. Usually you will not need it on a production host anyway.

- **Deny Access to This Computer from the Network.** This should include the local Administrator account. There is no legitimate need for the local account to access the system over the network.
- **Deny Logon as Batch Job, and Deny Logon as Service.** You should set both of these to the local administrator. By doing this, you ensure that if the local Administrator account is compromised, the attacker will not be able to immediately install a service or batch job to further compromise system security.

Security Options

This group of settings is also extensive (Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options) and offers important security settings that impact the entire system, instead of individual accounts. For most of these settings, the purpose and function will be fairly obvious, but a brief description should hopefully clear up any lingering doubts. The following list represents the settings that are the most important for securing your systems:

- **Do not display last username in logon screen.** You should set this to Enabled. By displaying the last user to log on, you are giving any attacker that can get to that logon screen a first clue as to what a viable account name to attack might be. This may not prevent attempts by attackers who have done some reconnaissance, but it will not provide 50 percent of the credentials to be available to the casual attacker.
- **Message text for users attempting to log on.** You will see this message when someone attempts to log on to the console directly (after entering **Ctrl + Alt + Delete**). This setting, also known as the *logon banner*, gives some legal protection against unauthorized access. Fill this in with a message stating that only authorized users should be accessing the system. Your organization's IT security staff and its legal department should work together to develop a suitable message. The purpose of this message is primarily to remove attackers' claims that they didn't know they were doing anything wrong.

- **Message title for users attempting to log on.** This is the title for the preceding message. Something suitably ominous such as “**Warning**” or “**Authorized Users Only**” should be adequate. Consult your legal department to be safe.
- **Number of previous logons to cache (in case domain controller is not available).** If this is a stand-alone server, there should be no credentials to cache and you can set this to zero (which disables caching). Even if you are using domain authentication on your bastion host, the most secure setting is a setting of 0.
- **Rename Administrator account.** As mentioned earlier, the local Administrator account is the most popular user account for attack. Changing this account name to something other than the default can help prevent the success of some automated attacks, such as an automated password-cracking attack against the local Administrator account. You should avoid any obvious alternatives, such as “Admin” or “root.”
- **Rename Guest account.** For the same reasons as the Administrator account, you should select this option as well. While the Guest account has few privileges, it can still provide a local logon account and act as a first step toward elevating an attacker’s privileges. Code Red, for example, adds Guest to the local Administrators group. Since this account name is mentioned specifically in Code Red’s payload, merely renaming this account would prevent such a group membership modification from succeeding.

**TIP**

While some of the policy settings will take effect immediately, some will take effect only after the system is rebooted. For this reason, you should immediately reboot after making any policy changes to ensure that the changes take effect.

Hardening Linux Systems

When you install a given Windows operating system (OS), you will always get the same basic installation for that version of OS, including the default security settings. Basically, all versions of Windows XP are the same out of the box, unless you have customized the installation yourself. With Linux, this isn't true. Some distributions you install are very secure as part of their default configuration and others are very insecure. At a high level, the hardening steps remain the same for Linux as they do for Windows, but some of them may be done for you to a greater or lesser extent. The following list represents the high-level hardening tasks that you would need to perform:

- Remove unnecessary software.
- Disable unneeded services/daemons.
- Patch the operating system and any remaining software.
- Configure user and group accounts to provide only the minimum required access.
- Tighten operating system parameters (login requirements, timeouts, and so on).
- Configure network access to permit only the minimum required connectivity (IP/ports).

These are the same steps you would perform to harden a Windows system. A special-purpose distribution, such as the Smoothwall firewall, performs most of these steps except for applying the latest patches as part of the basic installation. At the other end of the spectrum are distributions that emphasize user friendliness, such as Ubuntu or Red Hat. If a default install is performed with these types of distributions there will be a lot of hardening work left to do. As such, they may not represent the best choice in a high-security environment.

General Hardening Steps

Some of the most basic ways to implement logical access controls is through the use of users and groups (to establish authentication), and then to apply

file-level access controls based on those identities. The basic procedures are the same for Linux as they are for Windows, but the commands are different, of course. We will walk through how to create users and groups, assign users to groups, and grant access to users and groups within the file system.

Users and Groups

Most Linux distributions that include a GUI will include a GUI interface for managing users and groups in addition to the normal command-line tools. If you have access to a GUI utility, that will probably be the easiest way to configure your users and groups. The only time the command line would be preferable would be if you need to make a large number of changes at one time. Using Fedora Core 5, you can navigate to **System | Administration | Users and Groups** to open the GUI interface for the User Manager, as shown in Figure 3.10.

Figure 3.10 Linux GUI User Manager



The interface probably looks a little “busier” than the Windows interfaces, but managing users isn’t too difficult. Let’s start by looking at the GUI to see what information we have available. First, you can see that this system has several users created, including ones named *user*, *sguil*, and *snort*. You may notice that the root user is missing from the list. There are actually an additional 40

or so accounts, but you have to navigate to **Preferences** and remove the checkmark next to **Filter system users and groups** in order to see them in the list. The User ID was automatically generated for these accounts, incrementing them from 500 upward. The user ID is analogous to the username, in that it is a unique identifier for a user; however, the system will use the user ID to track file and directory permissions.

If you click **Add User** you are presented with a window offering all the standard fields, including User Name, Full Name, Password, and so on. This process is very straightforward and doesn't really need much explanation. After you create a user, you can highlight the user in the list and click **Properties**. The tabbed window shown in Figure 3.11 allows you to set various options, including an expiration time for the account, which can be valuable when you have temporary staff or other accounts you need for only a limited time.

Figure 3.11 Linux GUI User Properties



Other tabs allow you to change the password (User Data), configure password aging policies (Password Info), and assign users to groups (Groups). All of this may seem overly simplistic, but that's because a lot of things were done automatically by using the GUI. You may not always have the GUI available, and being able to quickly add a user from the command line can be a valuable ability. You can add a user via the command line with the *adduser* command. A simple example is *adduser test2*. This will create a new user named

test2, and it will use the defaults for all the other required settings. In most cases, this will also create the user directory at */home/test2*, though you can override this behavior with the *-M* switch. You can set a password for the new account using *passwd test2*, after which you will be prompted to enter the new password and then confirm the new password.

Every user account that you create must belong to at least one group. By default, a group will be created with the same name as the newly created user account. There is no “everyone” group, like there is in Windows, so if you need a global account to which to assign broad access rights you have two choices. You can create a group yourself and assign the required access to that group, or you can rely on the access that is granted when no other access has been explicitly defined. In my experience, one of the most difficult things about using Linux is that if you are not familiar with the various commands to use, finding them can be difficult. For this reason, I have compiled the following list of the various user/group administration commands dealing with user or group administration that are both most useful and most likely to be supported on a variety of Linux distributions:

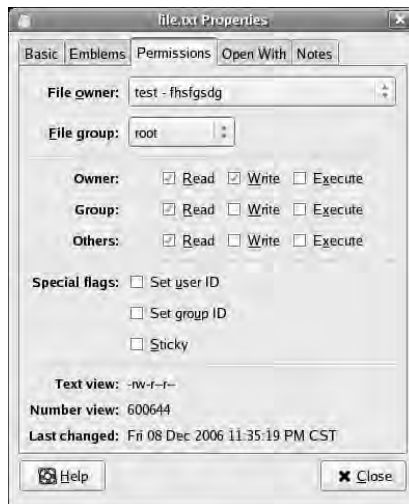
- **adduser/useradd.** *adduser test* will add a user to the system named *test*.
- **chage.** Used to change the time the user’s password will expire and various other parameters related to password aging.
- **chown.** Changes the owner of the file(s) to another user. Using *chown test file.txt* would change the owner of *file.txt* to the user *test*.
- **gpasswd.** Allows you to add or remove users from groups. Using *gpasswd -a test test2* would add the *test* user to the *test2* group. Using *-d* will delete a user from a group.
- **groupadd.** Creates a new group.
- **groupdel.** Deletes a group.
- **groups.** Prints all the groups a user is in.
- **id.** Prints the real and effective user id and group ids.
- **passwd.** Set a user’s password. Use *passwd newuser* to set the *newuser*’s password.

- **su.** Allows you to change the effective user and group of the current user. Entering `su testuser` would change the current user to `testuser`. If the current user is root, no password will be needed. The `-m` option will preserve the original user's environment variables instead of resetting them.
- **userdel.** Deletes a user account.
- **usermod.** Allows you to configure various settings for a given user account.

File-Level Access Controls

Linux file permissions are the source of a lot of frustration for those not familiar with Linux. The system, while elegantly simple, is not intuitive. To help sort it out, we will again start with the GUI interface. Open the GUI file manager for your distribution, right-click on a file, and select **Properties**. The window you see should look similar to the one shown in Figure 3.12.

Figure 3.12 Linux GUI File Permissions



You can clearly see the file owner and the group to which the file belongs. The next section lists which permissions (read, write, and execute) the file owner, file group, and all others have to the file. The special flags sec-

tion lists whether the set user ID, set group ID, or sticky bit is set. These bits are important, so we will discuss them in detail. The last section, with the text view and number view, displays the respective views for the file permissions listed above. We will explore those as well, but for now, just make note of them.

The set user ID and set group ID bits are also referred to as the *SUID* and *SGID* bits. In simple terms, the SUID bit causes a given file/application to run as the user who is the owner of the file, rather than the user who is executing the file. Most often this is used to allow a normal, nonprivileged user to run a script or process requiring root access. The first thing that comes to most people's minds is that this would be a big security risk. Well, like so many other things, it can be a big risk, but it can also increase security. An obvious example of a big security risk would be one where the application or script is interactive, or allows manipulation of other system parameters. Obviously, allowing an unprivileged user to run a file manager as the root user would be a big security risk. Many security scanners will report the number of SUID files which are also owned by root for exactly this reason. The following command would search the entire file system from the root for files that are owned by root, and have the SUID bit set:

```
find / -type f -perm /4000 -uid 0 -ls
```


Find is the utility that is doing the searching (type **man find** at the Linux command prompt for more information). The *type -f* option says to look for normal files (instead of named pipes and other, more unusual options). The *-perm* option specifies that we are looking for the SUID bit being set. The *-uid* option specifies the owner as user ID 0, which is always root. Finally, *-ls* tells *find* to format the output such that all unusual characters are escaped. This is for your own protection in case any filenames are using nonprinting characters that could cause problems within your terminal. You can use the same command with *-perm /2000* to list the SGID files belonging to the root group.

SUID can increase your security as well. Suppose you have a program or script that you want the user to run, to perform some administrative task automatically. If this program requires root permissions to function properly, you would have to give the user the root password, or run the program yourself. Instead, the administrator can set the program SUID bit and make root

the owner, allowing the user to run the program without ever needing to know the root user password. Basically, whether SUID is an asset or a liability depends on the program to which it is attached. The SGID bit does the same thing, except the program is run as the owning group. All of the same security considerations apply for SGID as for SUID.

The sticky bit is much less interesting than it sounds. When set, it tells the operating system not to unload the executable from memory after a user closes it. In this fashion, the program (the sticky bit applies only to executable files) will now start faster. This can have some security implications, but in reality, is it not used much anymore and is mostly a holdover from when Linux was intended to serve as a multiuser operating system.

With those special bits out of the way, let's look at the basic file system permissions. You can list the files in a directory in long list format, including the detailed file permissions, using `ls -o`:



```
ls -o
total 116
drwxr-xr-x 5 root  4096 Dec  9 00:10 Desktop
-rw-r--r-- 1 test   164 Dec  8 23:32 file (copy).txt
-rwSr--r-- 1 test   164 Dec  8 23:32 file.txt
-rw-r--r-- 1 root 26225 Sep  9 14:37 install.log
-rw-r--r-- 1 root  3973 Sep  9 14:36 install.log.syslog
```

You'll notice that the `file.txt` file we looked at earlier is different from all the rest. The leftmost column contains the permissions for the file, and the series of dashes and letters corresponds to the grid of checkboxes in the GUI permissions tab. The breakdown of the permissions designations is as follows.

```
1222333444
```

1. This first character designates the object type. Usually the object is a directory (d), but other options are possible, such as a socket (s) or a named pipe (p). A directory will contain a d in the leftmost position.
2. The next three characters indicate the permissions granted to the owner of the object (*r* = read, *w* = write, *x* = execute). *x* is replaced with *S* when the SUID bit is set.

3. These characters represent the permissions granted to the group the object belongs to, using the same *rwX* notation. The *X* bit is replaced with an *s* when the SGID bit is set.
4. These last characters indicate the permissions granted to users who are neither the object owner nor in the object's group. The *rwX* notation is the same. The *X* is replaced with a *t* when the sticky bit is set.

If you refer back to the file permissions in Figure 3.12, you will notice that the “number view” is listed. This is actually yet another way to represent file permissions. As you check or uncheck the various permissions, the number view will be updated and will display the new number view. If the pattern to the numbers is a little hard to sort out, here is how it works. The rightmost digit indicates the permission for “other” users. The next character to the left represents the permissions for the group, and the next one to the left represents the permissions for the object owner. Read access corresponds to a *4*, write corresponds to a *2*, and execute to a *1*. Each of these (*rwX*) is added together to indicate the access. Thus, read access by the owner, group, and other would be *444* for the rightmost numbers. Read and write access for all three would be *666*, and finally, *rwX* for all three would be represented numerically as *777*. You can use the *chmod* utility to change the permissions on a file from the command line, as shown here:

```
SYNGRESS  
syngress.com # ls file.txt -o  
-r--r--r-- test 164 Dec  8 23:32 file.txt  
# chmod 777 file.txt  
# ls file.txt -o  
-rwxrwxrwx test 164 Dec  8 23:32 file.txt
```

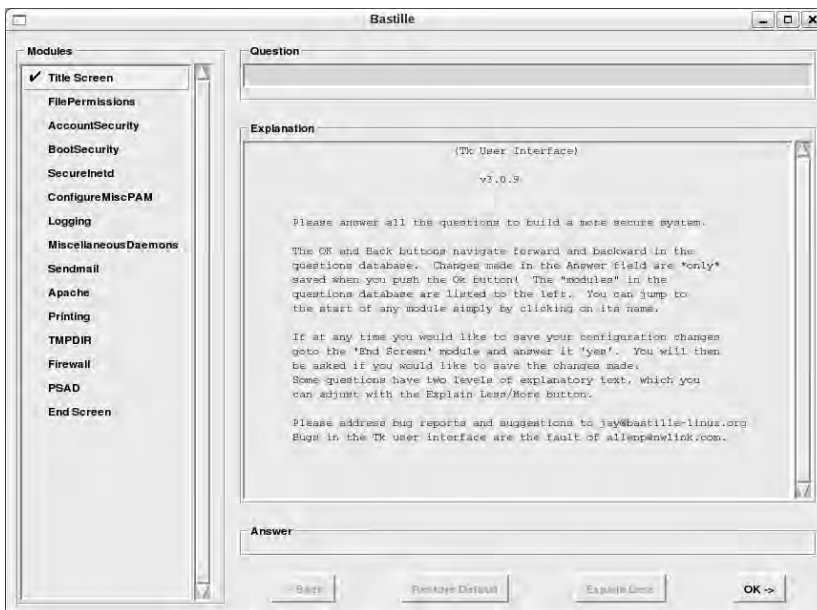
Administering Linux file permissions is not more difficult than doing so for a Windows file system, but there are a lot of utilities involved when administering from the command line. Understanding permissions will be essential to properly lock down users and restrict their access to only the minimum that is needed. For more advanced configuration suggestions, you might want to research the following commands related to disk usage quotas: *quota*, *quotaoff*, *quotaon*, *quotacheck*, *repquota*, and *edquota*. You also can enable process accounting using the *accton* command.

Using the Bastille Hardening Script

There are many settings to configure when it comes to hardening your Linux distribution, and all you have to do is miss one setting and a hacker can compromise the system. To help combat this, many people have developed semi-automated scripts to help harden a Linux system. These scripts basically just make a bunch of configuration changes automatically based on configuration selections that you would have otherwise had to make manually. One of the oldest hardening scripts, and thus one of the most mature and well developed, is the Bastille Linux script. It has evolved from a crude basic script to a well-refined hardening system with a GUI interface. Bastille currently supports Red Hat Enterprise Linux, Fedora Core, SuSE, Debian, Gentoo, Mandrake, and HP-UX, with a Mac OS X version in development. You can read and download Bastille from www.bastille-linux.org.

You can pass only three options to Bastille, each of which tells it to run in a different mode. If you use `bastille --report` it will generate a Web-based report of your current “hardness.” The `-c` option will run the actual hardening script in a text-based mode, and `-x` will run it in a graphical mode, as shown in Figure 3.13.

Figure 3.13 Bastille Graphical Configuration



All you have to do to use Bastille is start it up, and it will present a series of yes/no questions to which you must respond. Based on your answers, it will configure various security settings automatically. These settings include removing SUID bits from some programs, disabling insecure services (such as *rshell*, for example), changing user account expiration, and much more. The script does a good job of presenting you with reasonable defaults, providing the most secure option that will not overly impact normal usage of the system. When you are finished answering the questions, you will be prompted to choose whether to save the resulting Bastille configuration. Note that saving the configuration is *not* the same as applying it, as the next question is whether you want to *apply* the configuration.

Bastille is a powerful tool that helps harden a default install. Depending on your selections, you can easily make a system too secure and unsuitable for use as an everyday workstation. A little experimentation, combined with reading the excellent explanations that Bastille provides, should help you create a very secure system with a minimal amount of effort.

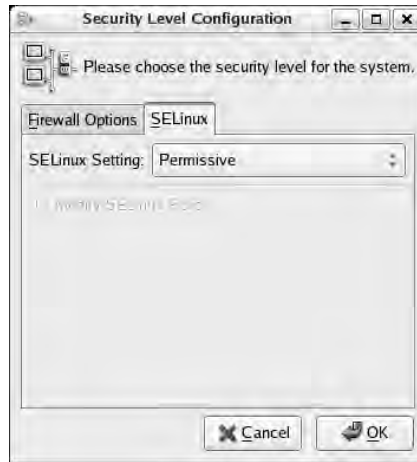
Using SELinux

SELinux stands for *security-enhanced Linux*, and it was developed in partnership with the National Security Agency (NSA). It provides a higher level of security by enforcing mandatory access controls (MAC) through the kernel. *Mandatory access controls* are very different from the standard *discretionary access controls*. Most systems are said to use discretionary access controls because someone (the file owner, or the root/administrator user account) has discretion over who has access to what. Mandatory access control is based on the principle that a given role has predetermined access rights, and these are immutable. Basically, the only way to change access permissions is for the user to be assigned to a new role (called *contexts* in SELinux terminology). Because the enforcement is through the kernel, it can restrict the actions of *any* process, even a process run by the root user. As far as the underlying components of SELinux are concerned, there is no concept of a root user, only security policies and security contexts. SELinux is available for many Linux distributions (see which ones at <http://selinux.sourceforge.net/distros/redhat.php3>). SELinux is installed by default (though disabled) on Fedora Core 5. You can

read the SELinux FAQ from the NSA at www.nsa.gov/selinux/info/faq.cfm for more information.

SELinux is a work in progress. Currently the setup and configuration can be rather complicated. You can enable SELinux on Fedora Core by navigating to **System | Administration | Security Level and Firewall**. On the **SELinux** tab, you enable SELinux by changing the **SELinux Setting** from Disabled to **Enforcing** or **Permissive** and clicking **OK**, as shown in Figure 3.14.

Figure 3.14 Enabling SELinux

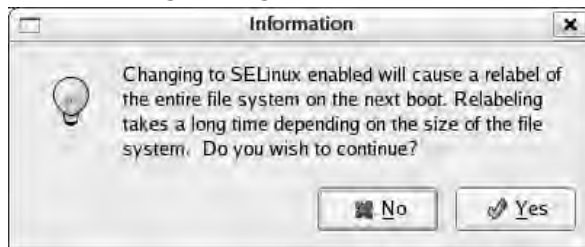


SELinux uses the *xattr* labels within the file system to generate labels describing the security context of a file or directory which are persistent across reboots. These labels are not normally used; thus, when you enable SELinux, you will get the warning dialog shown in Figure 3.15. You must select **Yes** in order to enable SELinux. Then click **OK** on the **Security Level Configuration** window, and then reboot.

It is recommended that you select Permissive initially, in which case the system will generate logs based on the configured SELinux policy but will not actually restrict any activities. This allows you to see the impact of a policy without impacting productivity. You can then use the results of the log files to adjust your policy until it is usable. Once you are satisfied with the SELinux rules, you can enable enforcing mode, which will actually apply your configured policy. You can define your policies as targeted or strict. A *strict* policy

applies to all processes and files on the system, and a *targeted* policy is applied to only specific files. Strict mode is very difficult to configure properly. Targeted is easier to configure and is the default policy type when SELinux is first enabled. You can check the status of SELinux by typing **sestatus**.

Figure 3.15 Relabel Warning Dialog



If you navigate back to the **Security Level Configuration** window, and then click to expand **Modify SELinux Policy** on the **SELinux** tab, you will be presented with a list of options to toggle various SELinux settings. These options represent only a limited set of preconfigured choices to toggle settings in the SELinux policy files. For any serious configuration, you will need to edit the files manually. You can also download a third-party SELinux policy editor from the SELinux Policy Editor Project (<http://seedit.sourceforge.net/index.html>). This package includes a simplified set of tools that is slightly less functional than the normal package. You can, however, switch between using one or the other.

You can see what actions the policy *would* have denied if it were enforced by reviewing the logs. You can find log messages from SELinux in `/var/log/messages` or `/var/log/audit/audit.log`. Enabling SELinux is not a project to be undertaken lightly. Implementing SELinux in a manner that is both useful and functional will likely require a good deal of investigation and research. It is recommended that you read the documentation on the Web site of your chosen distribution if SELinux is supported, as well as the documentation on the official SELinux Web site (www.nsa.gov/selinux).

Hardening Infrastructure Devices

Don't overlook hardening your infrastructure devices. Not all routers and switches have administrative capability, but many do. For those that do,

referred to as *managed* devices, they usually allow you to control many aspects of the device, including redirecting traffic to ports of your choosing and basically enabling or disabling all traffic flow through the device. Given the often central role these devices fulfill in your network, control over one of them will often mean control over your entire business. For this reason, you should exercise the same care and due diligence in securing your infrastructure devices as you would your critical servers. The same high-level bullets for hardening host-based systems also apply to managed infrastructure devices.

Most managed devices will have a means to authenticate using a local account as well as a central authentication server, such as TACACS or RADIUS. Ensure that the accounts are secured and a high-quality password is used. Sometimes even routers and switches will have unneeded services installed by default. One common example is enabling an HTTP interface for managing the devices. While this can certainly be handy, often the Web interface opens up an entire category of potential security risks that would not otherwise be present. The highest level of security is achieved by disabling any services that are not needed. Conservative timeouts for abandoned sessions and a login warning banner are advisable security measures.

You will also need to update the software on the device. Given the criticality and potential scope of impact for these devices if an update causes a problem, these devices are rarely configured to update automatically. In most cases, this will be a manual process which you must incorporate into your patch management and change management processes. Pay extra attention to any device connected to the Internet as these are going to be attacked on a regular basis. You must secure them before you connect them to the Internet, or you will likely lose control of them in short order.

Patching Systems

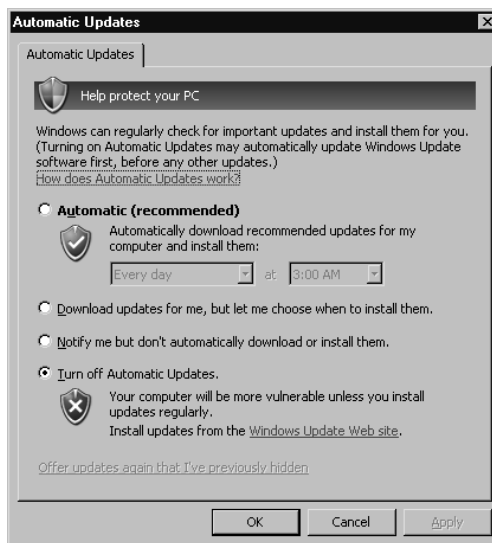
Patching systems is an age-old chore that no one likes. Besides the time and labor involved, sometimes an update will actually do more harm than good, breaking some functionality or, in a worst-case scenario, rendering the system unusable and requiring a complete rebuild. While both the labor involved and the number of adverse reactions to patches have decreased as patching methodologies have matured, it is still not a fun task. In this section, we will look at ways to keep your systems patched with a minimal amount of effort.

Patching Windows Systems

Windows systems are known for breaking from updates. This is true to some degree with patches, but service packs (which are nothing more than several patches rolled into one) are almost legendary for breaking things. Microsoft has made the task of keeping your systems up-to-date nearly painless with Automatic Updates. Automatic Updates allows a system to download and install patches automatically as they are released. Now, given the fact that any patch runs a risk of causing undesirable side effects, you might question the safety of using Automatic Updates. This is a reasonable precaution. Keep in mind that millions of users are using Automatic Updates every day without incident (Automatic Updates is enabled automatically on Windows XP when SP2 is installed). The configuration options allowed within Automatic Updates can help lessen the risk of applying updates automatically.

Within the Control Panel you should have an Automatic Updates icon. Clicking it will open the Automatic Updates configuration window shown in Figure 3.16.

Figure 3.16 Windows Automatic Updates

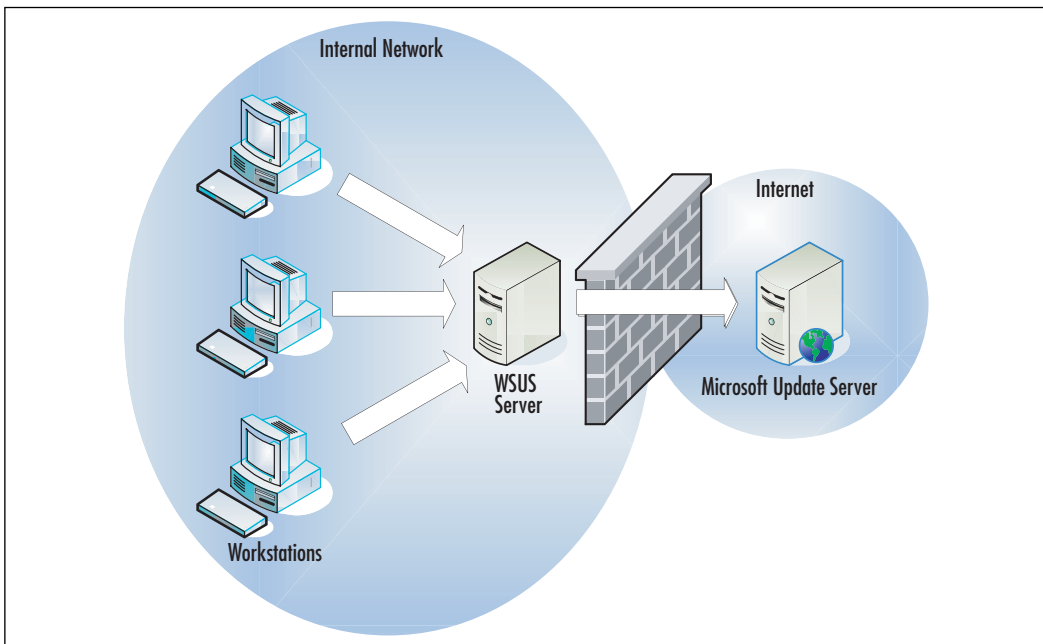


Each of the four options offers a different level of invasiveness on behalf of the Automatic Updates service. At the bottom, you can disable automatic

updates completely. This offers the most control over the update process. Because you will need to apply any updates manually, this ensures that you will be on hand and that the updates won't catch you by surprise. The function of the other options is self-explanatory. While these options do offer some security, in many cases they do not provide enough control for some organizations.

Windows Server Update Services (WSUS) offer increased control over the update process, and some other advantages as well. You can read about WSUS at www.microsoft.com/windowsserversystem/updateservices/default.aspx. Basically, WSUS acts as an intermediary between the systems using Automatic Updates and the Windows Update server at Microsoft. You configure the internal hosts to use the WSUS server to retrieve updates, and the WSUS server will offer greater control over which updates are to be installed and when. WSUS is currently considered beta software and you must register in order to download the server component. Figure 3.17 depicts the WSUS update process.

Figure 3.17 WSUS Process



This solution offers more advantages than just control of the updates installed using Automatic Updates. You also can realize considerable bandwidth savings through the use of WSUS. If 20 machines download a 25MB update, you have to download 500 MB. If you are using only a T1 for your Internet connectivity, this will saturate the link for around 45 minutes solid. If you use the Internet for business reasons, this could be very inconvenient. Using WSUS, the WSUS server will download the 25MB update once, and all other systems will download the updates from the WSUS server without needing to use any Internet bandwidth at all. As tempting as WSUS may be, the server component will run only on Windows Server 2003 or newer. The client computers can be Windows 2000 (SP4) or newer.

Patching Linux Systems

The methods for patching your Linux distribution vary greatly from other distributions. In the case of Fedora Core 5, entering **yum update** without any program name specified will update all programs on the system, including the kernel. Of course, you can update individual programs using yum as well, by specifying the program name—**yum update tcpdump**, for example. You can cause yum to check the repositories and produce a list of all programs for which newer versions exist using the **yum check-update** command. Similar functionality is offered on Debian-based systems using the *apt-get* utility. Using **apt-get update** followed by **apt-get upgrade** will update all installed software. The upgrade option is used to tell *apt-get* to update its application listing database. You can install or update individual packages using **apt-get install tcpdump**. If you are using a GUI desktop environment, most likely a GUI interface will be provided. For Fedora Core 5, there is a GUI package manager at **Applications | System Tools | Software Updater** (shown in Figure 3.18). You can also install a GUI front end for yum, called *yumex* (which stands for yum extender).

Figure 3.18 Fedora Software Updater

Personal Firewalls

Personal firewall is a term that refers to firewalls that protect only the single host on which they reside. While many excellent personal firewalls are free versions of commercial software, you should read the license agreement for these firewalls carefully. In most cases, the free use extends only to home users and specifically excludes use in a business environment. One example is ZoneAlarm, from Zone Labs (www.zonelabs.com), which comes in a free version as well as a commercial offering. The free version offers great program control but is for nonbusiness use only. Kerio Personal Firewall (www.sunbelt-software.com/Kerio-Download.cfm) follows a similar business model, with the free version being for “personal use” only. For any freeware products you find, and there are several excellent ones, be sure to review the license agreement carefully to ensure that you can deploy it within your organization legally. If you are unsure, consult your organization’s legal council to determine whether a given product license is suitable.

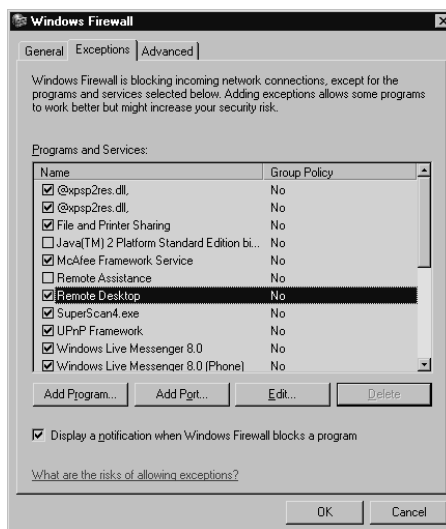
Windows Firewall

Although it hasn’t been around for as long as the netfilter firewall (for Linux), the Windows firewall enjoys a similar advantage in that it is included with all modern Windows operating systems (Windows XP and Windows 2003).

Because it is enabled by default, most users have some firewall protection in place even if they don't know it. While fairly ubiquitous, the Windows Firewall is very simple in its configuration and capability. It cannot perform any advanced packet manipulation and is really designed only to protect the host on which it is running. You can access your Windows Firewall by navigating to **Start | Settings | Control Panel | Windows Firewall**. The **General** tab has only three settings: an **On** setting, and **Off** setting, and a **Don't allow exceptions** option. If you select **Don't allow exceptions**, any selections you make on the **Exceptions** tab are simply ignored.

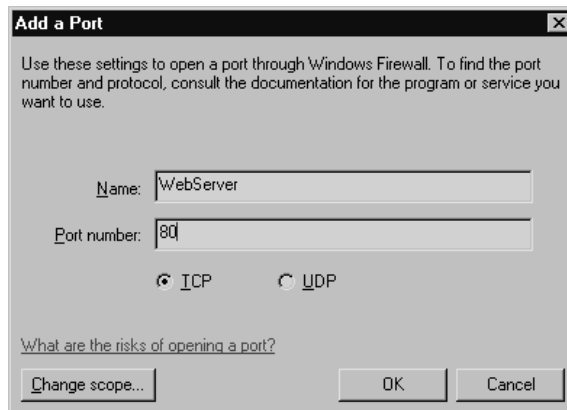
The **Exceptions** tab is where all the fun happens. The Windows Firewall operates as most personal firewalls do by default, which is to allow the local system to communicate outbound unhindered. The only traffic that is allowed inbound to the interface is reply traffic to sessions that were established outbound first. In this way, your surfing and other network access is unimpeded, but others cannot initiate a session to the protected system. This is usually perfectly adequate; however, if you happen to be running any type of server on the local machine, no one will be able to initiate a connection to the listening port. In other words, if you had a Web server running on the protected system, the Windows Firewall would block connection attempts to port 80. The **Exceptions** tab allows you to configure an exception to that normal behavior and allow a particular port in. Figure 3.19 shows the Exceptions tab.

Figure 3.19 Windows Firewall Exceptions



One nice feature of the Windows Firewall is the ability to permit access based on the program that is running instead of on the port number. For the standard services, you probably know what port is needed, but sometimes a single application might need a large number of ports, or it may use a custom port and you're not sure which port it needs. In these cases, you can permit the application to open a listening port, and the firewall will allow inbound connections. To do this simply click **Add Program**, navigate to the program needing access, and then click **OK**. If you happen to know the port you need to open, this is easy as well. Click **Add Port**, and enter a reference name for the rule, such as **Web Server**. Then enter the port number—**80**, for example. As a final step, select whether to open the port for TCP or UDP connections using the radio buttons. Figure 3.20 shows the Add a Port window.

Figure 3.20 Windows Firewall Add a Port Window



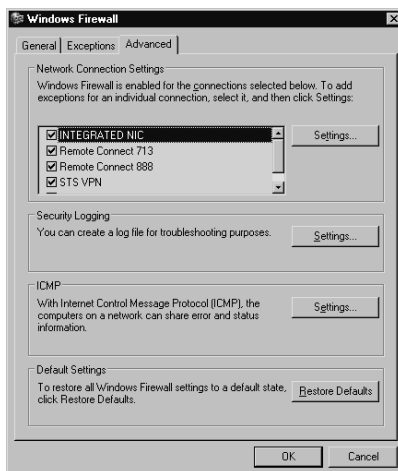
By clicking the **Change Scope** button, you can change the systems for which the port will be open. This allows you greater control over who will be able to connect to these ports. Your available options are Any Computer, My Network (meaning the subnet that the host is on), or a custom list of IP addresses. File sharing access would be a good example of when changing the scope from the default (Any) may be advisable. As long as your entire network is on the same subnet, there is no reason to leave the scope at Any. By setting it to My Network, you ensure that only local IP addresses can connect.

WARNING

Depending on how your Internet connection is configured, if the Internet router will NAT inbound traffic, traffic originating from the Internet or an external network may be NATed to an internal address. If this is the case, the external address would appear as an internal address from an IP subnet perspective and a Windows Firewall scope of My Network would still grant them access.

The Advanced tab (shown in Figure 3.21) has a few interesting settings of its own. Starting from the bottom of the window and moving upward, the setting with the simplest options is the **Restore Defaults** button. Of course, this will restore the default settings for the Windows Firewall, which is useful if the configuration is messed up and starting fresh is the easiest option. The next section is ICMP. By clicking the **Settings** button, you can configure a list of checkboxes, with each one corresponding to another type of ICMP packet. If you highlight a particular option a description will be displayed below. The descriptions are pretty helpful, and in most cases you shouldn't need to edit these unless you have some specific needs. The Security Logging section also has a **Settings** button. Clicking the **Settings** button will allow you to configure whether dropped packets and successful connections should be logged. You can also specify the path for the log file and enter a maximum log size.

Figure 3.21 Windows Firewall Advanced



The top section, Network Connection Settings, allows you to select which interfaces will have the Windows Firewall enabled. In this way, it doesn't have to be an all-or-nothing proposition. You can specify that one network card has the Windows Firewall working and another one doesn't. This section also has a **Settings** button of its own. The **Settings** button applies only to the highlighted interface. This allows you to configure additional port exceptions on a per-interface basis. Clicking the **Settings** button will open an **Advanced Settings** window. From within the **Advanced Settings** window you can permit services, similar to the **Exceptions** tab, however these services apply only on the interface you highlighted when you clicked **Settings**. You can also use the **ICMP** tab to configure your ICMP permissions on a per-interface basis.

When it comes to Windows-based systems, little administration is done from the command line. In most cases, GUI tools are used. If you need to script things, often for use in login scripts, command-line tools can be invaluable. Microsoft added Netsh firewall support as of Service Pack 2, which allows you to manipulate the Windows Firewall from the command line. To access the firewall context of Netsh, enter **netsh** and then **firewall** at the command prompt to use Netsh interactive mode. Your command prompt will change to **netsh firewall>**. You can view useful diagnostic information by entering **show state** or **show config**. You can display the same information without entering interactive mode by entering **netsh firewall show state** from a command prompt, as follows:



```
C:\>netsh firewall show state
```

```
Firewall status:
```

```
-----
Profile                               = Standard
Operational mode                       = Disable
Exception mode                         = Enable
Multicast/broadcast response mode     = Enable
Notification mode                      = Enable
Group policy version                   = None
Remote admin mode                      = Disable
```

```
Ports currently open on all network interfaces:
```


Port	Protocol	Version	Program
137	UDP	IPv4	(null)
139	TCP	IPv4	(null)
138	UDP	IPv4	(null)
3389	TCP	IPv4	(null)
445	TCP	IPv4	(null)

Additional useful commands for scripting purposes might be **netsh firewall show allowedprogram**, **netsh firewall show logging**, and **netsh firewall show portopening**. All of these commands can provide valuable data, which you can use to scan systems programmatically for certain settings.

With the command-line utility, you can easily make many changes to the Windows Firewall configuration at one time using a batch file, or login script. This is advantageous in that making many changes via the GUI would be very time consuming, but making those changes via the command line could be done with a batch file very quickly. If you wanted to add an allowed program to the Windows Firewall, you would use the following command syntax to add a program, and then verify that it was added successfully:



```
C:\>netsh firewall add allowedprogram C:\program.exe appname ENABLE
Ok.
```

```
C:\>netsh firewall show allowedprogram
```

```
Allowed programs configuration for Domain profile:
```

```
Mode      Name / Program
```

```
-----
Enable    Remote Assistance / C:\WINDOWS\system32\sessmgr.exe
Enable    AOL Instant Messenger / I:\Internet\AIM95\aim.exe
```

```
Allowed programs configuration for Standard profile:
```

```
Mode      Name / Program
```

```
-----
Enable    Remote Assistance / C:\WINDOWS\system32\sessmgr.exe
Enable    AOL Instant Messenger / I:\Internet\AIM95\aim.exe
Enable    appname / C:\program.exe
```

You can also enable a particular port using any of the following syntax examples:

```
netsh firewall add portopening TCP 80 Webserver
netsh firewall add portopening UDP 80 Webserver
netsh firewall add portopening ALL 80 Webserver
```

As yet another means to configure and control the Windows Firewall, you can use GPOs. You can find the settings for the Windows Firewall in the MMC Group Policy Editor snap-in. Navigate to **/Computer Configuration/Administrative Templates/Network/Network Connections/Windows Firewall/** to locate the appropriate settings.

The Windows Firewall can usually do an adequate job of protecting the local system. The Windows Firewall configuration allows you to configure global settings on the General and Exceptions tabs, and allows a higher degree of granularity using the Advanced tab. The Advanced tab allows you to configure many of the same settings as the global settings, but you can configure them on a per-interface basis. You also can configure these same settings from the command line using the Netsh utility. While I wouldn't recommend using the Windows Firewall as a perimeter firewall gateway, it can make an excellent personal firewall for your Windows-based network resources.

Tools & Traps...

What's in a Name?

The ability to specify access through the Windows Firewall based on the program name is pretty handy. By doing so, you don't need to figure out which port it needs, or whether the program needs to listen on a lot of ports or manage a large list of ports to allow through the firewall. This ease is not without its pitfalls, however. The access is based on the program name and nothing else. This means if the Windows Firewall is configured to allow inbound connections to *helpdesk.exe*, and a user renames *badapp.exe* to *helpdesk.exe*, connections to *badapp* will be permitted through the firewall. This is why renaming an executable of the user's choosing is a popular method of bypassing Windows Firewall restrictions.

Netfilter Firewall

Netfilter is the firewall component that is included with the Linux kernel. Unless firewall support has been explicitly removed, netfilter should be present on all Linux distributions. Unlike the Windows Firewall, which is really suited only for use as a personal firewall, netfilter is just as capable of serving as a personal firewall as it is a dedicated firewall for your perimeter Internet connection. Detailed usage guidelines, including several recommended GUI interfaces for configuring netfilter, is discussed in detail in Chapter 2. All of the same configuration recommendations that were discussed when implementing netfilter as a perimeter firewall will apply when using it as a personal firewall.

Configuring TCP Wrappers

You can find similar functionality to the Windows Firewall's capability to permit access based on a program name in the Linux utility, TCP Wrappers. TCP Wrappers works in conjunction with the network-level filtering of a firewall. Whereas a firewall permits or denies traffic based on data contained in the IP header of the packet, TCP Wrappers filters access to services (by name) on the host on which the service is running. Only services that are compiled against the libwrap.a library can use TCP Wrappers. With TCP Wrappers enabled, attempts to access a given service will be compared against the `/etc/hosts.allow` file, and then the `/etc/hosts.deny` file. Rules are checked sequentially and processing of the rules files stops when a match is found. If a match is not found in either file, access will be granted. As an example, if you wanted to allow only connections from `syngress.com` to SSH on your bastion host while rejecting all other attempts, you would have the following two lines in your `hosts.allow` and `hosts.deny` files:

```
/etc/hosts.allow  
sshd : .syngress.com
```

```
/etc/hosts.deny  
sshd : ALL
```

While you would not want to rely on TCP Wrappers as your only means of protection, it does have the advantage that in the preceding example, access

would be permitted to *sshd* regardless of the port on which it was listening. The two filter files accept several wildcards, such as *ALL*, *LOCAL*, *KNOWN*, *UNKNOWN*, and *PARANOID*. You can enable logging in the rule files as well, and configure the facility and severity of the log entry. With TCP Wrappers' limited functionality and syntax, you might wonder why you would ever use it over simply using netfilter. Because iptables works at the packet level, if you want to deny access to a particular process, such as HTTP, you must do it based on port number. So if you use netfilter to explicitly block connection attempts to port 80, and the user starts up the Web server and tells it to listen on port 8080, the connection will be allowed. With TCP Wrappers, you permit or deny access to a process. In this way, you can ensure that a given process will work only on the port you want it to work. This distinction could prove invaluable if you have a service that uses a large number of listening ports, or some type of service that is spawned as needed and the port number isn't always consistent, or if the packets are tunneled in another protocol, rendering identification via port numbers impossible.

Providing Antivirus and Antispyware Protection

Antivirus and antispyware efforts are a necessary evil in today's world. Having no protection from viruses and spyware will almost guarantee that you will fall victim to one or the other eventually. Almost everyone has a good feel for the risk a virus can pose, from just being a nuisance to rendering a system inoperable. Spyware, despite all the hype, often doesn't get the attention it deserves. I have seen many systems that have become so choked with spyware-consuming resources that they have slowed to a crawl and are no longer usable. Worse yet, some spyware is very difficult to remove, which translates into more lost time and effort just to clean the computer. In the end, it's definitely better not to let any spyware or viruses on a system in the first place than it is to have to deal with the hassle of trying to clean up the mess after the fact.

Antivirus Software

Antivirus software is required on all systems. Most computers these days come with antivirus software installed, typically with a limited duration of updates for the virus definitions (often as short as 30 days or, at best, a year).

Unfortunately, many systems are never updated and continue running the out-of-date virus signatures. While an out-of-date antivirus program is still better than no antivirus program at all, it will not offer you much protection from the latest virus threats. Whichever antivirus methodology you use (Automatic Updates, live Web scans, manual updates), you will need to ensure that the signature files are maintained and remain current. Several of the best free antivirus programs have license agreements limiting their use to home or personal computers, specifically excluding a business environment. However, there are still some excellent free alternatives.

Clam AntiVirus

Clam AntiVirus (www.clamav.net) is an open source, free antivirus program with a lot of commercial-grade features. Clam was natively written for use on UNIX/Linux systems, but a Windows port also is available. Clam provides a command-line scanner for spot scans or scripted scanning. This is the primary way Clam was intended to be used. There is also an (experimental) on-access scanner, similar to what you'd find in most commercial products. It even includes a utility to update your signature files automatically. You can view the latest online documentation at www.clamav.net/doc/latest/html. Because Clam AntiVirus is one of the few antivirus solutions available without stipulation that it be used for “personal” use, I will discuss using it on both Linux and Windows systems.

Installing Clam AntiVirus on Linux

You should be able to download the needed files to install Clam AntiVirus from your standard repositories. If they are not available, you can download the base packages from www.clamav.net and install them manually. After installation, there is some configuration required. Follow these steps to get Clam AntiVirus running:

1. Install the ClamAV package and any dependencies.
2. You will need to create a ClamAV user and group. Typing the command **useradd -s /bin/false clamav** will create the ClamAV user and will set the user shell to */bin/false*. On Fedora Core 5 */home/clamav* will be created automatically, but you will want to verify that this is true on your distribution. If the directory is not created automatically, you will need to create the directory manually.

You can now run the command-line scanner using the syntax *clamscan <target>*. As an example, *clamscan -r /home* would scan all the home directories and all subdirectories (*-r* specifies a recursive scan). Clamscan includes a large number of command-line options. The most significant ones are explained in the following list (you can enter **clamscan --help** for a more detailed explanation of the many command-line options):

- **--infected**. This causes clamscan to list only infected files in its output.
- **--bell**. This will cause a bell sound to be played if a virus is found.
- **--recursive / -r**. This will cause clamscan to scan recursively—that is, to scan all subdirectories of the specified path.
- **--move=<directory>**. You use this to specify that infected files be moved to a particular location. Typically the target directory is one that normal users have no access to, effectively placing the file in quarantine.

That's about all there is to the on-demand portion of Clam AntiVirus. As soon as you run any of the example commands, or any on-demand scan, clamscan will notify you that the definitions are out-of-date. You can see this error in the following example:

```
[root@localhost ~]# clamscan /input.txt
LibClamAV Warning: *****
LibClamAV Warning: *** The virus database is older than 7 days. ***
LibClamAV Warning: *** Please update it IMMEDIATELY! ***
LibClamAV Warning: *****
: OK
```

```

----- SCAN SUMMARY -----
Known viruses: 75410
Engine version: 0.88.6
Scanned directories: 0
Scanned files: 1
Infected files: 0
Data scanned: 0.00 MB
Time: 3.023 sec (0 m 3 s)
[root@localhost ~]#

```

The next step would be to update the virus definition files. You do this using the *freshclam* updater (the *clamav-update* package in the Fedora repository). Follow these steps to get *freshclam* up and running.

1. Install the *clamav-updater* package.
2. Edit the `/etc/freshclam.conf` file. You may want to configure many settings. These are the settings which you *must* configure at a minimum in order to use *freshclam*:

Locate and edit the following line:

Example

Either delete this line or add a `#` to the beginning of the line so that it will be a comment. This will prevent *freshclam* from running because it assumes you have not edited the example configuration file yet.

Locate and edit the following line:

```
#DatabaseMirror db.XY.clamav.net
```

Remove the `#` to uncomment the line. Change *XY* to your two-letter country code.

3. Verify that the `/var/log/freshclam.log` file is writable by the owner and is owned by the *clamav* user.
4. Verify that the `/var/lib/clamav/` directory is writable by the owner and is owned by the *clamav* user. This is the directory where the signature database files are stored.

 TIP

Unless you use the `--user=` option to specify a different user for *clamav* to run as, it will run as the *clamav* user. This is true even if you are running the process as the root user. As soon as the *clamav* process starts, it drops from the running user to the *clamav* user. This explains why you may run the process as root but still receive a permissions error. In my case, with Fedora Core 5, the permissions were not set correctly for *clamscan* or *freshclam* to run and had to be edited.

You can run *freshclam* with no options and it will use the Internet to update the database definitions. You can automate the database update process in several ways. You could run the same command as the command line and schedule it with *cron* to run as frequently as you like. You could also run *freshclam* in database mode by adding the `-d` switch. If you are going to use database mode, the default is to connect and check for updates every two hours. You can edit this in the *freshclam.conf* file with the line reading `#Checks 24`. Uncomment this line to cause *freshclam* to check for updates every hour (i.e., 24 times per day).

This covers running on-demand scans. You gain a lot of protection by running these scans regularly in this fashion. You may want to evaluate the pros and cons of running an on-access scanner so that all files will be scanned as they are accessed. By scanning files as they are accessed, you may prevent the payload of an infected file from ever being triggered. With a nightly scan, an entire system could be infected by the time the nightly scan runs. You can configure *clamav* to run in this fashion; however, this additional protection comes at a price. By scanning every file as it is accessed, a small additional delay is incurred for each file. On a system that accesses a large number of files, this delay can bring the system to a grinding halt. *Clamuko* functionality (discussed shortly) is also not as stable as the base package, so even if the expected level of file access is low, you should evaluate whether an unexpected lockup is acceptable. The best bet is to run the on-access feature on a test system and evaluate the stability for yourself.

If you do want to enable on-access scanning, you can do so by installing *Clamuko*, which is a thread within the *clamd* daemon process. Unfortunately,

Clamuko requires the use of Dazuko for the file access tracking. Dazuko is not available as a precompiled binary, and in order to install it you must recompile your kernel and incorporate Dazuko into the kernel as a kernel module. You should check your particular distribution, as a few have Dazuko support already integrated into their kernel. If you do not have Dazuko support already enabled, you will need to download the kernel source and the Dazuko source and recompile both, which is beyond the scope of this book.

Installing Clam AntiVirus on Windows

You can download the Windows port of Clam AntiVirus from several different locations (see www.clamav.net/binary.html). One of the most well-polished versions, which includes a GUI front end for performing various scanning tasks and right-click integration with Windows Explorer, is located at www.clamwin.com. It will run on Windows 98, ME, 2000, XP, and 2003. As a bonus, clamwin is available as a portable application (meaning you do not need to execute any type of installation, so you can run it from a USB drive) from <http://portableapps.com>. To install clamwin, download and run the setup utility. Follow the prompts, accepting the license agreement and choosing the installation folder. You can probably accept the defaults for the entire installation. At the end of the installation, it will automatically update the virus definitions (assuming you left that option checked). When it's finished, you should see a clamwin icon that looks like a target in the System Tray.

If you double-click on the icon in the System Tray or navigate to **Start | Programs | ClamWin Antivirus | Virus Scanner** you will open the main windows shown in Figure 3.22.

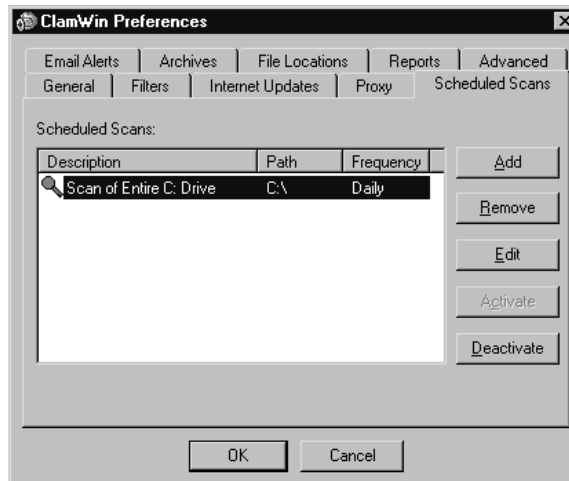
Remember that the Windows version has no support for on-access scanning. You must perform all scanning manually, although you can still run it from the command line, and thus script and schedule it. Scanning files is easy. Simply use the main window to select the file or folder you want to scan and click **Scan**. The four icons across the top, in order from left to right, are for accessing preferences, Web-based Virus Signature Updates, scan memory, and scanning files or folders (the same as the **Scan** button).

Figure 3.22 ClamWin



By clicking the **Preferences** button, you can open another window and configure many ClamWin options (as shown in Figure 3.23).

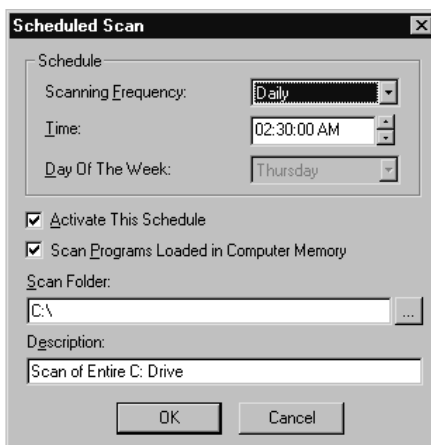
Figure 3.23 ClamWin Preferences



You probably will be most interested in the Internet Updates, Scheduled Scans, and Email Alerts tabs. Internet Updates allows you to specify when ClamWin should update itself, and how often. A daily update in the middle of the night is usually adequate. Click on the **Scheduled Scans** tab and you can add a scanning process to the scheduler. To do so, click **Add** to open the

Scheduled Scan window shown in Figure 3.24. Set the **Scanning Frequency** and **Time**. Make sure that **Activate This schedule** is checked. Select a **Scan Folder** and, optionally, a **Description**, and click **OK**. I recommend a scan time when the machines will be on, but hopefully not in use, so that the scan does not slow down the system's responsiveness. If the systems are routinely left running all night, this shouldn't be an issue.

Figure 3.24 ClamWin Scheduled Scan



You can schedule e-mail alerts for when a virus is found, by using the **Email Alerts** tab. The configuration is straightforward. Enter the SMTP server name and, if required, your username and password. You can send a test e-mail using the button at the bottom. When you are finished configuring the preferences, click **OK** to accept the settings. You can view the reports ClamWin generates by navigating to **Tools | Display Reports** and then selecting the **Virus Database Update Report** or the **Scan Report**.

While ClamWin lacks on-access scanning, which *is* supported with virtually every commercial antivirus product, the integrated scheduling functionality and automated e-mail alerts are very nice features in a free product. The ClamWin virus signature database is updated several times a day, so you have the ability to keep your signatures updated without any manual configuration. If you are trying to keep your network secure without spending a lot of cash, you should definitely consider ClamWin.

Using Online Virus Scanners

Another option is to use any of several free online virus scanners. These are not as hands-off as installing a product is, because you need to go to the Web site and initiate a scan. This process will also require installation of various browser plug-ins to function. Although many Web sites offer free online scans, many actually require you to download software, which isn't really an "online" scan. Trend Micro offers a true online scan using HouseCall, from <http://housecall.trendmicro.com>. When you run the scan, HouseCall will allow you to choose between performing the scan using a Java-based application or via an ActiveX browser plug-in. Loading HouseCall will take some time. Once HouseCall is completely initialized, you can choose to scan the complete computer, or selected files and folders. If you select the entire computer, there are no other options to configure and the scan will commence. If you choose selected files and folders, you will be presented with a tree view to choose which folders to scan. After making your selection, click **Next**. One word of caution: Stick with reputable antivirus vendors' Web sites for online scans; otherwise, you could fall victim to a hacker posing as an online virus scanner.

Antispyware Software

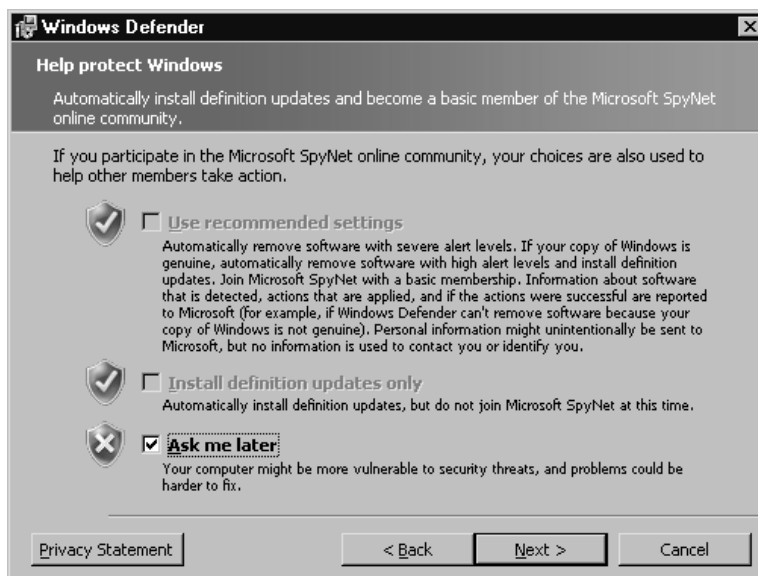
Spyware can cause a lot more problems than most people realize. Even if a corporate firewall prevents the software from ever revealing any confidential information, the processes that are running can rob the system of valuable resources. Right off the top, these programs will consume processor cycles, and with enough of them, or with a poorly written program, the impact can be substantial. Also, the running spyware will consume some amount of memory. In addition to these resources, the spyware will consume *some* amount of disk space, though this is typically the least of your worries. All these are present even if the spyware in question is working perfectly. The truth is this is rare. In most cases, the people installing the spyware don't care whether your system crashes, and these programs are rarely tested adequately. The source of many unexplained problems is often spyware. Luckily, many free antispyware utilities are available. Take note that the same licensing caveats that apply to antivirus software (and all free software, really) apply to antispy-

ware software. In many instances, the default business model is to offer a feature-limited version for free, in an attempt to convince you to buy the commercial version. In most of these cases, the license agreement will expressly exclude installation in a business environment. Be sure to review the license carefully and ensure that you are using the software legally. When in doubt, consult your legal team.

Microsoft Windows Defender

Microsoft Windows Defender is a relatively new offering which will attempt to block and defend you from spyware and other malware. You can read about it and download Windows Defender from www.microsoft.com/athome/security/spyware/software/default.mspx. Unfortunately, Windows Defender will run on only Windows XP SP2 or Windows 2003 SP1. Windows Defender does include real-time protection, which few of the other “personal” antispyware products include. You can download and run the installation directly from the Microsoft Web site. During the installation, you will be asked whether you want to join the Microsoft SpyNet online community via the window shown in Figure 3.25.

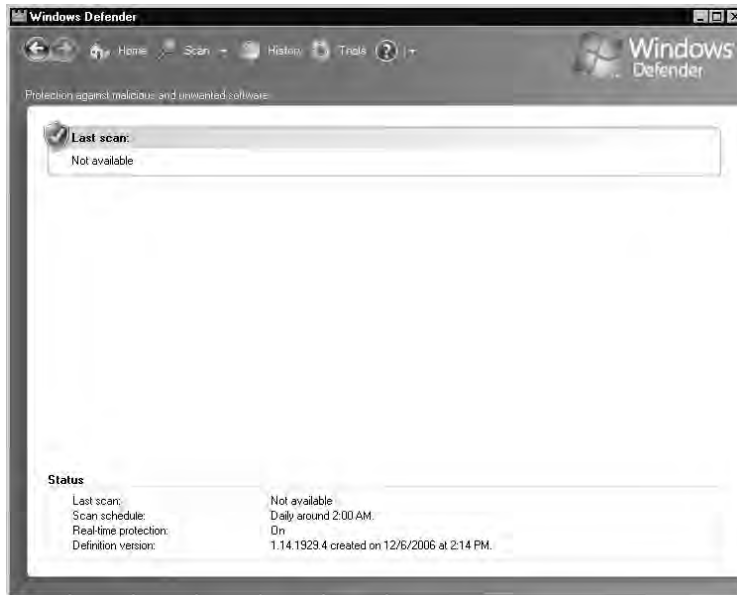
Figure 3.25 Microsoft SpyNet Community



A brief description is provided next to each choice. SpyNet is a system whereby the actions users take collectively form a profile of a given program to help determine whether it is malicious. Participating in this will mean that some nonuser-specific data will be sent back to Microsoft. If you don't want to send any data, the more conservative option is to select **Ask me later**. If you want to participate in SpyNet, select the top option. The option in the middle allows you to update your spyware signatures without joining SpyNet. After making your selection, click **Next**. Choose between a complete install and a custom one. If you're like me, you will almost always click **Custom**, just because you want to see what the options are. There really aren't any, other than choosing the installation directory. Click **Next** and then **Install**. When the installation is complete, you can click **Finish**. You should leave the checkbox selected to **Check for updated definitions and run a quick scan now**.

The main Windows Defender window will open (see Figure 3.26).

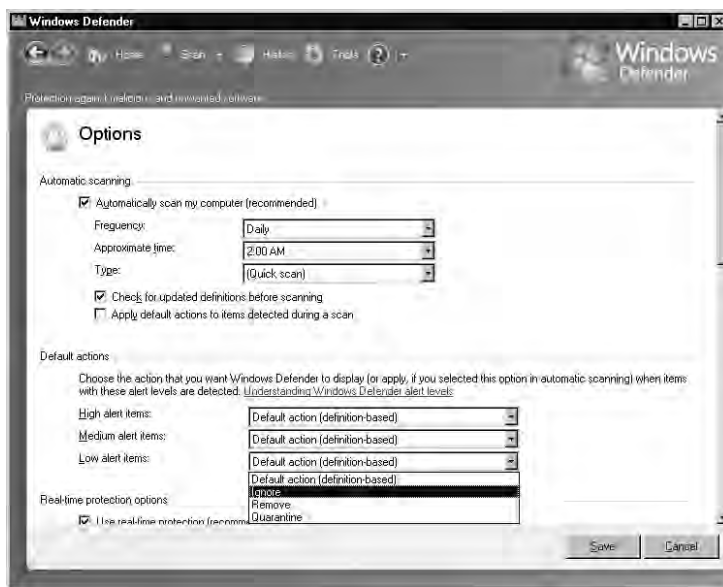
Figure 3.26 Windows Defender



Windows Defender has some nice features. If you select **Tools** at the top, the window shows several icons for different functions. Selecting **Options** will allow you to configure the Windows Defender settings. The options are

not exhaustive, but they are functional. You can choose to enable automatic scanning (enabled by default daily at 2 A.M.). You can also choose for the automatic scan to be a quick scan or a full scan. Most of the default selections will probably be appropriate for most users. Default Actions is a critical configuration area. This determines how invasive you want Windows Defender to be. You can choose available actions for detected items which are high-, medium-, or low-risk. Each one offers the same options: **Default action** (which is defined by Microsoft's signature database), **Ignore**, **Remove**, and **Quarantine**, as shown in Figure 3.27.

Figure 3.27 Windows Defender Options



By allowing Microsoft to use the default action, you are trusting that the company will know what's best. If you want to ensure that nothing critical can be accidentally deleted you can change all actions to **Ignore**.

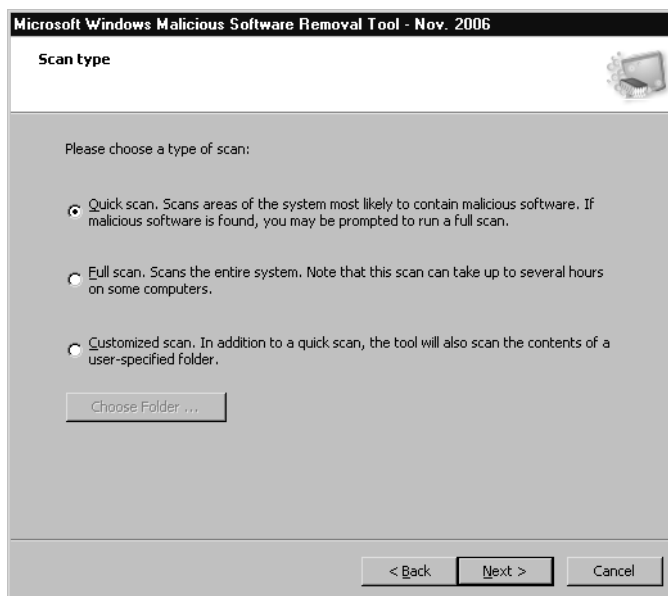
Another interesting window is under **Tools | Software Explorer**. This is a useful screen that will give you information on all running processes, just startup processes (the default view), programs using the network, or Winsock service providers. These can be very informative lists to see what is happening on your current system.

Microsoft Malicious Software Removal Tool

The Microsoft Malicious Software Removal Tool is a specialized utility designed to help you remove a specific subset of the most damaging viruses and worms. Because some of these software packages can be both very damaging to your system and very hard to remove, this tool was designed to help. The goal of the Microsoft Windows Malicious Software Removal Tool is to remove the offending software while minimizing any lasting damage to existing files. This utility does not offer any type of ongoing protection; it is purely intended to help you clean up and recover after an infection has occurred. You can download and read about the tool at www.microsoft.com/security/malwareremove/default.aspx. To see a list of the malicious software that the tool checks for, refer to the knowledge base article located at <http://support.microsoft.com/?kbid=890830>.

Running the tool is easy. Just download the tool and run the downloaded executable. You will be prompted to choose the type of scan you want to run, as shown in Figure 3.28.

Figure 3.28 Microsoft Malicious Software Removal Tool



After making your selection, click **Next** and the scan will begin. A quick scan really is very quick, and the next window you see will be the **Scan Results** window. That's really all there is to it, unless the tool detects something, in which case you will be prompted to determine what action you want to take.

Encrypting Sensitive Data

Another way to protect your sensitive data is through encryption. In theory, even if logical controls are compromised and bypassed, an encryption mechanism can still keep data out of a hacker's hands. There are some critical caveats to employing any form of data encryption, whether for personal use or within an organization. You should never encrypt something without having a means to decrypt the data in an emergency. You typically would do this through a password escrow mechanism, whereby a central repository contains all the needed passwords to decrypt the data.

The objective is to ensure that the encryption does not become a liability. This can occur in a surprising number of ways. An employee who has the password for critical encrypted data could quit or become unreachable. A malicious employee could intentionally encrypt important data, or a criminal could even use your encryption mechanism to encrypt critical data and then offer the password up for ransom. If the encryption protocols and methods are truly secure, the data will be as unrecoverable for you as it would be for any other attacker. The requirement for a password recovery mechanism exists whether you are implementing encryption that is supported natively by the operating system, or in the form of third-party software. This warning is so important that many security experts advise *disabling* the native Windows encryption (encrypted file system, or EFS) if you do not implement a means (of which there are several) to recover encrypted data (see the next section for more information).

You should also consider the types of encryption you need. Encrypting single files is a convenient way to secure specific files on an as-needed basis. Another alternative is to create a normal file, which you can access as though it were an entire disk volume. The volume is integrated into your operating system and appears as another drive letter. Any files placed on this virtual disk

will be encrypted automatically. This type of encryption can be useful for large numbers of frequently changing files. The native Windows encryption is a sort of hybrid solution, encrypting and decrypting data files automatically using your Windows credentials. You may not need to use all of these methods of encryption, or you may benefit from a combination of methods. Ultimately, one of the most common determining factors is ease of use. An encryption method that is easy to use will probably be used far more consistently than one that is hard to use. We will review some of these methods in the next section in order to help you make an informed decision.

EFS

Microsoft's EFS really shines when it comes to ease of use. Once you configure it, there are no additional passwords to remember (other than your Windows login credentials), and you don't need to manually encrypt or decrypt anything. EFS was introduced with Windows 2000, and you can implement it only on Windows 2000 or newer. With Windows 2000, a recovery policy was required in order to implement EFS. This allowed for an alternative means to recover encrypted data, in addition to the data being unencrypted by the original user who encrypted it. Without a recovery policy defined, EFS would simply not work. With Windows XP and 2003, this requirement was removed, meaning that if EFS is enabled and there is no recovery policy, a user could encrypt data that was not recoverable by the organization. In most cases IT policy is worded in such a way that any data the user might store belongs to the organization, and therefore, the organization has a right to ensure that the data is recoverable. Because the default data recovery agent is the domain administrator, the risk of data becoming unrecoverable is particularly high with stand-alone workstations.

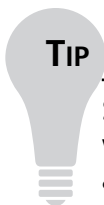
Because EFS is enabled by default, it's easy to encrypt an individual file or folder. Select the file you want to encrypt, right-click it, and select **Properties**. Click **Advanced** and check the box next to **Encrypt contents to secure data**, as shown in Figure 3.29.

Figure 3.29 Windows EFS Encryption



Afterward, click **OK**, and then **OK** again. You will receive an encryption warning, which alerts you that the folder the file resides in is not an encrypted folder. You will need to click **OK** to close the warning. Under these circumstances, you can edit the file and close it and it will remain encrypted, but if you use **Save As** and rename the file, it will no longer remain encrypted. This opens up the possibility that a user will either forget to encrypt the file after modifying it, or neglect it due to the inconvenience of doing it every time a filename is changed due to versioning or other factors. This is why it is recommended that you set entire folders as encrypted, instead of individual files. The process is the same, except you select a folder instead of an individual file. After you do this, all files placed in the folder will be encrypted, even ones moved or copied from other folders. This makes encryption painless to use.

A significant disadvantage to using EFS is that it does not support access to encrypted content based on group membership. This means that once a user encrypts the data, only that user can encrypt the data, or individual users manually given access on a per-file basis (Windows XP and newer only). This makes EFS unsuitable, or at least cumbersome, for files that are shared among a group of users. To configure additional users to access a specific encrypted file, open the **Advanced Properties** for the file and click **Details**. You can then click **Add** to add an individual user to be able to access that encrypted file.

**TIP**

Some applications will remove the certificate information associated with a file. This certificate information is used to allow multiple users to access the file. As a result, all the additional users that were granted access to the encrypted file individually are removed.

With Windows 2000, if the administrator changed the password for a user account, that user could still access all of his encrypted content. With Windows XP, that is no longer true. Instead, the user will lose all access to his previously encrypted files. The only way to retrieve the encrypted data is to use the data recovery agent. Stand-alone XP workstations with no recovery policy defined (and thus, no data recovery agents defined) make it very easy to permanently lose access to your encrypted data. If you elect to disable EFS entirely, you can do so via local GPOs on individual computers. You can also disable EFS via the domain GPOs, though if the workstations are part of a domain, you could instead define the data recovery agent and thus eliminate the risk of lost data in the first place. You could even enable EFS for specific subsets of systems based on a GPO applied to an OU. To do so, navigate to \Computer Configuration\Windows Settings\Security Settings\Public Key Policies\Encrypting File System within the appropriate policy snap-in (local or domain). Right-click **Encrypting File System** and select **Properties**. You will see a single **General** tab, with a single checkbox. Uncheck the checkbox next to **Allow users to encrypt file using Encrypting File System**.

If you elect to use EFS, you must have a data recovery agent. Windows XP does not create one by default. The recovery policy defines the data recovery agent to be used. The recovery policy is inherited from the domain for machines that are members of a domain. For stand-alone machines, you must create the recovery policy and data recovery agent manually. To manually create the recovery policy and assign the data recovery agent on XP machines, follow these steps:

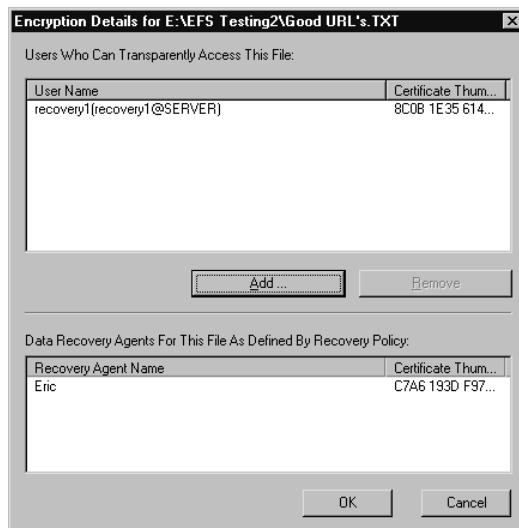
1. Using the credentials of the desired data recovery agent, enter **cipher /r:<filename>** to generate encryption keys. In this example, I generated the keys while logged in as the user Eric. You will be asked to

provide a password and verify the password in order to complete the process. This will generate two files, named filename.cer and filename.pfx.

2. Navigate within the Local Computer Policy snap-in to \Computer Configuration\Windows Settings\Security Settings\Public Key Policies\Encrypting File System. Right-click **Encrypting File System** and select **Add Data Recovery Agent**.
3. Click **Next** on the welcome screen. On the **Select Recovery Agents** screen, click **Browse Folders**, select the .CER file you created previously, and click **Open**.
4. Click **Next**, and then **Finish**.

Now you can use this certificate to decrypt data on that local machine, regardless of which user account encrypted it. You can see that the data recovery agent's key was incorporated into the encrypted file by again viewing the **Encryption Details**, as shown in Figure 3.30.

Figure 3.30 Encryption Details with the Data Recovery Agent



You can see that the certificate we created earlier and configured in the recovery policy is now listed as the data recovery agent. To recover data using this certificate, follow these steps.

1. Right-click the file you want to decrypt (in my example, the file was created by a user named recovery1) and select **Properties**.
2. Click **Advanced** and then remove the checkmark next to **Encrypt contents to secure data**.
3. Click **OK** and **OK** again to close the **Properties** window.

The files will no longer be encrypted. The designated recovery agent can also simply open the files in question; however, they will remain encrypted when the data recovery agent closes them. The recovery agent will only have access to decrypt files encrypted *after* the policy was configured. If files were encrypted prior to defining the data recovery agent, you can update the encryption on all files using the *cipher /u* command. Note that this command will be able to update only the files you have access to. If you had three users with files that were encrypted prior to the data recovery agent being defined, all three would need to run the command to update their encryption keys with the data recovery agent information.

Cipher.exe also has several other uses. You can use it to encrypt a file or folder from the command line, or to display a listing of the encryption status of a directory and its contents. *Cipher* with no options will list the files and folders in the current directory and show which are encrypted. A *U* indicates that the file or folder is unencrypted, and an *E* indicates that it is encrypted:



```
E:\>cipher
```

```
Listing E:\
```

```
New files added to this directory will not be encrypted.
```

```
U Burned to CD
```

```
U Docs
```

```
U backup Files
```

```
E EFS_ERIC
```

```
E EFS_RECOVERY1
```

```
E EFS_RECOVERY2
```

To encrypt a file, use *cipher* with the */a* option. You can encrypt a directory with */a* (which will not mark the directory as encrypted) or */e* (which will mark the directory to be encrypted). To decrypt a directory, use the */d*

option. To decrypt a file in a directory, you must use `/d /a <filename>`, which will decrypt the specified file:

```
SYN@RESS
syngress.com
E:\>cipher /D EFS_RECOVERY1
Decrypting directories in E:\
EFS_RECOVERY1 [OK]
1 directorie(s) within 1 directorie(s) were decrypted.
```

Here, I have included examples of encrypting and then decrypting a file, with a plain listing from *cipher* to illustrate the file's status, to clarify the use of the switches. I have removed blank lines to conserve space:

```
SYN@RESS
syngress.com
E:\EFS_RECOVERY1>cipher
Listing E:\EFS_RECOVERY1\
New files added to this directory will not be encrypted.
U test.TXT

E:\EFS_RECOVERY1>cipher /e /a test.txt
Encrypting files in E:\EFS_RECOVERY1\
test.TXT [OK]
1 file(s) [or directorie(s)] within 1 directorie(s) were encrypted.
```

Converting files from plaintext to ciphertext may leave sections of old plaintext on the disk volume(s). It is recommended to use command `CIPHER /W:directory` to clean up the disk after all converting is done.

```
E:\EFS_RECOVERY1>cipher
Listing E:\EFS_RECOVERY1\
New files added to this directory will not be encrypted.
E test.TXT

E:\EFS_RECOVERY1>cipher /d /a test.txt
Decrypting files in E:\EFS_RECOVERY1\
test.TXT [OK]
1 file(s) [or directorie(s)] within 1 directorie(s) were decrypted.
```

```
E:\EFS_RECOVERY1>cipher
Listing E:\EFS_RECOVERY1\
New files added to this directory will not be encrypted.
U test.TXT
```

A final, very useful function you can get from *cipher* is the */w* option. You can use the */w* option to remove remnant data from “empty” portions of the hard disk. When you delete a file, only the pointer to the file in the file allocation table is really removed. The actual data from the file stays right where it was on the hard disk, until something else randomly comes along and needs that particular piece of disk real estate. The various programs designed to “undelete” files work by scanning the disk and finding the original data, and then rebuilding the file allocation table entry for it. The */w* option of *cipher* will intentionally overwrite all unallocated disk space to ensure that files cannot be undeleted. You can find additional information on the Encrypting File System at www.microsoft.com/technet/security/guidance/cryptographyetc/efs.msp.

Summary

Many security solutions are available for protecting your network resources. You should perform basic hardening on all network resources, from workstations, to servers, to routers and switches. After basic hardening, regular patches and updates are required to ensure that you are running the most secure software possible. Personal firewalls can serve as your first line of defense, providing logical access control based on the contents of network packets. After the network is secured, your defenses turn inward, to protecting your assets from such things as viruses and spyware. Finally, we wrapped up with the last line of defense—file encryption—to protect sensitive data even if the system or encrypted files should fall into the wrong hands.

Solutions Fast Track

Performing Basic Hardening

- ☑ Have an IT security policy in place so that you have a concrete set of standards against which to measure. Having specific objectives in mind when you are hardening systems will increase the odds of you achieving those objectives.
- ☑ Requiring high-quality passwords is a balancing act between requiring a good password and making the requirements so difficult that the user ends up writing down the password in order to remember it.

Hardening Windows Systems

- ☑ Renaming the administrator account and disabling the user account are simple and effective security measures.
- ☑ Utilizing GPOs is a means to provide centrally managed security configurations to all of your protected hosts.

Hardening Linux Systems

- ☑ Understand how file system permissions work in order to lock down access to only the minimum that is required.
- ☑ Employ TCP Wrappers to restrict network access based on applications (daemons) instead of based on network ports or IP addresses.

Hardening Infrastructure Devices

- ☑ You must perform the same high-level tasks for infrastructure devices as for any other workstation or server.
- ☑ Remember to disable the unneeded services that most managed infrastructure devices will enable by default.

Patching Systems

- ☑ Remember that a controlled patch management methodology does not mean applying all the patches the instant they are released. You should conduct proper research and testing before making any changes to a system's software.
- ☑ Most modern operating systems (Linux and Windows) include tools to automatically check for software updates online and apply them if desired.

Personal Firewalls

- ☑ Pay careful attention to the license agreements and make sure you are not using a great “personal” firewall illegally.
- ☑ Windows Firewall provides adequate controls for what types of access are permitted, but it only really is suitable as a personal firewall, not as a network perimeter firewall.

Providing Antivirus and Antispyware Protection

- ☑ Free antivirus and antispyware software often prohibit use in a business setting. Verify that the license agreement allows you to deploy the software in your environment.
- ☑ Both antivirus and antispyware software require regular patching with the current signature files in order to remain effective.

Encrypting Sensitive Data

- ☑ You should use EFS only if a data recovery agent has been configured and enforced via group policy. Using EFS without a data recovery agent runs the risk of permanently losing access to encrypted data if the original user who encrypted it becomes corrupt or unavailable.
- ☑ Consider carefully the ease of the encryption methods you choose. A high-quality encryption program that is hard to use probably will go unused.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: Is there any way to automate the live Web-based virus scans?

A: While you can do it, such a project would likely be very time-intensive. You could start with the set of tools called libwww, from www.w3.org/Library/Distribution.html. This includes tools for command-line HTTP tools for retrieving Web sites, and even sending input back. You could probably patch together a script that would perform the scan, though again, I don't recommend it unless you really need that functionality desperately.

Q: I haven't had a virus or any spyware before. Do I really need to worry about installing any software to protect myself?

A: First, I would encourage you to run some scanners. Many systems are infected with a virus or spyware without it causing enough of a problem for anyone to notice. Second, even the most well-informed users can sometimes make mistakes, and with no protection, it will take only one mistake to potentially bring your entire network to its knees. If you want to see how easy it is to fall victim, try this. Burn a CD-ROM and configure the autorun file to execute a program that will e-mail you when run. Do the same with an executable on the disk. Give the files interesting-sounding names, such as quotes, incomes, merchants, or discounts. When finished, leave the unlabeled CD-ROM lying around somewhere near your organization's building. Sit back and see how long it takes before you receive your e-mail. All it takes is one person getting curious and putting the disk in the drive.

Q: How can I disable the configuration of Windows Firewall so that my users can't turn it off?

- A:** You can read about the process of configuring Windows Firewall using GPOs in more detail here: www.microsoft.com/technet/security/small-business/prodtech/windowsxp/fwgrppol.mspx. The appropriate settings are located in the policy under Computer Configuration\Administrative Templates\Network\Network Connections\Windows Firewall. When your configuration changes are completed, you can apply the policy by running `gpupdate`, rebooting the machine, or waiting for the default policy refresh to occur.
- Q:** What happens if I'm using EFS and copy an encrypted file to my USB drive? Will it remain encrypted, or does Windows decrypt it before copying it out?
- A:** It depends on a lot of factors. If the destination file system is not formatted as NTFS, the file will be decrypted and copied. If you are using Windows 2000 EFS, this decryption takes place transparently, so you may not be aware that the file was decrypted. With newer versions of Windows, you will receive a warning reminding you that the file will be decrypted. Without getting too long-winded, EFS behaves differently depending on (1.) whether you are doing a move or a copy operation; (2.) the way the destination file system is formatted; and (3.) what OS and directory access the destination operating system has.
- Q:** I'm a home user. Will Windows Firewall protect my host if I plug it in directly to my cable modem?
- A:** Placing your system directly on the Internet with no firewall at all ensures that the system will be compromised and used for some purposes you probably wouldn't approve of—possibly without you even knowing. Enabling Windows Firewall is a big improvement over no firewall and will offer some protection. A real firewall (even an inexpensive home model) offers additional security that Windows Firewall doesn't. This includes source NAT for one, which will help hide your machines real IP address and make directing attacks toward it more difficult. So will Windows Firewall protect you? Sure, to a degree, but remember defense in depth is the objective.

Q: How long should I wait before applying patches to my protected hosts?

A: This is really going to depend on your environment, and the requirements will likely be different for different resources within your environment. The development servers probably deserve a different patching schedule than the critical Web portal that makes the majority of company revenue. Another important determining factor will be the severity of the issue(s) the patch addresses. If the patch closes a remotely exploitable security hole, applying it will be more urgent than if the patch addresses a vulnerability that can only be exploited while someone is logged in to the machine locally.

Configuring an Intrusion Detection System

Solutions in this chapter:

- Intrusion Detection Systems
- Configuring an Intrusion Detection System
- Configuring Snort on a Windows System
- Configuring Snort on a Linux System
- Other Snort Add-Ons
- Demonstrating Effectiveness

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

No matter how secure your network is, sooner or later something will happen that wasn't supposed to. If you don't take steps to identify security events, your only notice that something has occurred might be when there is a production outage or other undesirable disruption. At that point you are in a reactive mode instead of a proactive one. It would be preferable to know the instant a security event takes place so that you have a head start on correcting the issue and minimizing any damage. The two most common ways to keep yourself informed of security events across your network are through intrusion detection systems (IDSes) and by monitoring event logs. In this chapter we will demonstrate how to install and configure a first-class IDS on both Linux and Windows systems, and we will discuss the various tools for managing event logs, including syslog and Windows event log formats.

Intrusion Detection Systems

Intrusion detection systems do exactly what it sounds like. They spot undesirable activity and, typically, send an alert to someone so that action can be taken. The undesirable activity does not necessarily have to be from an actual intrusion, it can be any activity that you don't want to occur, such as the use of a file sharing program on the corporate network. The most common way to implement an IDS is by having a system monitor and inspect (sniff) all traffic over a given link. The system then compares the traffic with a database of known signatures for undesirable traffic. This is an example of a *signature-based IDS*. Some IDS systems are *anomaly based*, which means the IDS attempts to build a list of "normal" traffic from your actual network data, and then it flags anything that doesn't fit the list of normal traffic that it has built. The process of identifying the normal traffic can be very time consuming and requires a lot of human intervention and judgment. Some systems combine the two methods.

Anomaly based and signature based are both terms that describe how the undesirable traffic is identified. In addition to this, there are also different terms to describe *where* the IDS is doing the monitoring. A *host-based IDS* (HIDS) inspects the activities on a particular system. In this way, you can monitor your key servers, such as a Web server, and look for attempts to brute

force a password. A *network-based IDS* (NIDS) inspects the actual packets flowing across the network. The difference in methodologies between the two is significant. Sometimes malicious traffic at the host level may be indistinguishable from legitimate traffic at the network level.

As a final bit of vocabulary it's worth mentioning *intrusion prevention systems* (IPS). An IDS serves only to *detect* the malicious activities; it then notifies someone so that human judgment can be applied and the needed steps taken in response to the malicious activities. An IPS is more powerful in that in addition to merely detecting malicious activities, it has the capability to take action on its own to prevent the malicious activity from being successful. As an example, an IDS can detect a user who uses ICQ messenger on the corporate network. If configured to do so, the IDS can send an e-mail, or even page the network administrator who can take any needed action. The IDS could also merely log the occurrence to be included in a report at the end of the month. At its most invasive, the IPS can, upon detecting the connection, send a TCP RESET packet to the ICQ server and tell it to terminate the connection. The user would only see that the application would not work properly.

Such authority to take direct action without human intervention may sound appealing, but such capabilities must be used with extreme caution. A packet could be generated with a false source IP, to make it look like another machine is trying to connect using ICQ, and a reset packet would dutifully be sent to the incorrect machine. A little creativity and a malicious user could turn your IPS against you and cause problems on your network. The real drawback is that no matter how well programmed it is, the IPS lacks human judgment. For this reason, when an IPS is implemented, it is generally done very gradually and with very conservative settings. There are Snort modules that provide IPS functionality, but we won't be covering those in this book.

Configuring an Intrusion Detection System

When it comes to IDSes, there really isn't a very large playing field for free solutions. That's not to say that there isn't an excellent free IDS, just that there is only one, Snort (<http://snort.org/>). Snort is an excellent, free, signature-based NIDS. You can take advantage of a subscription service, which enables

you to download newer signatures as they are released, but it is not a requirement. You can also create your own rules to match on various portions of the IP packet relatively easily. The configuration of Snort is a little different when it's installed on Linux versus when it's installed on Windows, so I will demonstrate both. If you would like to pursue more in-depth reading on Snort see *Snort Intrusion Detection and Prevention Toolkit* (Syngress Publishing, 2006).

Hardware Requirements

The prevailing theme in this book is to improve security without requiring a direct purchase of anything. In the majority of cases there will be an old PC lying around somewhere to use. In the case of an IDS, this device will need to see traffic at critical chokepoints in your network, possibly placing the IDS host, or an interface on the host, outside the protected perimeter. For this reason, the IDS hosts are often dedicated systems. You would need to weigh carefully if you planned on using the IDS host as a normal workstation in addition to performing the IDS function.

The point of all this is to touch on the fact that with your IDS, probably more than with any other system in this book, you will need to ensure you have hardware with adequate resource to do its job. Most of the tools mentioned here are simply other processes that could run on almost any system and usually require very few resources. The IDS has the potential to log a large amount of data and process a lot of network traffic. These functions will require both memory and hard drive space. This doesn't mean you can't use an old PC as your IDS. It only means that you want to give your IDS (and your firewalls) the "best" of the old hardware you have available. After you implement the IDS, monitor the system resources closely until you gain a level of comfort with the resources that are required.

Placing Your NIDS

When it comes to implementing an NIDS, the single biggest factor in its effectiveness is its placement within the network. The value of the NIDS is in identifying malicious traffic and obviously it can't do that if it can't see the traffic. This means you want to place the NIDS in a location to maximize the data it will see. In a very small office where there is only one switch or hub, this is a pretty simple decision. Depending on your objectives, you may place

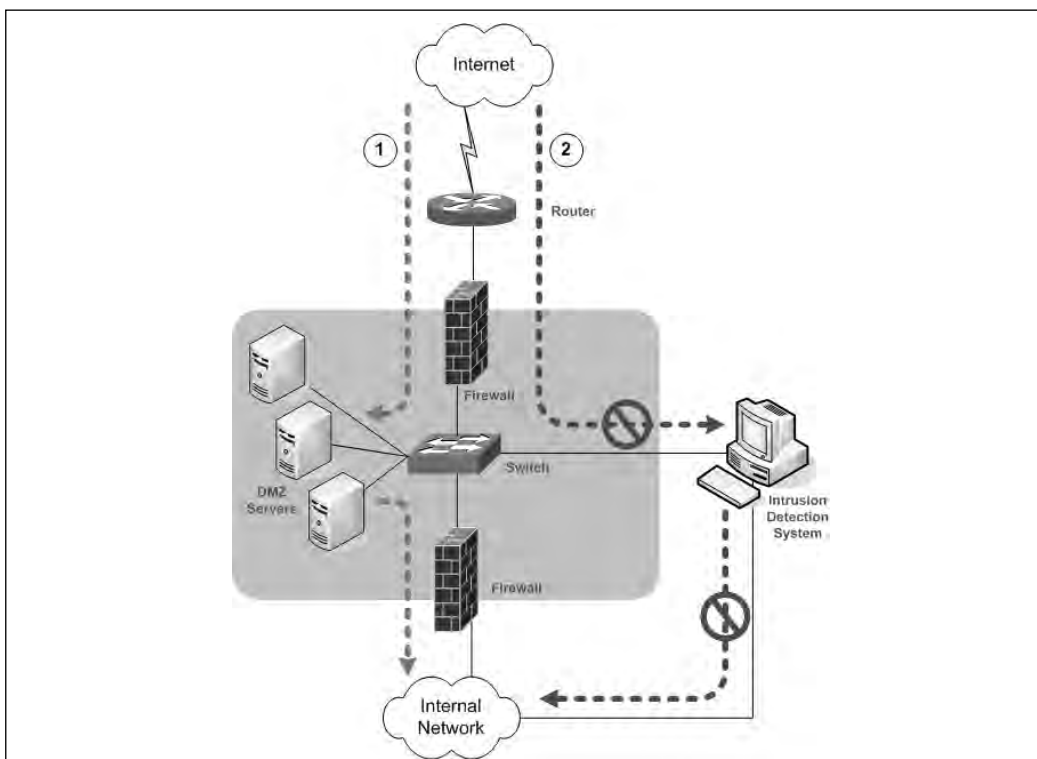
it inline with the Internet connection only, so that you are inspecting traffic only to or from the Internet. In a larger installation, you may need to place multiple network cards in the NIDS so that it can inspect traffic from several chokepoints in your network. Also see Chapter 7, “Network Reporting and Troubleshooting,” for additional discussions on sensor placement.

Notes from the Underground...

Further Considerations

Remember that an IDS is also a target for a hacker just like any other system, and often even more so. As such, the IDS host system should be hardened and locked down as much as possible. In addition to being a target because it can alert administrators to their activities, the hacker might target the IDS system itself because it often contains logs with valuable information in it about various systems. The IDS also has the capability of capturing packets that match its rulebase, and these packet dumps can contain valuable data as well. Don't neglect securing your IDS or you may be creating a security liability instead of the asset you intended.

Be cognizant of the fact that if you do choose to install multiple network cards to monitor multiple segments that you have the potential to create an alternate data path that enables traffic to bypass a firewall. As part of your hardening of the IDS host, you must ensure that routing is not enabled so that the IDS cannot forward traffic from one segment that it is monitoring to another. There are multiple approaches to protect against this happening. The simplest is to use a network *tap* instead of just plugging in a normal network card. A tap is a specially designed piece of hardware that will only listen to traffic but will not transmit. Because it is hardware, there is no possibility of hacking the configuration or making a mistake in the configuration and accidentally allowing routing. Unfortunately, network taps are not free. Disabling routing, ensuring the host has no static routes, and disabling any routing protocols are the free ways to ensure that you don't create a path around a firewall. Figure 4.1 illustrates bypassing the firewalls using your IDS.

Figure 4.1 Bypassing the Firewalls Using the IDS

The first dotted line (data flow #1) represents the desired (secure) data flow. Traffic from outside can only terminate on a server in the DMZ, and traffic going into the internal network can only come from a server in the DMZ. With this configuration traffic from the Internet can never pass all the way through directly to a host on the internal network. The second data flow, #2, represents how an *incorrectly configured* IDS could be used to route traffic from the outside (untrusted) network into the internal network.

When it comes to placement of your IDS, you need to be aware of the difference between a switch and a hub. A hub operates by sending any traffic it receives on any port to every other port. Therefore, when using a hub, the IDS will see all the traffic passing through that hub, which is usually what you want for your IDS. A switch is more advanced than a hub, and most new devices are switches. A switch listens and learns what machines are connected to which port. It then uses this information to construct a forwarding table.

After it has learned which port a given host is on, it will then only send traffic destined for that host to that specific port. This means that without any additional configuration, when you plug your IDS into a switch port, it isn't going to be seeing much traffic.

Luckily, there are some options for getting around this feature. Most enterprise switches (that is, the more expensive ones) have a port mirroring option. The terms used to describe this functionality varies from one manufacturer to another, Cisco calls it *Switched Port Analyzer* (SPAN). This enables you to configure a specific port such that it will see traffic from other designated ports (or all other ports) even though the traffic is destined for a different port. Typically, one port is configured to mirror all other ports, and the IDS is attached to this port. On a Cisco 3750 switch with 24 ports you could configure mirroring by entering the following commands:



```
switch(config)# monitor session 1 source interface gig1/0/1 - gig1/0/23
switch(config)# monitor session 1 destination interface gig1/0/24
switch(config)# end
```

This setup is straightforward. Line one specifies which ports to forward traffic from, and line two specifies which port the traffic should be mirrored to. You will need to refer to the user guide for your specific switch hardware to see if port mirroring is supported, and if it is, how to configure it.

Configuring Snort on a Windows System

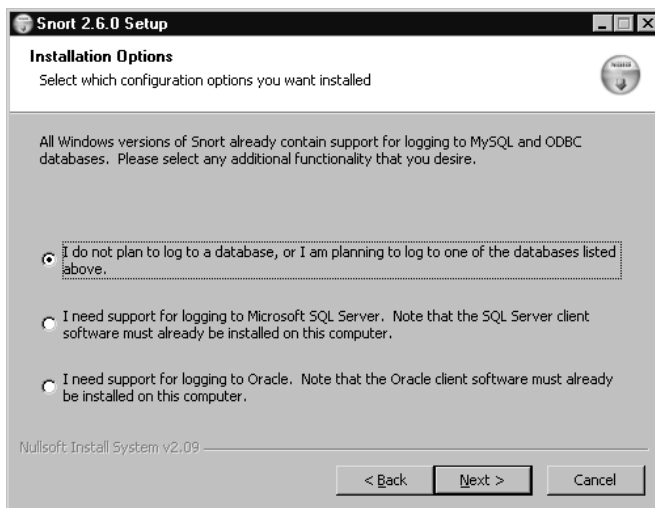
From the start, the developers of Snort wanted it to be available on a wide variety of platforms. The current version will run on Linux, UNIX, Windows, and Macintosh OSX. This is good since you will probably be running your IDS on whatever system happens to be lying around. There are some caveats to be aware of when running Snort on Windows. For one, the documentation is very *nix-centric. Many times what is referred to as the “default” behavior is not the default for Windows Snort. The second, more noticeable difference is that getting Snort up and running on Windows is actually more difficult than setting it up to run on Linux. Still, if your comfort level is much higher on Windows (as it is for many), then Windows still may be the best choice. For my example I installed it on Windows Server 2003 with all the latest updates applied.

Installing Snort

Begin by browsing to <http://snort.org/> and clicking on the **Get Snort** link on the left-hand side of the Web page. Click on **Binaries**, then **Win32**, and download the latest Installer file. When this is done, navigate to the file you downloaded and double-click it to start the install process.

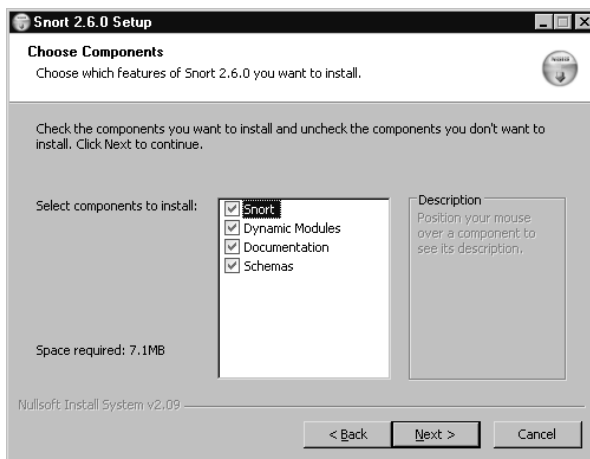
1. You must click **I Agree** on the **License Agreement** window to proceed with the installation.
2. The next screen enables you to configure support for oracle or SQL server logging (see Figure 4.2). MySQL and ODBC are already supported by default. For a smaller installation the (first) default option will usually be adequate. After making your selection, click **Next**.

Figure 4.2 Snort Setup Logging Options



3. On the **Choose Components** screen shown in Figure 4.3, you should probably select the default, which is to install all components. The schemas are needed only if you plan to log to a database; however, the full install is only about 7 MB, so there isn't much space to be gained by trying to trim down the install. After making your selections, click **Next**.

Figure 4.3 Choose Components for Snort



4. The next screen enables you to choose your installation location. The default is C:\snort. Remember, this server is a prime target for attackers and should be hardened as much as possible. As a general rule, non-default paths are almost always at least slightly more secure than default ones. After you've selected the installation path, click **Next**.
5. When the Installation has completed, click **Close**.
6. You will see a window, as shown in Figure 4.4, alerting you that Snort requires WinPcap to function and that it can be download from www.winpcap.org.

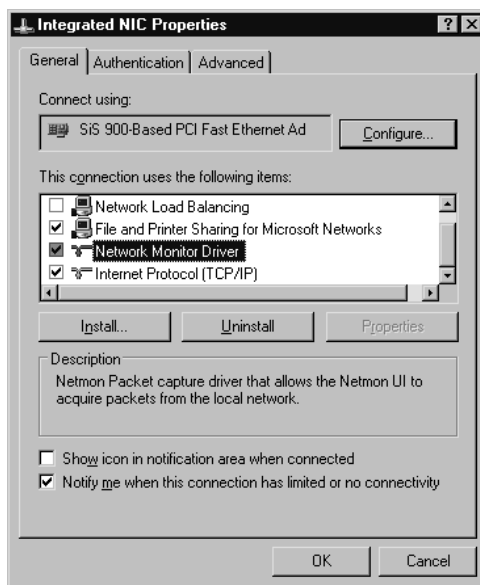
Figure 4.4 WinPcap Reminder



7. WinPcap is basically a Windows version of the UNIX libpcap API. This enables applications to interact with network packets directly, bypassing the Windows protocol stack. You will find WinPcap is required to run many networking tools on Windows. You will need to download WinPcap by clicking **Get WinPcap** on the left side of the Web page.
8. Save the setup file to a location of your choice and double-click it to begin the installation routine.
9. The first screen contains news and update information. Click **Next** to continue.
10. The next window is the License Agreement; you must click **I Agree** to continue the installation.
11. The install will complete. Click **Finish** to close the Installation Wizard.

Navigate to **Start | Control Panel | Network Connections | Local Area Connection**, right-click, and then choose **Properties**. You should see a new network driver in the properties list, as shown in Figure 4.5.

Figure 4.5 Local Area Connection Properties



It would probably be a good idea to test the installation of WinPcap and the packet capture functionality before moving on to configuring Snort, that way if you need to troubleshoot Snort later, you can at least know WinPcap is working. The easiest way to test WinPcap is by starting up WinDump, which is a command-line packet sniffing utility for Windows that uses WinPcap. Windump can be downloaded from www.winpcap.org as well. See Chapter 7, “Network Reporting and Troubleshooting,” for detailed instructions on using WinDump.

Configuring Snort Options

After you have verified that WinPcap is working, it’s time to configure the various options that determine how Snort will behave using the Snort configuration file. The configuration file is excellently documented and very easy to use. The configuration file is divided up into six “steps” annotated within the comments. To get Snort working the way you want it to, follow these simple steps:

1. Start by opening the main Snort configuration file. By default it will be located at **C:\Snort\etc\snort.conf**. If you open it in Notepad it may not display properly, so WordPad would probably be a better choice.
2. Configure the HOME_NET variable, if desired, by removing the # from the line you need. (# is a comment indicator in the Snort configuration file.) The HOME_NET variable defines which networks are the “trusted” internal networks. This is used with the signatures to determine when the internal network is being attacked. By default, HOME_NET is set to any network with the *var HOME_NET any* line in the snort.conf. Setting this to accurately reflect your internal address space will reduce the number of false positive alerts you receive. A common example is *var HOME_NET 192.168.1.0/24*.
3. Configure the EXTERNAL_NET variable if desired. This is the network you expect attacks to come from. The recommended setting is to set this to everything *except* your HOME_NET using the following *var EXTERNAL_NET !\$HOME_NET*. (Default: *var EXTERNAL_NET any*).

4. Next, define what servers are running specific services. For example, by setting `HTTP_SERVERS` to only specific servers, Snort will only watch for HTTP attacks targeted at those servers. If you wish to see attacks targeting servers that are not running the affected services, leave the defaults, which are to watch for attacks directed towards *any* internal servers. (Default: *var DNS_SERVERS \$HOME_NET*). If you had a Web server running on 192.168.1.11 and 192.168.1.12, you could tell Snort to only look for HTTP attacks targeting that server by setting the following variable: *var HTTP_SERVERS [192.168.1.11/32,192.168.1.12/32]*.
5. If desired, configure the specific ports that services are available on. For an example, the default for HTTP is defined on the following line: *var HTTP_PORTS 80*. Similar to defining the servers in the preceding section, this will tell Snort to look only for attacks targeting specific ports. With the default configuration, Snort would ignore an HTTP attack to port 8080.
6. If you are interested in detecting the usage of AOL Instant Messenger (AIM), the various IP addresses of the AIM servers are defined in the `snort.conf` file. This is done because the IP addresses change frequently, and by using a variable, the rules don't have to be updated each time the IP address changes. If you don't wish to trigger based off AIM usage, don't worry about changing these IP addresses.
7. Configure the `RULE_PATH` variable, which tells Snort where to find the rules used for triggering events. This is one of the differences between Snort on Windows and Snort on other operating systems. Most operating systems will use a relative path, which is what is configured by default (*var RULE_PATH ../rules*), but on Windows you should use an absolute path. By default, the path would be *var RULE_PATH C:\snort\rules*.
8. The next section has some commented-out lines to disable certain detections of some infrequently seen types of traffic. Unless you are having some issues with those alerts or your IDS is very low on resources, it's probably fine to just leave those at the default (enabled) configuration.

9. The last few lines of the “step 1” section enable you to configure the detection engine for systems with limited resources. Unless you are having issues, you can leave this option alone.
10. After that the “step 2” and “step 3” sections of the configuration file allow you to enable or disable specific functionality and detect particular types of attack, such as fragmentation attacks, stateful inspection, and stream reassembly options. Delving into the specifics of each of those options is beyond the scope of this chapter and for many, the default settings will serve them well.
11. The section labeled “Step #4” contains output options for Snort. There are several valuable options in this section. Uncomment **output alert_syslog: host=hostname, LOG_AUTH LOG_ALERT** and enter the hostname of your syslog server. LOG_AUTH is the facility to use, and LOG_ALERT is the priority for the alert. In my example I used the following command: *output alert_syslog: host=192.168.1.99, log_local7 log_notice*; this will log to the local7 facility as a notice. Note that not all combinations of facility and severity are supported by Snort. You also need to include the **-s** switch on the command line. We will discuss syslog in more detail in the next chapter. If you don't have a syslog server to log to yet, just make note of the setting and come back to it when your syslog server is set up.
12. Edit the paths for the dynamically loaded libraries in section #2. Edit the lines as follows: **dynamicpreprocessor directory C:\snort\lib\snort_dynamicpreprocessor** and **dynamicengine C:\snort\lib\snort_dynamicengine\sf_engine.dll**. Note that for the preprocessor directory you are editing it for an absolute path (with no trailing slash). For the dynamicengine, you are altering the path from the default libsf_engine.so to the sf_engine.dll used in Windows.
13. Change *include classification.config* to an absolute path such as **include C:\snort\etc\classification.config**. Do the same for **include reference.config**.
14. The include section enables you to specify which rulesets are to be checked. Some rules are disabled by default, such as chat.rules, which are triggered by the use of various instant messaging clients. To enable

or disable a given ruleset, simply add or remove a # at the beginning of the include line. This entry can be left as relative (that is, include \$RULE_PATH/local.rules) because the RULE_PATH variable will be expanded to make it an absolute path.

15. After you are satisfied with your changes, save and close the configuration file.
16. The basic install does not include any rules. Go to www.snort.org and click **RULES** on the left side of the Web page. On the next page, click **DOWNLOAD RULES** on the far-right side of the page. Scroll down to **Sourcefire VRT Certified Rules – The Official Snort Ruleset (unregistered user release)** and click **Download** by the most current ruleset. The ruleset will be a compressed file so you will need a program to uncompress it; IZArc or FileZip are good options. There is also a selection of community-provided rules at the bottom of the page. If you are looking for something unusual, you might find it there without having to create the rule yourself.
17. Extract all files in the archive's signatures folder to **C:\snort\doc\signatures** and extract all files in the archive's rules folder to **C:\snort\rules**. This will take some time because there are currently about 3,700 rules.

You are now ready to start up Snort and see what it looks like in action. Go to a command prompt window and change your working directory to the \snort\bin directory, which is where the snort.exe is located. Type **snort -W** to list the available interfaces. In my case I get the output shown in Figure 4.6.

Figure 4.6 Snort Interface Listing

```
C:\Snort\bin>snort -W

,,_      -*> Snort! <*-
o" )~    Version 2.6.0-ODBC-MySQL-FlexRESP-WIN32 (Build 57)
'''      By Martin Roesch & The Snort Team: http://www.snort.org/team.html
         (C) Copyright 1998-2006 Sourcefire Inc., et al.
```

Interface	Device	Description
1	\Device\NPF_GenericDialupAdapter	(Generic dialup adapter)
2	\Device\NPF_{F95B71A4-C943-40BA-9F65-CD73D4B20769}	(Intel(R) PRO/100B PCI Adapter (TX))
3	\Device\NPF_{A7F703C5-7567-49BC-B6C1-1A1F14614CAF}	(SiS NIC SISNIC)

(Note: The line has been wrapped for Interface 2 to fit this page.)

When we start Snort, we can specify the interface to listen on using the `-i` switch. If you don't specify, it will use the first interface, which in my case won't see anything because it's a dial-up interface that is not in use. Use the `-c` option to tell Snort which configuration file to use. It can be useful to have multiple configuration files configured so that you can quickly switch configurations for special circumstances. You could prepare different configuration files to home in on certain issues, segments, or more in-depth logging. Another important option is `-A`, which tells Snort what type of alerts to generate. The options are fast, full, console, or none.

The following command example would start Snort listening on interface 3, with alerts going to the console only, using the configuration file at `C:\snort\etc\snort.conf`. The `-K` switch tells Snort what types of logs to generate. ASCII logs are easier for a human to read, but they take a little more time to log. If speed isn't a concern, the ASCII logs will probably be the easiest to read and analyze manually.

```
snort -A console -i 3 -c C:\snort\etc\snort.conf -l C:\snort\log -K ascii
```


You should see any triggered rules produce a message on the console. If you add the `-s` switch to the end of the line, it will tell snort to log to the syslog server you have configured in the `snort.conf` file; however, it will not also display on the snort console. If you want to create a rule for testing purposes to see what the results look like, create a test rule file, such as `TESTING.rules`, and place it in the rules folder (`C:\snort\rules\` by default). In this file you could place the following line, which would trigger on any attempts to ping another system.

```
Alert icmp any any -> any any (msg:"TESTING rule"; sid:1000001;)
```

Edit the `snort.conf` to include your new rule by adding the following line: **include \$RULE_PATH/TESTING.rules**. As a last step, edit the `snort\etc\sid-msg.map` file. This file provides a mapping between snort alert messages and alert IDs or numbers. Custom alerts should use an ID number of more than one million. Add the following line at the end of the file:

```
1000001
```

Placing the ID number is the minimum requirement for Snort not to output an error. You can certainly fill in all the other fields, following the existing message maps as a guideline. When this is done, you will need to stop and restart Snort. Here is the console output of a single ping and the reply:




```
08/10-18:22:19.823970  [**] [1:0:0] TESTING rule [**] [Priority: 0] {ICMP}
192.168.1.99 -> 192.168.1.1
08/10-18:22:20.284438  [**] [1:0:0] TESTING rule [**] [Priority: 0] {ICMP}
192.168.1.1 -> 192.168.1.99
```

You can also add your own custom rules to the `local.rules` file. When you open the file, you will find it is essentially empty, existing solely for you to place your custom rules in it. The `local.rules` is “included” in the `snort.conf` by default, so you will not need to add it there. You will, however, still need to edit the `sid-msg.map` file for any rules placed in `local.rules`. The aforementioned command example would display only to the console. For day-to-day operations you would probably want to use fast alerts in your log files, which look like the ones that are sent to the console with the *console* option.

```
snort -A fast -I 3 -c C:\snort\etc\snort.conf -l C:\snort\log -K ascii -s
```

Congratulations! You now have a working IDS. Packets will get logged by default to `C:\snort\log\`. A subdirectory will be created for each source IP that triggers an alert. In this subdirectory will be placed a log file named after the rule that was triggered. Additional instances of the same alert will be appended to the same file. Figure 4.7 shows an example of the log file `C:\snort\log\192.168.1.99\ICMP_ECHO.ids`:

Figure 4.7 ICMP Example Log



```
[**] TESTING rule [**]
08/10-20:25:51.282620 192.168.1.99 -> 192.168.1.107
ICMP TTL:128 TOS:0x0 ID:13266 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:28928 ECHO
```

```

61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+

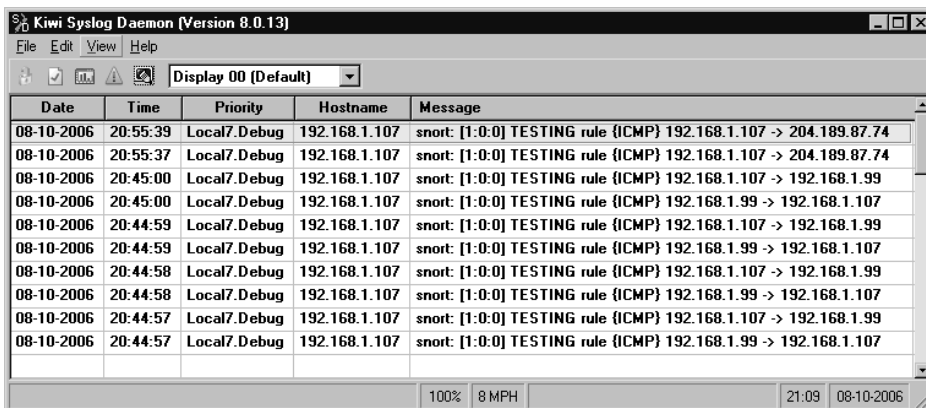
[**] TESTING rule [**]
08/10-20:25:52.282888 192.168.1.99 -> 192.168.1.107
ICMP TTL:128 TOS:0x0 ID:13274 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:29184 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+

```

Note that the output on the console (same as fast) is not the same as those logged in `\log\`. The logged packets also include the data portion of the ICMP ping (a through z repeated). The preceding configuration will log to the syslog server you specified in the `snort.conf`. In my case, the syslog server is Kiwi syslog. The incoming alerts for the ICMP test rule are shown in Figure 4.8.

Figure 4.8 Snort Sending Syslog Alerts to Kiwi Syslog



Using a Snort GUI Front End

Many times the command-line options for programs with lots of functionality can seem cryptic, opaque, or even overwhelming. At these times a GUI front

end can make things a lot easier. Rather than know a certain command-line option and syntax, a check box can often be a lot easier to get right. Even an experienced admin can find these front ends easier to use than the command-line versions. While it's always going to be preferable to know the command-line operation *in addition* to being able to use a GUI, there is no need to memorize a lot of syntax if you don't have to. Although it is capable of “managing” the execution of Snort, IDS Policy Manager (IDSPM) is primarily geared toward managing and customizing the Snort rules.

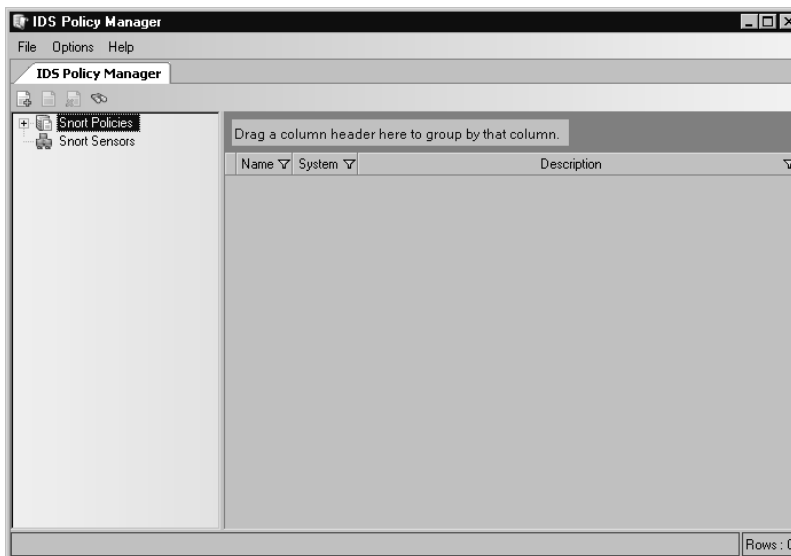
Configuring IDS Policy Manager

IDS Policy Manager is available for download from www.activeworx.org/programs/idspm/index.htm. This program will run on Windows 2000 and Windows XP and provides a graphical interface for Snort rule management and configuring Snort itself via the Snort configuration file. IDSPM does not need to be installed on the sensor itself; in fact, one of the strengths of IDSPM is that it can manage multiple sensors remotely. IDS Policy Manager's primary strength is in its capability to manage the Snort rules, making this a must have for anyone who will be customizing and working with their rules extensively. IDSPM also supports the automated download of the newest Snort rules, using Oinkmaster. Setting up IDSPM can be accomplished by following these steps.

1. Download and run the installation program.
2. If you do not currently have the Microsoft .NET 2.0 framework installed you will be asked if you want to install it. The window that prompts you will refer to it as an optional component. In my case the product would not install until I had installed .NET V2, so I'm not sure how optional it really is. This shouldn't pose any issues unless you are running some other software that relies on older .NET features and is incompatible with the newer version.
3. Follow the installation prompts, accepting the license agreement and choosing the installation directory.
4. When you first run the software, you will see a pop-up window alerting you that your oinkcode is not set up; click **OK** to get past this message.

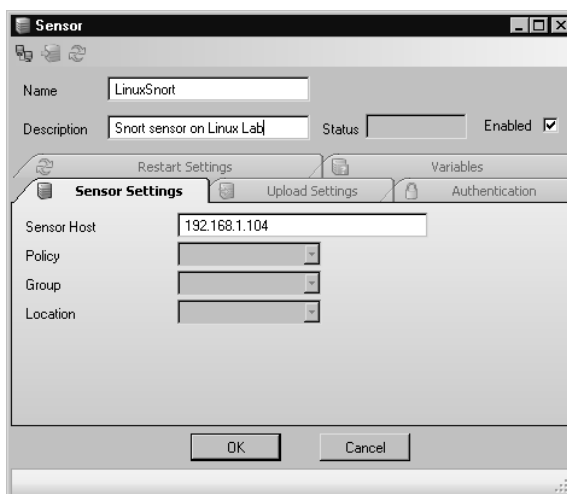
- Next open the IDS Policy Manager shortcut. The opening screen is shown in Figure 4.9.

Figure 4.9 IDS Policy Manager



One of the first steps is to configure adding a sensor and then configure Oinkmaster. Add a sensor by right-clicking **Snort Sensors** and selecting **Add Sensor**. There are several tabs of information to fill out on the Sensor properties window shown in Figure 4.10.

Figure 4.10 IDSPM Add Sensor



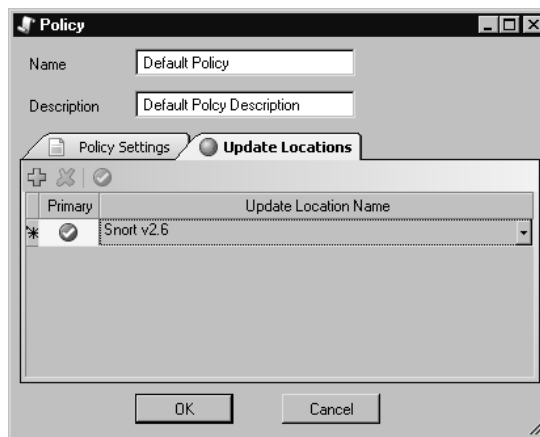
6. At a minimum, fill out the **Name** and a **Description** for the sensor.
7. Also enter the **IP address** or host name on the **Sensor Settings** tab.
8. On the **Authentication** tab, enter the **username** and **password** to use to connect to the sensor (IDSPM will use SSH to communicate with the sensor). You can also use PKI for authentication. If you select PKI in the **Authentication Mode** drop-down box, the password fields will then change to fields to indicate the location of your public and private key files.
9. On the **Upload Settings** tab, ensure that the Upload Directory is configured; by default it's **/etc/snort/rules**.
10. When you are finished filling out the information, click the small monitors in the upper-left corner of the window. This will test the SSH connection to the server. The first time it connects, you will get the standard choice of accepting the RSA key or not. Choose **Yes**. Afterwards a brief Test connectivity Log will be displayed. All these should have a result of OK. Click **OK** to continue.
11. Click **OK** to close the Sensor properties window.

To configure the Oinkmaster portion of IDSPM, you will need to go to www.snort.org and register so that you can download the rules file. After registering, log onto the Snort Web site and click the link that says **User Preferences**. At the bottom of the page is a section titled Oink Code; click the **Get Code** button. Copy this code for use in the Oinkmaster configuration file.

12. Navigate to **Options | Settings**.
13. In the **Settings** pane on the left, select **Miscellaneous**.
14. You will need to paste the Oink Code you generated previously, so that Oinkmaster can download the latest Snort rules.
15. Use the drop-down boxes to select how often you wish to check for updates and how often to back up the rules database. After you are finished, click **OK**.

16. The next step is to create a policy. In this context, a policy is a definition of which rules to apply to a given sensor. Right-click **Snort Policies** and select **Add Policy**.
17. Provide a **name** and **description** for the policy. Use the drop-down box to select the Snort version. The **Initialize policy** check box should be checked, so that it will apply the new settings immediately.
18. Select the **Update Locations** tab shown in Figure 4.11. Click the “plus” to add a location.
19. Click the cell under **Update Location Name** and select the appropriate location. You can define alternate locations at **Options | Setting** under **Update Locations**. After selecting the update location, click **OK**.

Figure 4.11 IDSPM New Policy Locations



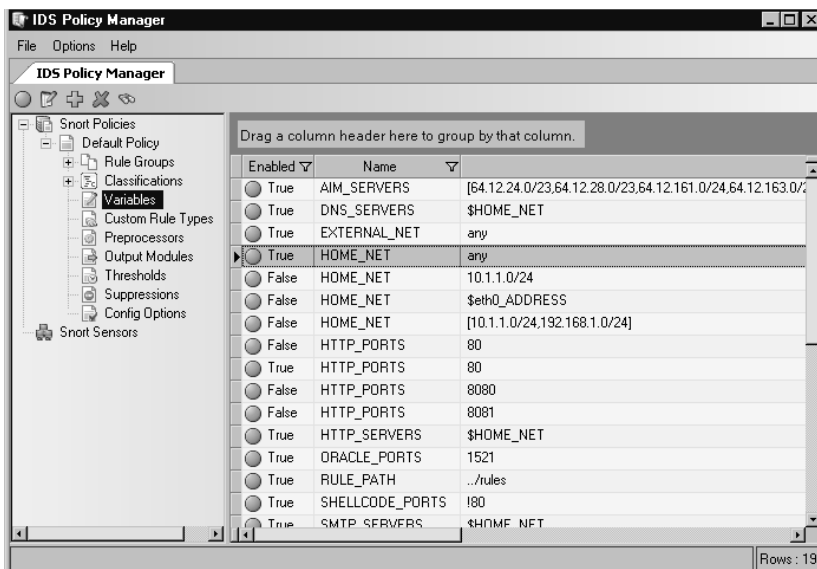
20. The **Initialize Policy** window will come up. This window enables you to pull your rules from a pre-defined location (in this case, the one called “Snort 2.6, which is a Web URL), a local file, or another HTTP address that has not been pre-defined. Select the proper location (or just leave the default) and click **Start**.
21. The next step is to edit the policies’ various properties to match your environment.

NOTE

There is no mechanism to import your current Snort configuration into IDSPM. This means that if you have a working Snort configuration already, you will need to redefine it within IDSPM. After you start using IDSPM to manage your Snort sensors, you shouldn't ever need to edit the sensors' configuration directly and, in fact, doing so would cause your changes to be overwritten the next time you applied the configuration from IDSPM.

When you click the plus next to Snort Policies, it should expand and show the newly created policy. After you expand the newly created policy, a number of property groups come into view, as shown in Figure 4.12. The primary one to configure is the **Variables** group. This is where you set the various variables in the configuration file so Snort knows what alerts to look out for.

Figure 4.12 IDSPM Variables



In the example you will see that there are multiples of many variables defined. This is done as a convenience to enable you to easily switch between them by right-clicking and selecting **Disable Item** or **Enable Item**. If, for

example, you don't want HOME_NET to be any (the default), you will right-click the highlighted variable and select **Disable Item**. You could then double-click (or right-click and select **Edit Item**) the HOME_NET that is defined as 10.1.1.0/24 and edit it. After changing the value to 192.168.1.0/24, click **Save**. Lastly, right-click the newly defined HOME_NET and select **Enable Item**. Refer to the beginning of this chapter for a more-detailed description of the various variables you need to set.

If you need to edit the output modules, such as if you wanted Snort to log to MySQL, you would select the **Output Modules** section. If you do want to use Snort to log to a MySQL database, select either of the output modules with a name of "database" and with "mysql" in the value column. There should be two available, and each is the same except one specifies the localhost for the DB user. After editing the value to match your user name, database name, and the MySQL password, click **Save**. After the rule value has been saved, right-click and select **Enable Item**.

To select which rule groups to apply, select **Rule Groups** in the left pane. Each category can be enabled or disabled. These settings correspond to commenting out the include statements in the snort.conf file. For example, to enable all the backdoor checks (over 500), right-click the row with "backdoor" in the name column and select **Enable Item**. By drilling down in the left column and selecting backdoor there, you can choose between individual rules to enable or disable in the right column. A very handy feature of IDSPM is the Find Rule function. With Rule Groups selected in the left pane, a small pair of binoculars will appear in the upper-left of the window; click this to open the **Find Rule** dialog. You can enter a **Rule ID** or **Rule Name** and then click **Search**. You don't have to know the entire name; you can enter a partial name and it will pull up a list of rules.

Perhaps the most compelling feature of IDSPM is the GUI interface for creating your own custom rules. Follow these steps to create your own custom rule.

1. Drill down into **Rule Groups** until you get to individual rules in the right pane (it doesn't matter which group you are in).
2. Click anywhere in the right pane and select **Add Item**. The Rule window is shown in Figure 4.13.

Figure 4.13 IDSPM Rule Editor

The screenshot shows the 'Rule - TestRule' window with the following configuration:

- Name: TestRule
- Group: experimental
- Enabled:
- Signature ID: 1000001
- Revision: 1
- Settings tab is active.
- Action: log
- Protocol: icmp
- Classification: icmp-event
- Priority: 3
- Source IP/Mask: any
- Source Port: any
- Direction: ->
- Destination IP/Mask: 192.168.1.104/32
- Destination Port: any
- Rule Options: (empty)
- References table:

Type	Value
- Add Reference button is present.
- OK and Cancel buttons are at the bottom.

3. Start by entering a **Name** for the rule.
4. Select a **Group** for the rule to go into. This drop-down selection is why it doesn't matter which group you are in when you click Add Item. The **Local** group has been created specifically for the placement of custom rules.
5. The Settings tab is where you specify what triggers the rule. For example, if we wanted to create a rule that would trigger any time ICMP was sent to the Snort sensor (192.168.1.104), we could easily do so. For **Action**, use the drop-down box to select the desired action. Log will log the packet, while Alert will show an alert on the Snort console. We will select **Alert** for this exercise.
6. For **Protocol**, use the drop-down list to select **icmp**.
7. For **Classification**, use the drop-down list to select **icmp-event**.
8. In the **Destination IP/Mask** field, you can type 192.168.1.104/32.
9. Enter a unique **Signature ID** number. Any custom rules should have ID numbers over 1,000,000 (the first one million IDs are reserved).
10. Take note of the **Rule Options** field, but for now leave it blank.
11. Place a **Check** in the **Enabled** box at the top and click **OK**.

The **Rule Options** field deserves a closer look. This is where you specify the bulk of the Snort rule logic. This is where the really interesting information is placed. There are currently four types of rule options: meta-data, payload, non-payload, and post-detection. Odds are good that the majority of what you might want to search for would be done using the payload option, which enables you to trigger based on defined strings being present (or absent) from the packet. While the rule options are behind the true power of Snort's custom rules, don't forget that there is a repository of user community rules available (from www.snort.org). Unless you are trying to match a rule based on very unusual characteristics, odds are good that the rule is already out there.

12. After you have finished all your customization, it's time to assign the new policy to your sensor and apply the policy. Select **Snort Sensors** in the left pane and then right-click and select **Edit item**, or double-click the sensor row in the right pane.
13. In the **Policy** drop-down box, select your new policy and click **OK**.
14. Now right-click the sensor and select **Upload policies to Sensors**.
15. The next window enables you to place a check next to each sensor you want to update. The status column will tell you if any rules applied to the selected sensor have been changed. If so, the status will read "Sensor needs to be updated." When satisfied with the selection, click **Start**.
16. After it is finished, click **Close**.

You will find that the `/etc/snort/rules/` directory contains a file called **local.rules**. The `snort.conf` file has an **include \$RULE_PATH/local.rules** entry to enable the rules in this file. If you open this file, you can see our custom rule is there:

```
alert icmp any any -> 192.168.1.104/32 any (msg:"TestRule"; classtype:icmp-  
event; sod:1000001; rev:1)
```

The resultant alert on the Snort console is also shown here.

```
12/01-12:16:41.236240 [**] [1:1000001:1] TestRule [**] [Classification:  
Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.99 -> 192.168.1.104
```

Configuring Snort on a Linux System

The process of installing Snort on a Linux system is very close to the process on a Windows system. The primary difference is that the default (relative) paths in the `snort.conf` file are much more likely to work without modification on the Linux system. You will need to download the latest version of Snort that is appropriate for your system. If you are using Fedora Core 5, this is as simple as typing `yum install snort`, or you could download and install the .rpm from snort.org.

Configuring Snort Options

The next step is to configure the various options that determine how Snort will behave using the Snort configuration file. The configuration file is excellently documented and very easy to use. To get Snort working the way you want it to, follow these simple steps.

1. Start by opening the main Snort configuration file. By default it will be located at `/etc/snort/snort.conf`.
2. Configure the `HOME_NET` variable, if desired, by removing the `#` from the line you need. `#` is a comment indicator in the Snort configuration file. The `HOME_NET` variable defines which networks are the “trusted” internal networks. This is used with the signatures to determine when the internal network is being attacked. By default, `HOME_NET` is set to *any* network with the `var HOME_NET any` line in the `snort.conf`. Setting this to accurately reflect your internal address space will reduce the number of false positive alerts you receive. A common example would be `var HOME_NET 192.168.1.0/24` or perhaps `var HOME_NET [192.168.1.0/24,192.168.2.0/24]`.
3. Configure the `EXTERNAL_NET` variable if desired. This is the network you expect attacks to come from. The recommendation is to set this to everything *except* your `HOME_NET` using the following: **`var EXTERNAL_NET !$HOME_NET`**. (Default: `var EXTERNAL_NET any`.)
4. Next, define what servers are running specific services. For example, by setting `HTTP_SERVERS` to only specific servers, Snort will only

watch for HTTP attacks targeted at those servers. If you wish to see attacks targeting servers that are not running the affected services, leave the defaults, which are to watch for attacks directed towards *any* internal servers. (Default: *var DNS_SERVERS \$HOME_NET*) If you had a Web server running on 192.168.1.11 and 192.168.1.12, you could tell Snort to only look for HTTP attacks targeting that server by setting the following variable: *var HTTP_SERVERS [192.168.1.11/32,192.168.1.12/32]*.

5. If desired, configure the specific ports that services are available on. For example, the default for HTTP is defined on the following line: *var HTTP_PORTS 80*. Similar to defining the servers in the preceding section, this will tell Snort to only look for attacks targeting specific ports. With the default configuration, Snort would *ignore* an HTTP attack to port 8080. Again, this setting will help focus where Snort looks for different types of attacks to occur.
6. If you are interested in detecting the usage of AOL Instant Messenger (AIM), the various IP addresses of the AIM servers are defined in the `snort.conf` file. This is done because the IP addresses change frequently, and by using a variable, the rules don't have to be updated each time the IP address changes. If you don't wish to trigger based off AIM usage, don't worry about changing these IP addresses.
7. Download the Snort rules from <http://snort.org/rules>. Click **Download Rules** on the right-hand side of the page. On the **Download Rules** page, scroll down to the section labeled **Sourcefire VRT Certified Rules (unregistered user release)**. Download the latest ruleset.
8. Extract the rules (and /docs) to the location of your choice, typically `/etc/snort/rules` and `/etc/snort/docs`.
9. Configure the `RULE_PATH` variable, which tells Snort where to find the rules used for triggering events. You can use a relative path such as *var RULE_PATH ../rules* or an absolute path such as *var RULE_PATH /etc/snort/rules*.

10. The next section has some commented out lines to disable certain detections of some infrequently seen types of traffic. Unless you are having some issues with those alerts or your IDS is very low on resources, it's probably fine to just leave those at the default (enabled) configuration.
11. The last few lines of the “step 1” section enable you to configure the detection engine for systems with limited resources. Unless you are having issues, you can leave this option alone.
12. After that the “step 2” and “step 3” sections of the configuration file allow you to enable or disable specific functionality and detect particular types of attack, such as fragmentation attacks, stateful inspection, and stream reassembly options. Delving into the specifics of each of those options is beyond the scope of this chapter and for many, the default setting will serve them well.
13. The section labeled Step #4 contains output options for Snort. Uncomment **output alert_syslog: LOG_AUTH LOG_ALERT** (the default). Despite what facility and severity you configure here, the snort alerts will be generated as auth.info. You also need to include the `-s` switch on the command line to enable syslog logging. We will discuss syslog in more detail in the next chapter. If you don't have a syslog server to log to yet, just make note of the setting and come back to it when your syslog server is set up.

Using the preceding example of LOG_AUTH and LOG_ALERT, you would need the following in your syslog.conf file to log to a syslog server at 192.168.1.99:

```
auth.info          @managmentserverIP
```

If you are using syslog-ng, you would need a logging destination defined, a filter that specifies what events to capture, and a log statement in the syslog-ng.conf file. An example of this configuration would be the following:

```
destination d_lab { udp ("192.168.1.99" port(514)); };
filter f_most { level(info..emerg); };
log { source(s_sys); filter(f_most); destination(d_lab); };
```

14. Edit the paths for the dynamically loaded libraries in section #2 to point to the proper path. Depending on your Linux distribution and installation method, these paths may not be the default. For example, on Fedora Core 5, using yum to install Snort, the settings would use the following paths: *dynamicpreprocessor directory /usr/lib/snort/dynamicpreprocessor* and *dynamicengine /usr/lib/snort/libsf_engine.so*. If you receive an error when you try to run Snort, along the lines of *Unknown rule type: dynamicpreprocessor directory* or *Unknown rule type: dynamicengine*, then your installation of Snort is not configured to use dynamically loaded processors. In this case, simply place a # in front of both of those lines to comment them out.
15. The last section (Step #6), contains various include statements that specify the rulesets to be checked. Some rules are disabled by default, such as chat.rules, which is triggered by the use of various instant messaging clients. To enable or disable a given ruleset, simply add or remove a # at the beginning of the include line. This entry can be left as a relative path (for example, include \$RULE_PATH/local.rules) because the RULE_PATH variable will be expanded to make it an absolute path.
16. If you need any custom rules that are not included with the standard Snort release, you can download rules provided by the Snort community from the Rules page on the Snort Web site. If you are looking for something unusual, you might find it there without having to create the rule yourself.

You are now ready to start up Snort and see what it looks like in action. When you start Snort you can specify the interface to listen on using the *-i* switch such as *-i eth0*. If you don't specify, it will use the first interface. Use the *-c* option to tell Snort which configuration file to use. It can be useful to have multiple configuration files configured so you can quickly switch configurations for special circumstances. You could prepare different configuration files to home in on certain issues, segments, or more in-depth logging. Another important option is *-A*, which tells Snort what type of alerts to generate. The options are fast, full, console, or none.

The following command example would start Snort listening on the first interface (no `-i` used), with alerts going to the console only, using the configuration file at `/etc/snort/snort.conf`. The `-l` switch tells Snort where the logging directory is located. The `-K` switch tells Snort what types of logs to generate. ASCII logs are easier for a human to read, but they take a little more time to log. If speed isn't a concern, the ASCII logs will probably be the easiest to read and analyze.

```
snort -A console -c /etc/snort/snort.conf -l /etc/snort/log -K ascii
```

You should see any triggered rules produce a message on the console and logged to your syslog server. If you add the `-s` switch to the end of the line, it will tell snort to log to the syslog server you have configured in the `snort.conf` file; however, it will not also display on the snort console. If you want to create a rule for testing purposes to see what the results look like, create a test rule file, such as `TESTING.rules`, and place it in the rules folder (`/etc/snort/rules`, in this example). In this file you could place the following line, which would trigger on any attempts to ping another system.

```
Alert icmp any any -> any any (msg:"TEST rule";sid: 1000001;)
```

Edit the `snort.conf` to read your new rule by inserting the following statement towards the end of the file: **include**

\$RULE_PATH/TESTING.rules. As a last step, edit the `snort\etc\sid-msg.map` file. This file provides a mapping between snort alert messages and alert IDs or numbers. Custom alerts should use an ID number of more than one million. Add the following line at the end of the file:

```
1000001
```


Placing the ID number is the minimum requirement for Snort not to output an error. You can certainly fill in all the other fields, following the existing message maps as a guideline. When this is done, you will need to stop and restart Snort. Here is a partial display of the console output of a single ping and the reply.

```
10/12-21:29:35.911089  [**] [1:0:0] TEST rule [**] [Priority: 0] {ICMP}
192.168.1.99 -> 192.168.1.103
08/10-18:22:20.284438  [**] [1:0:0] TEST rule [**] [Priority: 0] {ICMP}
192.168.1.103 -> 192.168.1.99
```

You can also add your own custom rules to the `local.rules` file. When you open the file, you will find it is essentially empty, existing solely for you to place your custom rules in it. The `local.rules` is “included” in the `snort.conf` by default, so you will not need to add it there. You will, however, still need to edit the `sid-msg.map` file for any rules placed in `local.rules`.


The `-A` option will alter the display of the alerts on the console, while the `-K` option controls how the alerts are logged to the log directory. You should experiment with the different display formats to find the one that provides adequate information with the minimal strain on the Snort host. For day-to-day operations you would probably want to use fast alerts in your log files, which look like the ones that are sent to the console with the `console` option. Available alert modes and logging formats are outlined here for handy reference.

- **-A console** Logs to the console in the following format:



```
10/12-21:29:35.911089  [**] [1:0:0] TEST rule [**] [Priority: 0] {ICMP}
192.168.1.99 -> 192.168.1.103
```

- **-A fast** Logs in the same *format* as console, but writes the alerts to a `/snort/alert` file with no output to the console.
- **-A full** Logs to the `/snort/alert` file in the following format:



```
[**] [1:0:0] TEST rule [**]
[Priority: 0]
10/12-21:38:53.741606 192.168.1.103 -> 192.168.1.99
ICMP TTL:64 TOS:0x0 ID:6350 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:7936 ECHO REPLY
```

- **-K pcap** This is the default mode if you don’t specify an alternate format on the command line. This file will contain the alert packets in their entirety. You can open this file using a network sniffer such as Wireshark.
- **-K ascii** Will create a folder under `/log` for each IP address. Within that folder each rule will create a log file. The log entries will be the same format as the “full” alert format.
- **-K none** No log file will be created.

Congratulations! You now have a working IDS. Figure 4.14 shows the syslog alerts from the TESTING.rule in the Kiwi Syslog Daemon console.

Figure 4.14 Snort Alerts in Kiwi Syslog Daemon Console

Date	Time	Priority	Hostname	Message
10-12-2006	21:58:32	Auth.Info	192.168.1.103	Oct 12 21:57:25 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.103 -> 192.168.1.99
10-12-2006	21:58:32	Auth.Info	192.168.1.103	Oct 12 21:57:25 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.99 -> 192.168.1.103
10-12-2006	21:58:31	Auth.Info	192.168.1.103	Oct 12 21:57:24 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.103 -> 192.168.1.99
10-12-2006	21:58:31	Auth.Info	192.168.1.103	Oct 12 21:57:24 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.99 -> 192.168.1.103
10-12-2006	21:58:30	Auth.Info	192.168.1.103	Oct 12 21:57:23 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.103 -> 192.168.1.99
10-12-2006	21:58:30	Auth.Info	192.168.1.103	Oct 12 21:57:23 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.99 -> 192.168.1.103
10-12-2006	21:58:29	Auth.Info	192.168.1.103	Oct 12 21:57:22 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.103 -> 192.168.1.99
10-12-2006	21:58:29	Auth.Info	192.168.1.103	Oct 12 21:57:22 localhost snort: [1:0:0] TEST rule {ICMP} 192.168.1.99 -> 192.168.1.103

Using a GUI Front End for Snort

Like the Windows version of Snort, some have felt the administration of Snort could be improved upon by implementing a more robust GUI interface. There are several Snort GUIs to choose from aimed at both the configuration of Snort, as well as the interpretation of the Snort alerts. Some really only offer buttons to configure options on the Snort command line, and offer very little additional functionality, while others bring some very powerful additional features to the table. We will discuss the operation of some of the better offerings in the next section.

Basic Analysis and Security Engine

Basic Analysis and Security Engine (BASE) is available for download from <http://base.secureideas.net/about.php>. The purpose of BASE is to provide a Web-based front end for analyzing the alerts generated by Snort. Base was derived from the ACID project (Analysis Console for Intrusion Databases). Whereas ACID is more of a general-purpose front end for viewing and searching for events, BASE is a Snort-specific utility. The instructions to configure BASE assume you have already installed and configured Snort. Snort must be installed with the *—with-mysql* switch because Snort does not support MySQL output by default. The Snort Web site has RPM packages with MySQL support already included for some operating systems. This is the list

of dependencies for running BASE: httpd, Snort (with MySQL support), MySQL, php-gd, pcre, php-mysql, php-pdo, php-pear-Image-GraphViz, graphviz, and php-adodb. Follow these steps to get BASE up and running.

1. Download and install MySQL and BASE
2. Edit the `/snort/snort.conf` file. Uncomment and edit the following line:

```
output database: log, mysql, user=snort password=snortpass dbname=snort
host=localhost
```

3. The next few steps are related to setting up the MySQL database and settings. After you install MySQL, enter the MySQL commands by typing **mysql** on the command line. This will place you in an interactive command mode. All commands must have a semicolon at the end of the line. By default, the MySQL installation will not have a password set at all. You should add a default password with the following commands.

```
mysql
mysql> SET PASSWORD FOR root@localhost=PASSWORD('somepassword');
```

After you have assigned a password to the root account, simply entering `mysql` will not enable you to access the interactive command mode. After a password has been assigned, use **mysql -u <username> -p**. You will then be prompted to enter the password for the user you specified (typically root).

4. The next step is to create the Snort database.

```
mysql> create database snort;
```

5. You now need to give the Snort user permissions to add the needed tables to the Snort database. Use these commands:

```
mysql> grant INSERT,SELECT on root.* to snort@localhost;
```

6. You should now set the password for the Snort user to the same password you used in the Snort configuration file.

```
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('snortpass');
```

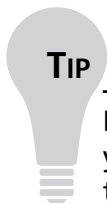
7. The next step is to add some additional permissions for the Snort database using the following commands:

```
mysql> grant ALL on snort.* to snort@localhost;
mysql> grant ALL to snort;
mysql> exit
```

- Now that the database has been created, you need to populate it with the tables Snort uses. Use the following command to create the tables:

```
mysql -u root -p < /etc/snort/schemas/create_mysql snort
```

When the command completes, it will not give any indication of its success; therefore, it will be necessary to manually verify that the tables were created.



If the package you installed did not include the `/snort/schemas/` directory, you can download the source package and extract the directory from there. With Fedora Core 5, for some reason installing the Snort with MySQL support did *not* include the schemas directory.

- Verify the MySQL tables were created in the Snort database by entering the following commands. You should see output similar to that shown in the following example:

```
SYNGRESS
syngress.com
mysql -u root -p
show databases;
+-----+
| Database |
+-----+
| mysql   |
| snort   |
| test    |
+-----+
use snort;
show tables;
```



```

+-----+
| Tables_in_snort |
+-----+
| data            |
| detail          |
| encoding        |
| event           |
| icmp_hdr        |
| ip_hdr          |
| opt             |
| reference       |
| reference_system |
| schema          |
| sensor          |
| sig_class       |
| sig_reference   |
| signature       |
| tcp_hdr         |
| udp_hdr         |
+-----+
exit

```

The list of databases is not significant, as long as the Snort database exists, of course. The table listing must be accurate. If any are missing, Snort will generate an error when you run it.

10. Install **php-gd**, which is used to generate the graphs in BASE. On Fedora Core 5 you can just type **yum install php-gd**.
11. Install ADODB, which is a database abstraction library for PHP. On Fedora you can simply enter **yum install php-adodb**.
12. It's now time to configure BASE itself. Edit the **/usr/share/base-php4/base_conf.php** file to ensure that the following lines are configured with paths and settings appropriate for your configuration.

```

SYNGRESS $BASE_urlpath = '/base';
syngress.com $DBlib_path = '/usr/share/ododb';
$DBtype      = 'mysql';
$alert_dbname = 'snort';
$alert_host   = 'localhost';

```

```
$alert_port      = '';
$alert_user      = 'snort';
$alert_password = 'snortpass';
```

You should not be able to access the BASE Web page at the following URL: `http://localhost/base/`.

Tools & Traps...

Troubleshooting Tips

- You can enable debugging in BASE by editing the `/usr/share/base-php4/base-php4.conf` file.

```
$debug_mode = 2;
```

- Use `chkconfig` to make sure that MySQL, Snort, and `httpd` are running.

```
Chkconfig --list | grep snort
```

```
Snortd      0:off  1:off  2:on   3:on   4:on   5:on   6:on
```

If all entries say “off,” then that service is configured not to start. Try **service snortd start**.

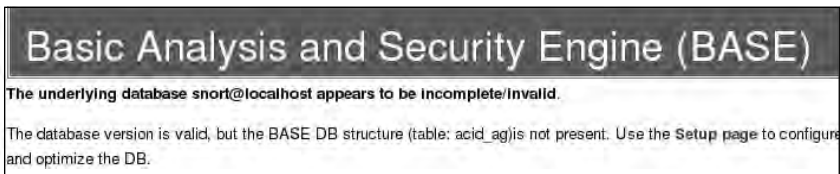
- `Httpd` may need to be restarted for some configuration changes to take effect; when in doubt, restart it just to be safe: **service httpd restart**.
- The `httpd` access log and error log can be found at `/etc/httpd/logs`.
- You can control the logging level of the `httpd` by editing `/etc/httpd/conf/httpd.conf`.

```
LogLevel debug
```

- If you are having issues with the URLs not being found, the `/etc/httpd/conf.d/base-php4.conf` file tells the Web server to alias `/base/` with the directory `/usr/share/base-php4/`.

The very first time you start up BASE, none of the database tables have been created. You will see something like the page shown in Figure 4.15.

Figure 4.15 BASE Setup



13. Click on the **Setup page** link.
14. Click the **Create BASE AG** button on the right-hand side. You see several success messages as shown in Figure 4.16.

Figure 4.16 BASE Success



15. Click the **Main Page** link. This should take you to the primary BASE interface as shown in Figure 4.17.

Although this window may not be too flashy, there is a wealth of information you can discover. Most of the fields are actually links. By clicking to the right of **Today’s alerts**, for example, you can get a sorted list of unique alerts, a listing of all alerts, or a list sorted by source IP address or destination

IP address. The other headings along the left side offer similar functionality. Of particular note are the links for the **Most Frequent 15 addresses** by source address. This would enable you to quickly see which systems are *generating* the majority of your alerts. If you open that window (shown in Figure 4.18) there are several additional fields that are also hyperlinked.

Figure 4.17 BASE Main Page

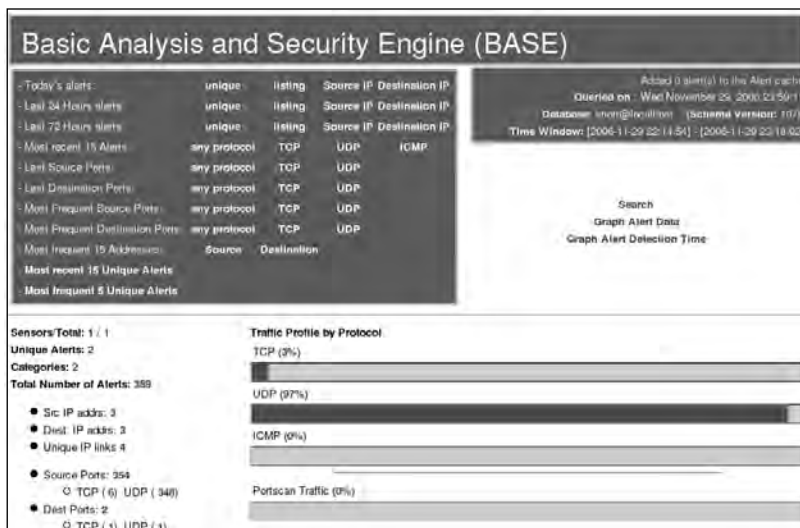
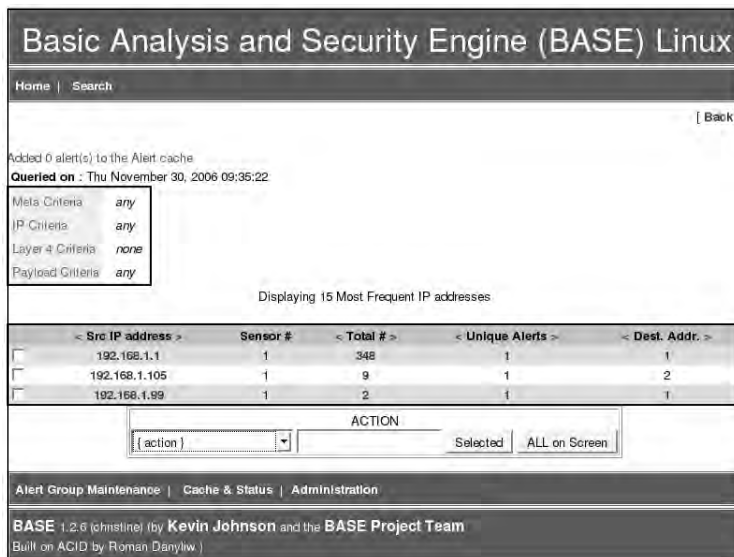


Figure 4.18 BASE Most Frequent by Source IP



Note the field at the bottom labeled **ACTION**. This enables you to configure the *alert groups*. Alert groups are basically shortcuts to enable you to view a subset of alerts quickly, without having to navigate through the various menus to get there. For example, suppose you want to know anytime that 192.168.1.1 generates an alert. You can check the check box to the left of 192.168.1.1, and then use the {action} drop-down box to select Create AG (by Name). In the action column, enter .1_ALERTS to use as the alert group name. Finally, click **Selected**.

The next screen enables you to enter a description for the newly created alert group. Enter a meaningful text description for the group and click Save Changes. The next screen will be a listing of all alerts from 192.168.1.1. This screen *is* the alert group. In the future, if you want to quickly see this group of alerts, you can click Alert Group Maintenance at the bottom of each page, and then click the alert group you want to view. In this way, any subset of alerts is only two clicks away, sort of like a shortcut straight to a particular set of filtering criteria.

Another feature of note is the Administration link at the bottom of each page. This will take you to a screen where you can configure users for BASE. There are four options on the administration screen: list users, create a user, list roles, and create a role. These screens enable you to create users and assign them to various roles. If you click List Roles, you can see the four predefined roles. If you want to assign a user in the administrator role, simply click Create a user. Enter the login name, a full name or description, and a password. Use the drop-down box to select a role and then click Submit Query. None of the settings here will take effect until you edit the `base_conf.php` file and change the value of `$Use_Auth_System = 1;`. A value of 0 (the default) means the authentication is disabled and everyone has full access to BASE. Only the admin role has access to the administration screen.

**TIP**

Remember the different logging options for Snort on the command line. Previously, we used `-A console`, which would log Snort events to the Snort terminal. If you are going to be using a different front end for viewing Snort alerts, there isn't much value in also logging to the console. You can

use *-A none* when starting Snort, which will cause Snort not to log anything to the Snort terminal, resulting in improved performance.

Other Snort Add-Ons

The number of Snort utilities and add-ons is impressive. Some of these address such key issues as keeping your Snort rulebase up-to-date, while others provide additional performance improvements such as faster logging. If you are looking for a particular feature or option, you should do some searching on the Internet, and you might find that the functionality you are looking for already exists. If you do find an add-in you are interested in using, remember to properly test it before deploying it in a production environment.

Using Oinkmaster

You may get tired of constantly having to update the Snort signature files. Because Snort is a signature-based IDS, having current signatures is vital. Without current signature files you could be unaware of intrusion attempts happening right in front of you. Although Snort itself does not include any means to automatically update the signature file, there is another utility that can help called Oinkmaster (<http://oinkmaster.sourceforge.net/features.shtml>). Oinkmaster is a Perl script that will update your Snort rules from the Snort Web site automatically. Because it uses Perl, Oinkmaster will run on a Linux or a Windows Snort host. The Oinkmaster Perl script can be scheduled to run and check for updates as often as you like. To download Snort rules without having to wait until the next release of Snort, you have to register on the Snort Web site. You can register for free at <https://snort.org/pub-bin/register.cgi>. A password will be sent to the e-mail address you provide during registration. The configuration of Oinkmaster is outlined here.

1. After logging on to the Snort Web site, click the link that says **User Preferences**.
2. At the bottom of the page is a section titled Oink Code; click the **Get Code** button.

3. **Copy** this code for use in the Oinkmaster configuration file.
4. Download the latest tar.gz from the Oinkmaster Web site.
5. Extract the folder in the archive to **/etc/oinkmaster**.
6. Edit the **oinkmaster.conf** file. Find the line that specified the URL for the current ruleset. (You can search for CURRENT.) Uncomment the line by removing the #, and then paste your oink code into the line in place of **<oinkcode>**.

```
url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode>/snortrules-
snapshot-CURRENT.tar.gz
```

7. Start Oinkmaster with the following command:

```
oinkmaster.pl -C /etc/oinkmaster/oinkmaster.conf -o /etc/snort/rules
```

When it completes, Oinkmaster will tell you what rules were changed/updated. You can also specify the URL to retrieve the rules from the command line using the `-u <URL>` option. To configure the Oinkmaster script to run daily, use *crontab* with the following command:

```
crontab -u <user> -e
```

Enter the username you are running Oinkmaster as in place of `<user>`. This will open the crontab for that user. Adding the following line to the crontab will cause Oinkmaster to run each night at 2:00 A.M. If you prefer, there are also several GUI's available for configuring the cron daemon, such as *gnome-schedule*.

```
0 2 * * * oinkmaster.pl -C /etc/oinkmaster/oinkmaster.conf -o
/etc/snort/rules
```

Now your Snort rules should stay up to date. Remember, if you change Snort versions, the URL to the appropriate rules may change, in which case you will need to update your `oinkmaster.conf` accordingly.

WARNING

Because the `oinkmaster.conf` file contains the path to update your Snort rules, if this file does not have secure access permissions on it, someone who could edit the file could render your IDS useless. With the ability to

edit the configuration file, a malicious user could point the URL to one of his choosing, with empty rule sets that will not trigger on anything, or even worse, rules that work perfectly except ignore the attacker's IP address. Make sure the `oinkmaster.conf` file is secured and only the account you are running Oinkmaster under has access to the file.

Additional Research

If the Snort utilities we have covered don't do everything you want them to do, there are other alternatives. Some of the utilities that are out there are more user friendly than others. Here are a few additional tools that are highly regarded and which may be helpful when running your Snort IDS. These include both Windows- and Linux-based tools.

- **ACID** ACID stands for Analysis Console for Intrusion Databases. You can download ACID from <http://acidlab.sourceforge.net/>. BASE was based off code from ACID, so the interfaces are strikingly similar. If you are only looking to use the Web front end for Snort logs, ACID probably doesn't buy you anything over BASE. If you plan to import data from additional non-Snort sources, however, ACID has the flexibility to do that.
- **Barnyard** Is available from http://sourceforge.net/project/show-files.php?group_id=34732. It is basically a utility to offload the logging overhead from Snort. Using Barnyard, you configure Snort to log binary data (which is the fastest way for Snort to log, but not very human-readable) and Barnyard will then take the binary logs and convert them to human-friendly ASCII or import them into a database. For a small environment with low-alert volumes on the IDS, Barnyard is probably not needed. Snort will support logging to a MySQL database natively without using Barnyard.
- **Sguil** Sguil (<http://sguil.sourceforge.net/>) is pronounced "sgweel" and stands for Snort GUI for Lamerz. It is also referred to as the Analysis Console for Network Security Monitoring. The objective of sguil is to provide more than just a console to view Snort alerts. Sguil

also gives the analyst the capability to delve deeper into an alert, all the way to the captured packet, to facilitate investigation. Basically, Sguil integrates multiple security tools into one interface for easy access. The Sguil developers provide a demo sensor that you can connect to from the Web to see Sguil in action. To use it, simply download and install the Sguil client, and then connect to the sensor demo.sguil.net on port 7734. When prompted, you can enter any user name and password, and then select the sensor named “reset” in the console. Sguil is a powerful tool for investigating Snort alerts, but the configuration and setup is not for the faint of heart.

- **Snortsnarf** This is a log analyzer targeted specifically at analyzing Snort logs. You can download it from www.snort.org/dl/contrib/data_analysis/snortsnarf/.

Demonstrating Effectiveness

One of the age-old debates when it comes to network data collection is placement of the sensors. This applies to both IDS sensors and reporting sensors such as PRTG Traffic Grapher. The most common difference of opinion is whether you should place the sensor outside your external firewall or inside it. This is relevant because the data you see will be drastically different between the two. With the sensor placed outside your perimeter firewall, you will see all traffic directed at you from the Internet, including all the traffic your firewall is blocking. If the sensor is placed inside the perimeter firewall, you will only see the traffic that has managed to pass through your firewall.

Undeniably, the traffic of the most security relevance is the traffic that has managed to traverse your firewall and get into your internal network. These are the potential attacks, probes, and whatnot that need to be inspected and monitored closely to make sure the network is not compromised. If everything is configured properly, an IDS inside the perimeter should really see very little traffic, except perhaps triggers related to IT policy, such as file sharing or instant messaging protocols. So if all the data a security officer would find “interesting” is on the inside, you might wonder what value a sensor on the outside would bring.

The best value for placing a sensor outside is really one of public relations. The unfortunate fact is that when it comes to network security, if everything is done properly, no one ever sees much of anything. There are no flashing lights or alarms that say the network is functioning properly and securely. If you place an IDS on the outside of the perimeter, you can extract reports based on the traffic the IDS sees. These can be used to demonstrate to management in concrete terms what your security efforts are accomplishing. Saying “the network is running fine” is great, but probably doesn’t have the impact that a one-page report with a pie chart would have. An executive summary of the attacks the sensor has seen could list some basic facts like “56,000 instances of code red worm were blocked, up 5 percent from last month,” and so forth. With an old PC and a little up-front effort, these types of reports would take very little effort to produce, but they could help you reap huge rewards when it comes to public perception of network security.

When exposing *any* system to the Internet at large, remember it will be attacked. If your IDS is outside your perimeter firewall, there is nothing protecting the IDS except the IDS itself. This means the IDS will need to be hardened and secured as much as possible to ensure that it doesn’t become a system for hackers to use. Under these circumstances, one of your best defenses would be for the IDS to use a network tap (not free) to ensure that it can only receive from the network and not transmit. There are various discussions on the Internet for making cables that can receive only. A little research will surely turn up some interesting designs to try. The success of these read-only cables will vary greatly depending on your system’s network card and the switch or hub you are connected to. While this doesn’t make the IDS sensor invulnerable to attacks or alleviate the need to harden it, this configuration will make it significantly harder to compromise.

Summary

Snort has the undisputed position as the lead open source IDS. As such, it enjoys several advantages. One advantage is the very large and diverse user base. This user base enables you to find a lot of help and information on the Internet for running, configuring, and customizing Snort. Although Snort may not enjoy the cohesive turnkey nature of a commercial package, you can

assemble several utilities and tools to make Snort into an enterprise-class IDS. With no cost in software you can have an industry-standard IDS with a large signature base and the ability to create your own custom signatures. Your signatures can be automatically updated to keep them current, and you can use several GUI front ends to remotely configure and manage several Snort sensors at one central location. All this adds up to a lot of value and increased security, with no additional software cost.

Solutions Fast Track

Intrusion Detection Systems

- ☑ The most common way to implement an IDS is by having a system monitor and inspect (sniff) all traffic over a given link.
- ☑ An IDS that compares traffic with a database of known signatures for undesirable traffic is an example of a signature-based IDS.
- ☑ Some IDS systems are *anomaly based*, which means the IDS attempts to build a list of “normal” traffic from your actual network data, and then it flags anything that doesn’t fit the list of normal traffic that it has built.

Configuring an Intrusion Detection System

- ☑ Placement of the IDS will be key. If the IDS is not placed properly you will miss alerts and possibly think you are more secure than you really are.
- ☑ Your IDS is probably the security host that will need the most hardware resources of any discussed in this book (with the firewall being a close second), so plan accordingly when selecting the hardware to use for your IDS.
- ☑ Remember that even with the proper physical placement, you need to have a hub in order for the IDS to be able to see traffic destined for other devices, or enable port mirroring if you are using a switch instead of a hub.

Configuring Snort on a Windows System

- ☑ Remember that every path in the `snort.conf` file needs to be an absolute path. A single incorrect path will prevent Snort from running properly.
- ☑ WinPcap will be required in order to use Snort on Windows. It is also required for using several other networking utilities on Windows.
- ☑ IDS Policy Manager can be used to centrally configure and manage the Snort process and Snort rules.

Configuring Snort on a Linux System

- ☑ You may want to consider a Snort alert front end such as BASE for viewing alerts.
- ☑ If your environment is primarily Windows, this will enable you to access the alerts from the Windows systems without having to view the Snort console on the Linux IDS host.

Other Snort Add-Ons

- ☑ A fully functioning IDS will not be of much value if no one is taking notice of the alerts it generates. An easy-to-use alert console can add a lot of value to your IDS in that it may increase the attention the alerts receive.
- ☑ We recommend using Oinkmaster to automatically keep your Snort signature files current.

Demonstrating Effectiveness

- ☑ One of the age-old debates when it comes to network data collection is placement of the sensors.

- ☑ The most common difference of opinion is whether you should place the sensor outside your external firewall or inside it.
- ☑ Undeniably, the traffic of the most security relevance is the traffic that has managed to traverse your firewall and get into your internal network. These are the potential attacks, probes, and whatnot that need to be inspected and monitored closely to make sure the network is not compromised.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: How do I configure Snort to send e-mail alerts?

A: You don’t. Snort includes no native way to send e-mail alerts. This was an intentional decision because processing e-mail alerts would place an undue burden on the Snort process, possibly resulting in dropped packets and missed alerts. Instead, the simplest way to accomplish this is with a log parsing tool, such as swatch. Swatch and other utilities for log management are covered in more detail in Chapter 7, “Network Reporting and Troubleshooting.”

Q: How do I turn Snort into an IPS instead of an IDS?

A: Snortsam (www.snortsam.net/) is designed to automatically adjust the rules on a firewall based on certain Snort alerts. It is a mature tool with relatively active development. Also check the user-contributed section of the Snort Web site for an assortment of utilities at www.snort.org/dl/contrib/patches/. Snort itself also has some limited capability to take actions, specifically when acting in “inline mode.” Refer to the documentation at www.snort.org/docs/snort_htmanuals/htmanual_260/node7.html for more on Snort’s native IPS support.

Q: How do I make a Snort rule to trigger for “X” application’s traffic?

A: Start by searching online; you can usually find the rule already made for you. If not, the general procedure is to do a packet capture (with Wireshark, for example) and then review the packets. The tricky part is to identify something all the packets (or if not all, at least the initial packet) has in common. Some string that can uniquely identify that application’s packet from any other’s. Then you place this string in the rule using the payload option. See the online Snort manual for more information on rule option fields.

Q: How can I make my Snort sensor more secure?

A: There are many ways. First, configure a firewall on the sensor itself to protect itself. You would only filter traffic with a destination of the sensor, so that you don’t accidentally filter the traffic you want to trigger alerts on. You can also have Snort listen on an interface without an IP address; this will make it a lot harder for an attacker to target the sensor. (See the main Snort FAQ for instructions on how to do this.)

Managing Event Logs

Solutions in this chapter:

- **Generating Windows Event Logs**
- **Generating Syslog Event Logs**
- **Securing Your Event Logs**
- **Applying Your Knowledge**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

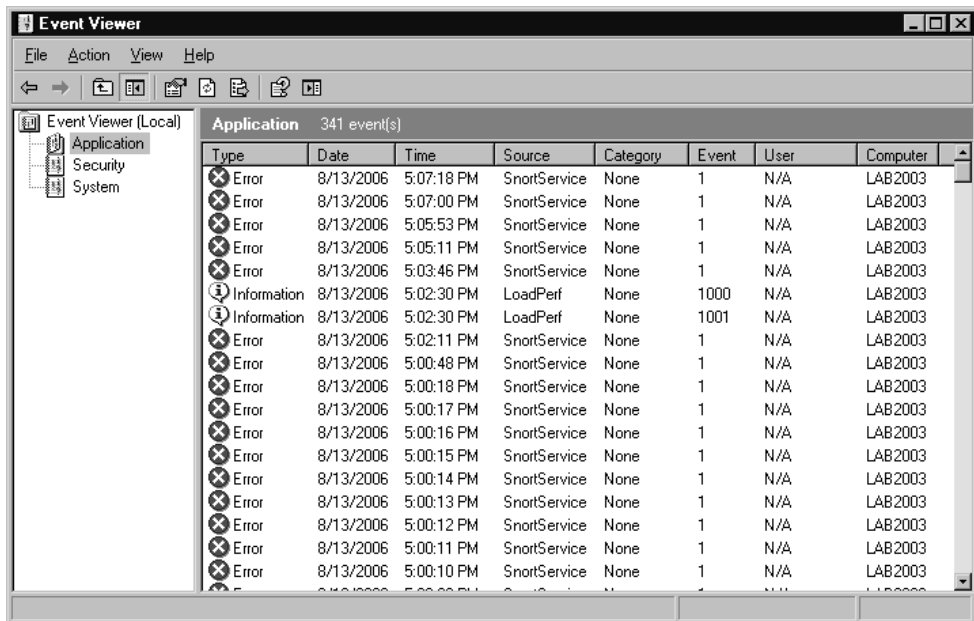
Event logs provide valuable insight into what is happening on a system. The scope of a typical event log is very different from the output of an IDS (intrusion detection system). This is because in general the event logs are more tailored to specific systems and applications. With the right utilities you can even create event log entries for custom applications or batch files/scripts. For example, if you have a batch file that deleted old temp files on a server, you could have it send a log message to the syslog server if there are errors during the delete process. When talking about free event logging (free software to generate the logs and free software to receive the logs), there are basically two formats you are likely to encounter and work with extensively. In the UNIX world, there is syslog, which is the de facto standard when it comes to centralized logging. On the other hand, there are the Microsoft proprietary event logs, which are used only by Microsoft systems. Because Microsoft chooses not to natively support syslog, we will discuss what you *can* do to make the best use of the Windows event logs. In addition to configuring and generating the logs, we will show you how to analyze the logs and create notifications for significant events.

Generating Windows Event Logs

Windows event logs are broken up into six major categories: Application, Security, and System; and on domain controllers: Directory Service, DNS, and File Replication. You can view the event logs on a Windows host by navigating to Start | Run | eventvwr.exe. The Event Viewer is really an MMC console snap-in, and as such you can add it to your MMC console instead of using the standalone Event Viewer if you choose to do so. Application logs are, of course, application specific and rely on the application that is generating the logs to provide meaningful information for the event log. Because of this, support from specific applications can vary greatly. When the application in question is a Microsoft application, event log support tends to be good, and the logs usually provide relatively detailed information. In other cases, the application may not support Windows event logs at all and you may not see the application generate any logs of its own. The Event Viewer is shown

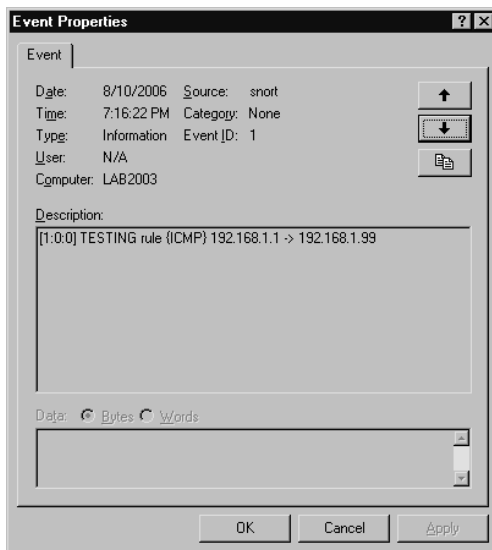
below in Figure 5.1. The Event Viewer is intended to be your primary access to view and save Windows event logs, though there are some additional tools available to manipulate the logs, both from Microsoft and third parties.

Figure 5.1 Windows Event Viewer



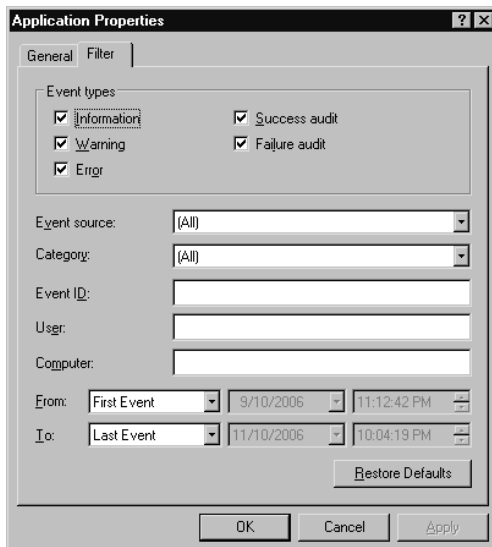
Navigating the Event Viewer is simple. You select the category of events you wish to view on the left-hand pane, and the right-hand pane will display the events. You may double-click an event in the right pane to view the details for that particular event. The Event Properties window will display the complete event entry, including all the summary information as seen on the main Event Viewer window, plus the full description of the event. Sometimes the description can include very valuable troubleshooting information. The Event Properties window is shown in Figure 5.2; you'll notice that Snort for Windows supports logging to the Windows event log.

Figure 5.2 Event Properties



The list of events can be quite lengthy. Fortunately, there are mechanisms to enable you to filter and search for events. If you wish to filter the list of events that are displayed, select View | Filter. This enables you to enter criteria to filter the event list as shown in Figure 5.3.

Figure 5.3 Filtering the Windows Event Log



Most of the fields are self-explanatory. The Event ID is often the quickest way to filter the list for a specific event *if* you happen to already know the event ID you are looking for. The User field is the user a process was running under when it generated the event. In most cases this will simply be NA or SYSTEM, which isn't of much use. This field is most useful when viewing the security log. After you have configured the filters to your satisfaction, click Apply and OK. The event list will be filtered immediately. To remove the filter and display all events, go back to the filter list and click Restore Defaults.

TIP

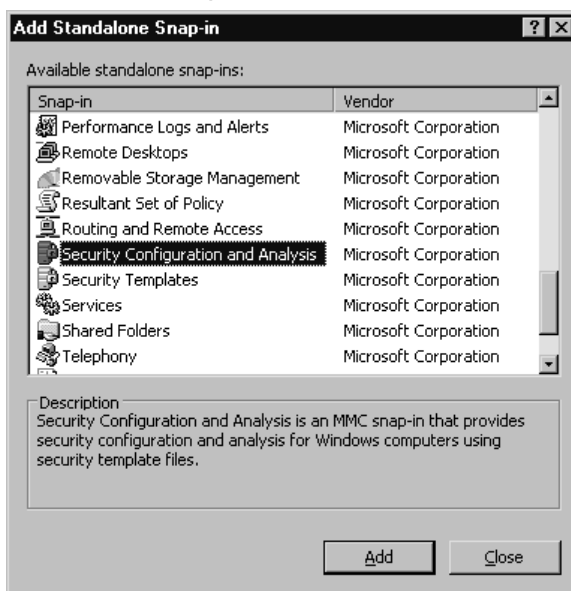
The event log details can sometimes be cryptic. If the meaning of a particular event is not clear, there are many places on the Internet to get information on particular events (including www.Microsoft.com). A handy site for investigating events is www.eventid.net. This site includes a form to enter the event ID and source, as well as links to search for information for that event on Google and Microsoft's Web pages.

Using Group Policy to Generate Windows Events Logs

By default, Windows systems will generate event log entries for certain types of events automatically. Individual applications will generate events based on their programming. While the level of logging is adequate for day-to-day use, there are a lot of additional things you might wish to generate log entries for, particularly when it comes to security events. You can configure the operating system to generate security-related event logs relatively easily by applying policies to enable auditing via the Microsoft Management Console (MMC). You can access the console by navigating to Start | Programs | Administrative Tools. All the listed utilities, such as Services, are actually MMC snap-ins. You can also customize your MMC console and add the snap-ins of your choice instead of just viewing the snap-ins individually. You will need to load the Security Configuration and Analysis snap-in to configure auditing. To do this, go to **Start** | **Run**, type **mmc**, and then click **OK**. This will open up the full

MMC. The first time you do this the MMC will have no snap-ins loaded. Click **Console | Add/Remove Snap-in**, and then in the **Add/Remove Snap-in** dialog box, click the **Add** button. In the **Add Standalone Snap-in** dialog box, select the **Security Configuration and Analysis** snap-in (see Figure 5.4), click the **Add** button, and then click **Close**. Click **OK** to close the Add/Remove snap-in dialog box and complete the process. This process can be repeated to add multiple snap-ins to your MMC console.

Figure 5.4 Add Standalone Snap-In Screen



Now that you have the snap-in added to your MMC, the next step is to create a new security configuration database. Do this by right-clicking the **Security Configuration and Analysis** snap-in and selecting **Open Database**. The first time this is done, there will be no security database; you will need to create a new one. Type a name ending in **.sdb** and click **Open**. The next window enables you to choose a security template to import. Select **Setup Security.inf** and click **Open**. After this is done, you need to right-click **Security Configuration and Analysis** again and select **Analyze Computer now**. On the next window, you should choose a path and file-name for the log file (or accept the default) and click **OK**. After the analysis is completed, this will reveal several expandable items under Security Configuration and Analysis.

The audit policy settings can be found by navigating in the MMC to **Security Configuration and Analysis\Local Policies\Audit Policy**. You can review Microsoft's recommendations for securing Windows 2000, including audit settings, here: www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/default.mspx. For Windows 2003 you can refer to this article: <http://www.microsoft.com/technet/security/prodtech/windowsserver2003/w2003hg/sgch00.mspx>. Some of the most significant audit options are outlined below, along with some recommendations for the most secure settings. Remember that enabling extensive auditing can have an adverse impact on the performance of the host system. You must carefully weigh the benefits of having the audit trail versus the resources that will be consumed by enabling various audit options.

- **Audit account logon events** There is a subtle yet important difference between “audit *account* logon events” and “audit logon events.” The account logon events occur only when a local account is used to log on to another computer. For standalone servers in isolated environments, this event should never be triggered. You will see the most logs generated from this type of auditing when it is used on domain controllers.
- **Audit account management** Keep this set to Success, Failure, which tracks when passwords are changed, accounts are created and deleted, and group membership changes. The audit records generated by this setting should be closely monitored for signs of unauthorized activity, which can indicate a system compromise.
- **Audit directory service access** This will create event log entries for active directory events, which can be useful for spotting security issues.
- **Audit logon events** Set to Success, Failure. This is the single most important audit item. These audit events will let you know *who* and *when* someone is trying to log on, and perhaps even more importantly, if they are succeeding. This setting will provide glaring evidence of brute force logon attempts and hopefully give you a jump start on investigating the source and taking corrective action.

- **Audit object access** Keep at Success, Failure. This is another important audit setting. It will record access to objects such as printers, registry keys, folders, or individual files. Basically, it will audit anything with a system access control list (SACL) defined.
- **Audit policy change** Keep at Success, Failure. This will trigger when someone modifies (or attempts to modify) your policies. An attacker who has successfully compromised a system may attempt to alter the system policy, to give him more freedom over the compromised system.
- **Audit privilege use** According to Microsoft, this setting will “Audit each instance of a user exercising a user right.” This might sound handy at first, but in reality it often generates far too much data to be useful. If your environment justifies enabling auditing on these events, setting it to audit only failures will usually suffice. Most often this is set to No auditing on production systems and enabled only for special troubleshooting activities.
- **Audit process tracking** Similar to “privilege use,” this setting can generate a large amount of data and can have a significant impact on performance. In most cases you should set it to No auditing, unless you’re diagnosing specific problems.
- **Audit system events** Keep at Success, Failure. This event triggers only for events affecting the entire system, such as system start and shutdown or for events that affect the system security or security log.

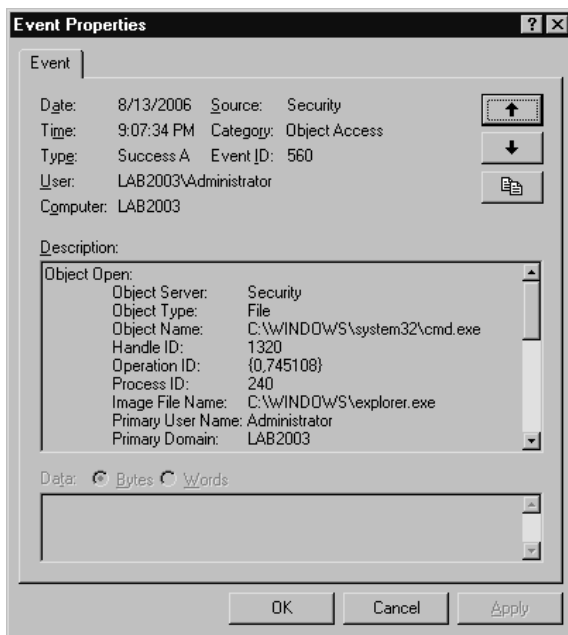
All of these will cause audit events to be generated and recorded to the event log. Another item for which it may be useful to generate audit events is access to specific files. With some files, and especially executables, you may find it advantageous to configure the creation of audit events when the file is accessed. To enable auditing on specific files, follow these steps:

1. In the left-hand pane, under Security and Configuration Analysis, select File System.
2. Navigate to the directory containing the file you want to audit and select it in the left pane. In this example will use cmd.exe, located in `%systemroot%\system32\`.

3. Locate **cmd.exe** in the right pane and double-click the file.
4. Click **Edit Security**.
5. Configure the access settings for various accounts to your preferences.
6. Click **Advanced**.
7. Select the **Auditing** tab at the top of the window.
8. Click **Add**.
9. Select the users and groups you wish to audit when accessing CMD.EXE and click **OK**.
10. Place a check mark next to any events you wish to audit; for example, **Traverse Folder/Execute File**.
11. Click **OK** four times to close all the windows.
12. Navigate to **Security Configuration and Analysis | Local Policies | Audit Policy** in the left pane.
13. In the right pane, double-click **Audit Object Access** and place a check next to **Success** and **Failure**.
14. Click **OK**.
15. Right-click **Security Configuration and Analysis** in the left-hand pane of the MMC and select **Configure Computer Now...**
16. Choose a name and location for the log file, and then click **OK**.

In this example, we checked **Traverse Folder/Execute File** for both **Success** and **Failure** for the Everyone group so that any time cmd.exe is executed it will generate an event in the event log. A sample event documenting successful access to cmd.exe by administrator is shown in Figure 5.5.

Figure 5.5 Sample Audit Event



As you can see, with a little customization you can cause the Windows event logs to generate a lot of auditing data. While the built-in auditing and logging capabilities are impressive, sometimes you may need still more flexibility when it comes to generating entries in the event log.

If you wish to enable auditing policies for all clients that log into your domain controller, you can apply the policy on the domain controller using group policy objects. These policies will be applied to computers as they are authenticated to the domain. The process for enabling them at the domain level is very similar to doing it at the computer level. You can configure the policies to be applied throughout the domain or to specific *organizational units* (OUs) within AD. To enable auditing on all computers within the domain, using global policy objects, follow these steps:

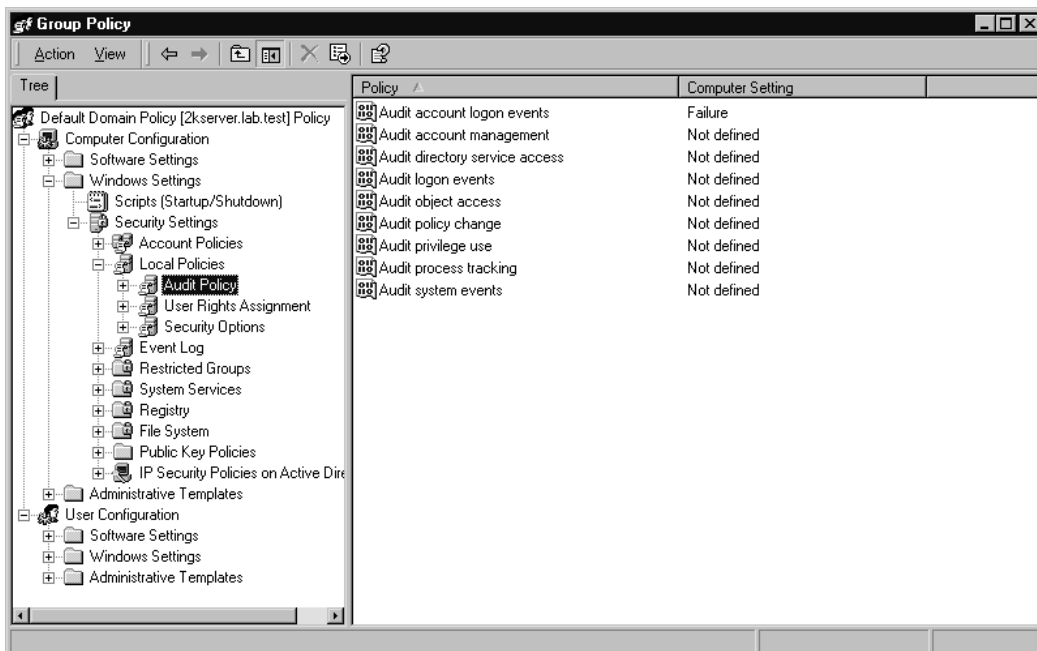
1. Navigate to Start | Programs | Administrative Tools | Active Directory Users and Computers.
2. Right-click the desired domain name and select **Properties**.
3. Select the **Group Policy** tab.
4. Click **New**.

WARNING

It is considered best practices to create a new policy object instead of editing the *default* domain policy. This is a safety precaution in case you run into a problem with the new policy changes. In an emergency situation, you can simply disable the newly created policy (**Group Policy** tab, click **Options**, check **disabled**;) instead of trying to troubleshoot the specific setting that is causing problems.

5. Enter a meaningful name of the new policy and press **Enter** to accept the name.
6. Highlight the Newly Created Policy and select **Edit**.
7. Configure the desired audit policies following the aforementioned guidelines for configuring a local audit policy (see Figure 5.6).

Figure 5.6 Configuring Audit Using Domain GPOs



Configuring your policies at the domain level also allows for some additional settings that are not available using a local computer policy object, such as event log settings. By navigating to **Default Domain Policy | Computer Configuration | Windows Settings | Security Settings | Event Log | Settings for Event Logs**, you can (and should) configure settings related to how the event log itself is accessed. The following list summarizes the event log policy settings that have a high impact on security.

- **Restrict Guest Access to <log type> Log** This should be set to enable to prevent a potential attacker who manages to gain guest access from gaining access to the event logs.
- **Retention Method for <log type> Log** This determines what is to be done with old log events. By default the event log will overwrite old events as needed. This method ensures that the system will not shut down due to an inability to log the security events, and this should be acceptable as long as you are collecting the important events on another secure system. The objective is to ensure that important security events cannot be lost because an attacker generates so many events that the incriminating ones are wrapped and lost. Shutting the system down due to running out of space, possibly even with the intent of creating a denial of service attack (versus legitimately attempting to break into the system) are most likely if you use the setting “Do not overwrite events (clear the log manually).”

Generating Custom Windows Event Log Entries

Granular per-file auditing enables a lot of control of when events are generated; however, suppose you were in the following scenario: You have a custom script or batch file that runs regularly and you would like to create event log entries based on the actions of that script. Auditing based on file access could create an event any time `cmd.exe` was accessed, but it wouldn't be able to know anything about the script files actions, only that `cmd` was accessed. In this case you need to be able to manually create event log entries containing your own custom data.

Logevent.exe (Windows 2000 and Windows NT) and eventcreate.exe (Windows 2003/XP) will do exactly that. Various command-line switches will enable you to specify the event data, the log to write to (system, application, or security), the event source, the event type, and the event ID to use. The following is an example command:

```
Eventcreate /T ERROR /ID 123 /L SYSTEM /D "Secure your network for FREE!"
```

This would create an event in the system log of type ERROR, with an ID of 123, and the contents “Secure your network for free.” Note that you cannot create custom events in the security log. Using this utility you could have portions of your custom script generate event log entries based on the script’s internal processing.

Collecting Windows Event Logs

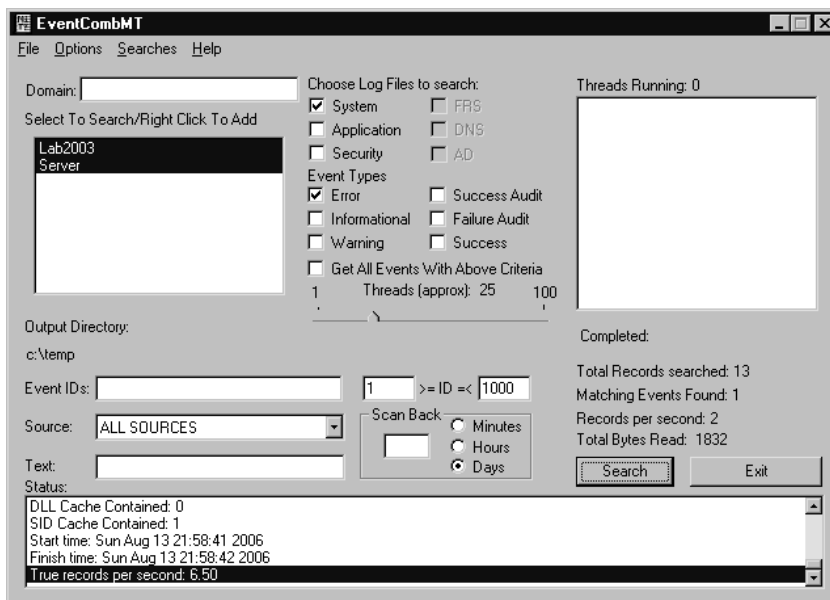
The MMC will enable you to view the event logs from remote computers. While useful, this is a manual process and far from ideal when it comes to day-to-day monitoring of your entire network. The best solution is to find a means to get all your logs from all your systems in one place. Then the logs can be centrally parsed, filtered, analyzed, and processed. Windows is somewhat limited when it comes to collecting event logs. Most of the feature-rich applications to do so are created by third parties and are commercial products. Eventlog.pl (Windows 2000 Resource Kit, supplement) will enable you to back up the event logs and export them as text files. These text files could then be moved to a central system and analyzed. Eventquery.vbs (Windows 2003/XP) enables you to filter and list events from a local machine or a single remote machine.

Another option is EventCombMT (from Microsoft). This has a GUI interface which will enable you to basically search remote event logs for events that match your criteria and log them to a text file. EventCombMT will search multiple systems simultaneously, while eventquery.vbs must be run for each system to be queried. EventCombMT is available as part of the Windows Server 2003 Resource Kit and the Windows Server 2003 Deployment Kit. The Windows Server 2003 Resource Kit Tools can be downloaded here:

www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&DisplayLang=en

In Figure 5.7, you can see the EventCombMT program window. The configuration and search criteria are pretty straightforward. By default, EventCombMT will log the output from its searches to C:\temp\. EventCombMT.txt provides summary information on the query, such as how many servers were searched and what search criteria was used. Additional logs will be created of the format <hostname>-<log>.txt.

Figure 5.7 EventCombMT



For example, we searched for events with an event ID between 1 and 1000. The resultant output was logged to C:\temp\Lab2003-System_LOG.txt. The log contained the following entry:

```
123,ERROR,system,Sun Aug 13 21:25:32 2006,LAB2003\Administrator, Secure your network for FREE!
```

```
The longest gap between all scanned records occurred at Sun Aug 13 21:25:32 2006 and was 0 days, 0 hours, 28 minutes, 18 seconds.
```

```
c:\temp\Lab2003-System_LOG.txt contains 1 parsed events.
```

As you can see, EventCombMT found the sample entry we made earlier using EventCreate.exe. The sample event had an event ID of 123.

An additional tool provided by Microsoft is Log Parser, which can be downloaded from www.microsoft.com/downloads/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en. Log parser is more of a general-purpose utility that can parse any log, not just Windows event logs. With the increased flexibility comes increased complexity. The syntax for using Log Parser (a command-line utility) can be complex at times and resembles that of SQL queries. Entering `logparser.exe -h examples` will generate a lengthy list of sample commands. If you review the documentation you will see that Log Parser is indeed very flexible and powerful but not for the faint of heart.

When it comes to collecting your Windows event logs in a centralized location in their native event log format, you are really limited to using a Microsoft host. This is because a Linux system cannot work with the event logs directly. Because of this, if you want to manipulate, filter, or otherwise process the Windows event logs on a non-Microsoft logging server, the best bet is to export them in a commonly understood format, such as syslog. The use of syslog will be discussed in depth later in this chapter.

Analyzing Windows Event Logs

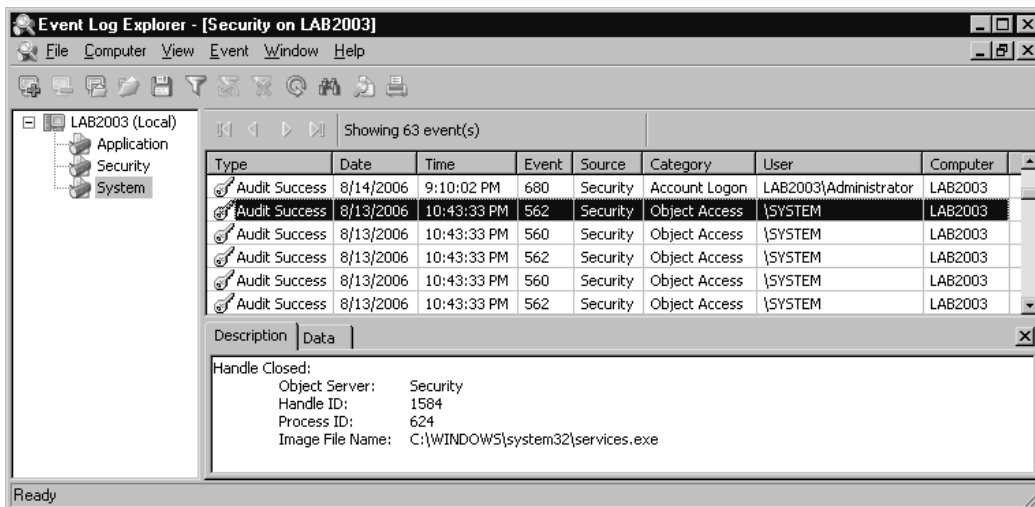
Microsoft provides some tools that can be used to sort through the large volume of event logs you are likely to generate. `Eventlog.pl` (Windows 2000 Resource Kit, supplement 1) enables you to back up event logs, export event lists to text files, clear event logs, and query properties of event logs. `Eventquery.pl` (Windows 2000 Resource Kit, supplement 1) enables you to display and filter event logs based on various fields such as time, date, source, category, ID, et cetera. Both of these could be scripted to sort through event logs and generate a listing of the most serious security issues. If you wish to do the processing on the host that is generating the logs, you can always use something like `eventtriggers.exe` (Windows 2003/XP), which enables you to configure a process to be executed when certain events are logged.

TIP

Remember that if you set up an event trigger to send an e-mail when a local account is locked out (for example), if a hacker manages to compromise the local system, he may be able to disable your event triggers. This means that from a security perspective, *exporting* the logs and then parsing them is going to be more secure (and preferable) than performing any sort of log analysis on the machine that is generating the logs.

If your event logs stay in Windows event log format, there are some third-party tools to help manage and view them centrally. Event Log XP is one such tool available from www.eventlogxp.com. It uses a simple Installation Wizard and provides some very useful functionality. The main screen is shown in Figure 5.8 with the System Log window currently displayed.

Figure 5.8 Event Log Explorer



If you select an event in the right-hand pane and right-click, you have the option to look up the event in an online knowledgebase—either the Microsoft knowledgebase or EventID.net. This can be handy when you're not

sure exactly what a specific event ID means. Another useful feature is the capability to configure an automatic refresh by navigating to View | Auto-refresh. If you wish to change the refresh interval, you can access the setting by navigating to View | Auto-refresh Interval. You can access a robust filter mechanism by navigating to View | Filter. The filters will enable you to quickly check a system's logs for events of particular importance. The downside to the Event Log Explorer is that it displays each log for each system in a separate window. This could make navigating a large number of logs tedious, though still better than doing it using Event Viewer in the MMC.

A good free utility targeted at inspecting IIS logs is NTLlast, available from www.foundstone.com. Unfortunately, free solutions that offer quality features for analyzing Windows event logs are few and far between. For serious event log parsing *without paying* for a commercial product, we would recommend converting the Windows event log to the syslog format and then analyzing the syslog logs on a Linux host. This architecture will offer the most functionality and options for log collection and analysis while providing the best performance. Even if you choose to perform your analysis on a Windows host, you will find that there is a wealth of *free* utilities for parsing logs that use the syslog format; the same cannot be said for the Windows event log format.

Generating Syslog Event Logs

Right out of the box, virtually every Linux system will be configured to generate a generous amount of logging information. In most cases, not only will syslog logs be generated with a default installation, but often a syslog server will be running unless you have specifically disabled it. Even if you choose to host your syslog server on a Windows system, there are far more options for analyzing syslog logs than Windows event logs (largely due to syslog's ASCII data format no doubt). Due to the ubiquitous nature of syslog logging, it would be advisable to familiarize yourself with some basic syslog concepts.

Syslog works in a client-server fashion. The server (or daemon in UNIX/Linux parlance) receives the messages that are generated by remote hosts, or even the same host. The client piece compiles the messages with the proper data and sends them to the server. A syslog entry is recorded all on a single line. Each entry consists of a *facility* and a *severity* level. The *facility*

loosely tells you what component subsystem generated the message, while the *severity* tells you the general severity or importance of the message. We use terms like “loosely” and “general” because not all applications will be written the same and one may think a given message is more important than another. When you are manually generating the messages you can typically enter facility and severity levels of your choosing. Table 5.1 provides a summary of the different severity levels that are available, in order from most severe to least severe. Note that while the severity is listed as LOG_EMER, it would appear using slightly different formats depending on the software you are using, such as emer, emerg, or even emergency.

Table 5.1 Syslog Severity Summary

Syslog Severity Summary		
Linux Syslog Severity	Numerical Identifier (Level)	Usage
LOG_EMER	0	Emergency : This indicates a panic condition. Emergency-level messages are usually broadcast to all users on a system and indicate the most severe of conditions.
LOG_ALERT	1	Alert : Indicates a condition that should be corrected immediately, such as a corrupted system database.
LOG_CRIT	2	Critical : Indicates a condition such as a hardware device error.
LOG_ERR	3	Error : Is the generic error level for a message that does not specify an alternate severity..
LOG_WARNING	4	Warning : These messages represent non-critical events or conditions.
LOG_NOTICE	5	Notice : Represents conditions that are not error conditions, but should possibly be handled specially. This type of level may be useful for manually generated messages.
LOG_INFO	6	Informational : These messages typically don't receive any special attention unless you are looking for a specific check or piece of information.
LOG_DEBUG	7	Debug : These messages contain very detailed information normally of use only when debugging a program or process. Due to the high volume of data that can be generated, debug-level logging is usually only enabled to troubleshoot a specific issue.

Some systems and utilities will refer to the various levels by name while others will use the numerical representation of the level. Because most non-Windows systems support syslog natively, they typically are configured with

default facilities to use in the event that the source of the message does not explicitly define one. For example, if *login* or *su* generates a syslog event without specifying the facility, it will automatically be assigned to the LOG_AUTH facility. Custom messages are typically assigned a facility of LOG_LOCAL1 through LOG_LOCAL7, also referred to as simply local1 through local7. The facility is written as *facility.level*, such as debug.local1 for example.

The syslog server typically listens on UDP port 514. Unless you take steps to secure the communication, the syslog messages will be sent in cleartext between the generating host and the syslog server. Frequently, the syslog messages will contain sensitive or at least confidential information, such as server names, the names of software you are running, and IP addresses. See Figure 5.9 for a capture of a test syslog message using NGsniff. We will discuss the use of sniffer software in Chapter 7, “Network Reporting and Troubleshooting.”

Figure 5.9 Syslog Cleartext



```
C:\ngSniff-1.3>ngsniff --interface 0
ngSniff v1.3 by NGSEC Research Team <labs@ngsec.com>
FREEWARE command line sniffer
Next Generation Security Technologies
http://www.ngsec.com
```

```
Sniffing...
IP HEADER 192.168.1.107 -> 192.168.1.99
-----
IP->version: 4
IP->ihl: 5
IP->tos: 0
IP->tot_len: 106
IP->id: 62345
IP->frag_off: 0
IP->ttl: 128
IP->protocol: 17
IP->checksum: 28972
```

```
UDP HEADER
```

```

-----
UDP->sport: 1733
UDP->dport: 514
UDP->ulen: 86
UDP->checksum: 6598

----- Begin of data dump -----
3c 39 3e 41 75 67 20 31 35 20 31 39 3a 32 30 3a <9>Aug 15 19:20:
35 38 20 65 76 65 6e 74 63 72 65 61 74 65 5b 77 58 eventcreate[w
61 72 6e 69 6e 67 5d 20 32 34 36 20 4c 41 42 32 arning] 246 LAB2
30 30 33 5c 41 64 6d 69 6e 69 73 74 72 61 74 6f 003\Administrato
72 20 20 54 65 73 74 20 45 76 65 6e 74 20 r Test Event
----- End of data dump -----

```

There are a variety of methods available to secure the syslog communications. We will walk you through setting up a syslog server on a Windows host and on a Linux host. We will also demonstrate some of the software that can be used to filter and analyze the syslog logs as well as the steps needed to encrypt your syslog data between the syslog client and syslog server.

Windows Syslog

Although it is not native to Windows, syslog has been around since the 1980s and is so prevalent that the inevitable Windows support has surfaced. When you are researching the products to use for Windows-based syslog, remember to differentiate between the syslog server (or daemon) software and the client software. Because the software to send and receive syslog messages is typically not the same software package, we will discuss each in turn. We will cover the various ways to generate your events to send to a syslog server, for both Windows and Linux, as well as the software that is used to receive the event messages (again on Windows and Linux).

Generating Syslog Events

The approaches to generating syslog messages from a Windows host basically fall into two broad categories. The first is to generate the message natively in syslog format and send it to the syslog server. Because current Windows offerings don't support syslog natively, you need to manually generate these mes-

sages in batch file or other type of script. While this might be a good solution for one-off processes or custom applications, odds are you will have a large number of items that can only log to the Windows event log, which brings us to the second alternative.

Perhaps the “cleanest” option of all is not to try to generate syslog events natively, but to use a utility to convert the Windows event logs into a syslog-compatible format and then process them in that format. Using this methodology will not only be easier to configure, but will also help ensure you don’t have events that are missed. We have found three quality free options to do this. In increasing order of complexity and functionality is the Eventlog to Syslog Utility (evtsys), NTsyslog, and SNARE.

The Eventlog to Syslog Utility is available at the following link: <https://engineering.purdue.edu/ECN/Resources/Documents/UNIX/evtsys/> Evtsys is the simplest option in terms of both configuration and features. Evtsys requires virtually no setup, but will only forward events with a facility of DAEMON and a priority of ERROR, WARNING, or NOTICE. In many circumstances this may be all you need, but for more control over what is forwarded there are other alternatives.

NTsyslog represents the middle-ground when it comes to complexity and functionality. To be fair, it isn’t hard to set up at all, but there are a few more options than evtsys. NTsyslog runs as a service for Windows NT/2000/XP and will reformat the event logs to the one-line syslog format and send them to a syslog server as they are generated. NTsyslog allows you to configure what types of events you wish to export and what facility and severity to label them as (Figure 5.10 shows the forwarding settings window). NTsyslog can be downloaded from <http://sourceforge.net/projects/ntsyslog/>. Because NTsyslog is a good middle ground and will likely be adequate for most installations, we will walk through setting up NTsyslog.

Similar functionality to NTsyslog is offered by SNARE, which is available for free from www.intersectalliance.com/projects/SnareWindows/. While it is a little more complicated to set up than NTsyslog, SNARE can provide additional control over which events get exported to the syslog server. Effectively, SNARE enables you to do some robust upfront filtering at the host before generating the syslog traffic, whereas NTsyslog only enables a less-flexible filtering mechanism based on the event type and event log. This can be advan-

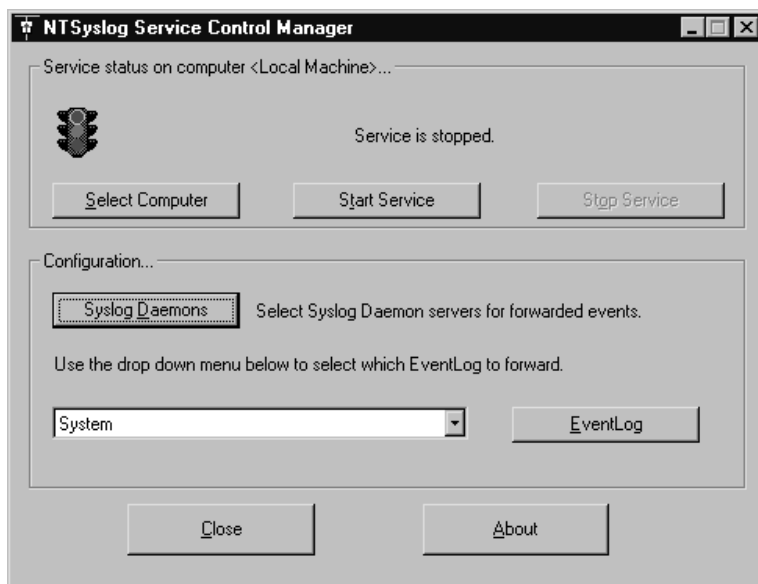
tageous when dealing with a large number of event log events. Between the three of them, they should be able to provide a good selection between functionality and ease of use. You should research and test all three products and determine which one best meets your needs.

Configuring NTsyslog

To configure NTsyslog, follow these simple steps:

1. Download and install NTsyslog.
2. Open a command prompt and navigate to the directory where you installed NTsyslog.
3. Enter **ntsyslog -install**.
4. Enter **ntsyslogctrl** to start the GUI NTsyslog Service Control Manager (shown in Figure 5.10).
5. Click **Syslog Daemons** and enter the name or IP address of your syslog server and click **OK**.
6. Verify that you wish to save the syslog parameters by clicking **Yes**.
7. Click **Start Service**.

Figure 5.10 NTsyslog Service Control Manager

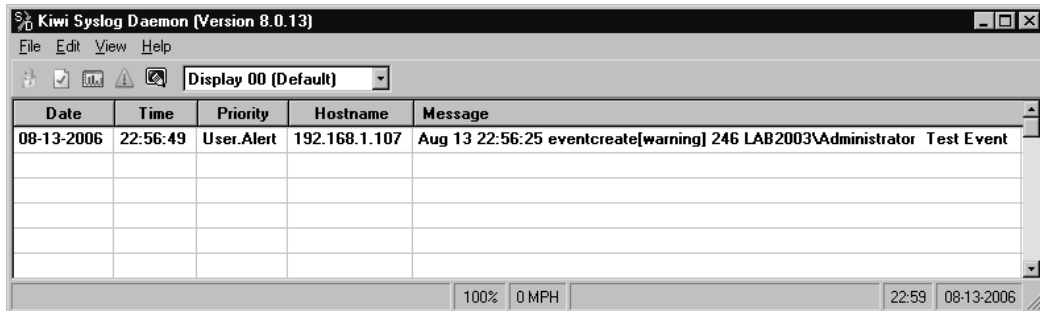


You should now be able to receive your Windows event logs on your syslog server. We used *eventcreate.exe* to create an artificial event for testing with the following syntax:

```
Eventcreate /T WARNING /ID 246 /D "Test Event"
```

The resultant event in my syslog console is shown in Figure 5.11.

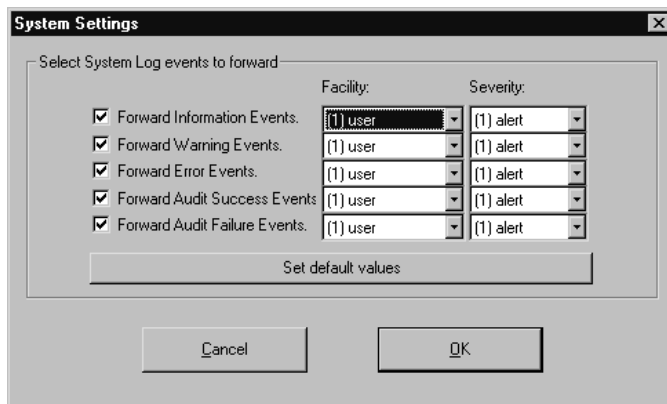
Figure 5.11 Sample Event from NTsyslog



Date	Time	Priority	Hostname	Message
08-13-2006	22:56:49	User.Alert	192.168.1.107	Aug 13 22:56:25 eventcreate[warning] 246 LAB2003\Administrator Test Event

If you so desire, you can configure NTsyslog to forward the events from only certain logs (such as the security log, for example), and to forward only certain types of events. You can also configure NTsyslog to forward the events using the facility and severity of your choosing, rather than the default translated values. Figure 5.12 shows the System Settings window, which will make the Forwarding configuration a lot clearer. Figure 5.12 shows the properties for the system event log. Each Windows event log can be configured separately, with the same level of control.

Figure 5.12 Forwarding Settings



Encrypting Syslog Traffic

Often, the systems we wish to monitor are in un-trusted networks, such as a Web server in an Internet-facing DMZ, or a DNS server. Because the messages could contain sensitive information, we should encrypt the data to maintain confidentiality while in transit. This is where the choice of using syslog over UDP or TCP becomes important. Your choice of encryption mechanisms changes dramatically based on whether you plan to use UDP or TCP as the underlying protocol. You *can* encrypt traffic with TCP or UDP, but TCP's reliability makes encryption much simpler to implement. Therefore, many more options exist for implementing encryption of arbitrary data over TCP than UDP. Syslog uses UDP port 514 by default. There are both syslog clients and syslog servers that can use TCP instead of UDP. The KiwiSyslog daemon (Server) is free and can receive syslog messages over either UDP or TCP.

The problem, however, is that at the time of this writing we could not find a *free* syslog utility that will convert the Windows logs to syslog format *and* use TCP. SNARE will use TCP, but only for the commercial version. Although there are many ways to encrypt TCP traffic using free solutions, the same is not true for UDP traffic. There is a version of TLS (Transport Layer Security) in the works, called DTLS (Datagram TLS), which will be able to encrypt over UDP, but there is no working implementation as of yet. TLS is the successor to SSL (Secure Sockets Layer), which is the standard for encrypting Web pages (i.e., <https://>). The Adiscon logger (www.monitorware.com/en/logger/index.php) is a free *command-line* utility for generating log messages that will support sending syslog over TCP or UDP.

If you have the ability to send your Windows event logs to the syslog server using TCP, there are a couple of encryption options that are easy to set up. One way is to use SSL. SSL is a very well-documented and simple-to-implement encryption for almost any TCP-based communications. Stunnel is open-source software available from www.stunnel.org. Stunnel can tunnel

TCP communications through an SSL-encrypted session with a minimum of configuration complexity. Yet another industry-standard option is to use SSH and its port-forwarding capability. In practice, this works very similarly to Stunnel, but SSH is a different encryption mechanism. The end result is the same, the capability to forward a TCP connection through an encrypted tunnel.

If you are limited to using UDP-based syslog architecture, there is still hope. You will need to set up an IP Security (IPsec) tunnel between the client and server. Although a little more complicated to configure, IPsec is the industry standard when it comes to setting up VPNs, which is just another form of encrypted tunnel. IPsec also has a major advantage in that it is protocol independent. IPsec can encrypt TCP, UDP, and even ICMP, basically anything that runs over IP with a few caveats. In the case of sending syslog messages, we only need the syslog communications encrypted, but IPsec can be used to encrypt *all* communications between a given host and destination. The primary downside to using IPsec over one of the previously mentioned solutions is of course the configuration complexity. Additionally, the out-of-box support for modern Windows systems (Windows 2000, Windows XP, and Windows Server 2003) is better (easier to configure) than that of a Linux host. Although Linux supports IPsec natively, the task of configuring it is made much simpler with the assistance of some third-party configuration utilities. We will demonstrate how to set up IPsec between the syslog server (in this case running on Windows XP) and the syslog client generating the syslog messages (in this case running on Windows Server 2003).

As a final option, if you happen to be sending your syslog messages from one Windows host to another Windows host (not likely given Windows' lack of syslog support), then you can use a free syslog tunneling application from Kiwi Enterprises. Kiwi Secure Tunnel can be downloaded from www.kiwisyslog.com/secure-tunnel-info.php. Kiwi Secure Tunnel supports the tunneling of both TCP- and UDP-based syslog, but there is no Linux software component. The most likely scenario where you could take advantage of this would be when you have several systems (including Linux hosts) generating syslog messages from a secure subnet, but the messages must traverse an insecure network, such as the Internet, to be received at a central location. In this case, all the hosts could send their logs to the Windows host

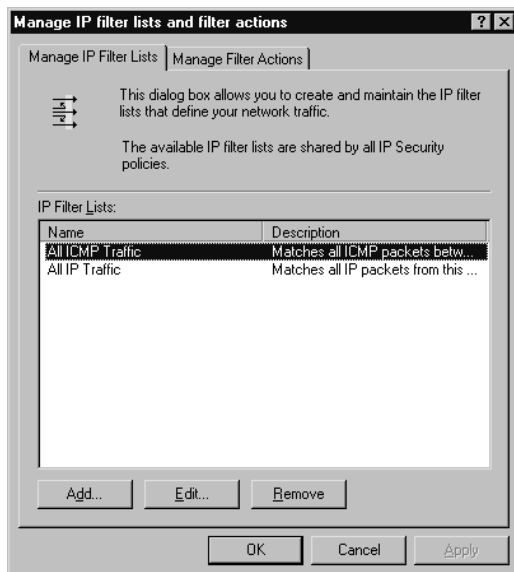
at the remote location, which could then tunnel the messages through the Internet and back to a syslog server at a central office.

Encrypting Syslog Traffic Using IPsec

For starters, let's cover some basic IPsec terminology. IPsec settings are controlled on a Windows system via the *IPsec policy*. Only one policy can be applied to a given host at a time. The policy is defined using *IPsec rules*. A rule defines what types of traffic to act on and whether the traffic is permitted, blocked, or encrypted. The rules also determine how to authenticate the IPsec peer and other encryption settings. *Filters* are used to identify what types of traffic should be processed by the IPsec policy. On some systems this is referred to as defining *interesting traffic*. *Security methods* are used to define the encryption and hashing algorithms to be used. Follow these steps to configure IPsec on the syslog client and syslog server.

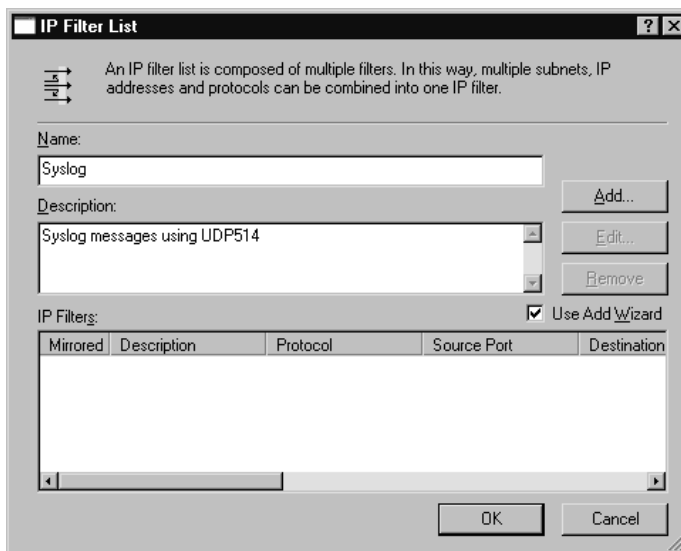
1. Open the MMC and add the **IP Security Policies** snap-in to your console if it is not already present.
2. Right-click **IP Security Policies on Local Computer** in the left pane and select **Manage IP Filter lists and filter actions** (see Figure 5.13).

Figure 5.13 Manage IP Filter Lists and Filter Actions



3. On the **Manage IP Filter Lists** tab, click **Add** at the bottom. This will bring up the **IP Filter List** window as shown in Figure 5.14.

Figure 5.14 IP Filter List

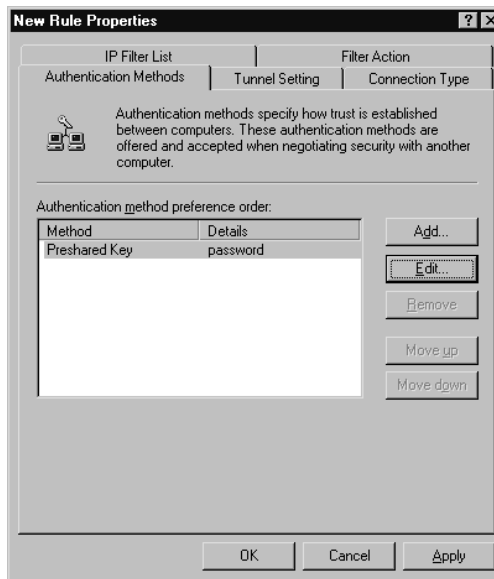


4. Enter a name for the filter; in this case we used “syslog.”
5. Enter a description for the filter.
6. Ensure that the **Use Add Wizard** check box is checked and click **Add**.
7. Click **Next** on the **IP Filter Wizard** welcome screen.
8. Enter a description of the filter and ensure that the **Mirrored** option is checked (it should be by default) and click **Next**.
9. On the next window, leave the source address at the default of **My IP Address** and click **Next**.
10. On the next window, for **Destination address**, select a **specific IP address**, enter the IP address of the syslog server, and then click **Next**.
11. For **Select a protocol type**, select **UDP** and click **Next**.
12. For **Set the IP protocol port**, leave **From any port** selected and select **To this port**.

13. Enter **514** in the **Port** box and click **Next**.
14. Click **Finish**, which will take you back to the **IP Filter List** window. There should be a new IP Filter in the bottom section of this window. This filter will match against any outbound UDP traffic with a destination IP of the syslog server and a destination port of 514.
15. Click **OK** to go back to the **Manage IP filter lists and filter actions** window. Click **Close**. We now have our filter defined, which will tell the system what traffic should be processed by the IPsec policy. We now must create that policy.
16. Right-click **IP Security Policies on Local Computer** in the left-hand pane and select **Create IP Security Policy**.
17. Click **Next** to begin the wizard.
18. Enter a **Name** and **Description** and click **Next**.
19. In **Requests for Secure Communications**, leave **Activate the default response rule** checked and click **Next**.
20. Choose your Authentication method for the default response rule. Active directory is the default and will be the best choice in most circumstances. However, some systems, such as DMZ hosts, may not have domain connectivity, and instead may be standalone servers. In those cases you will need to use certificates or a preshared key. A preshared key is basically a password and is the weakest of the options available; however, it is also the simplest to implement. After making your selection click **Next**.
21. Leave the check box checked to **Edit Properties** and click **Next**.
22. On the **Rules** tab, ensure that the **Use Add Wizard** is not checked and click **Add** to add an IP filter.
23. In the list of premade filters you should see the syslog filter you created earlier. Select the radio button for the filter you created earlier.
24. Select the **Filter Action** tab and select **Require Security**.
25. Click the **Authentication Methods** tab and click **Edit**.

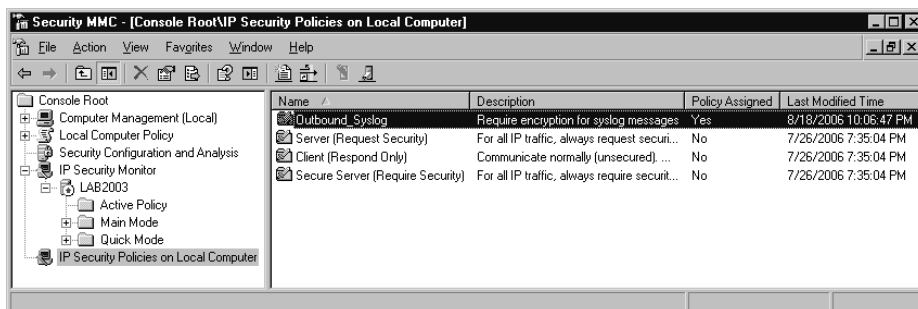
26. Select the radio button for the authentication method you want to use, and then click **OK**.
27. On the **New Rule Properties** screen shown in Figure 5.15, click **Apply** and then **OK**.

Figure 5.15 New Rule Properties



28. You will be back at the **Security Rules** screen; click **OK**.
29. Your Policy should now appear in the list in the right pane of the MMC window. Right-click this new policy and select **Assign**. After the policy is assigned, the MMC window should look similar to that shown in Figure 5.16.

Figure 5.16 Assigning IPsec Policy



Now all that is left is to configure the IPsec policy on the syslog server. On the server side, you need to perform a similar configuration; however, there are some implementation details to consider before settling on your configuration. For example, if all the systems connecting to the syslog server will be using IPsec, you can configure the policy on the server to require IPsec, instead of requesting it. If you will have a mixture, with some systems using IPsec secured syslog and some systems connecting with plain UDP syslog (internal trusted systems, for example), then the most secure option would be to require security from the individual systems that will be using IPsec, based on their IP address or IP segment.

Notes from the Underground...

Log Servers Are Hacker Targets Too

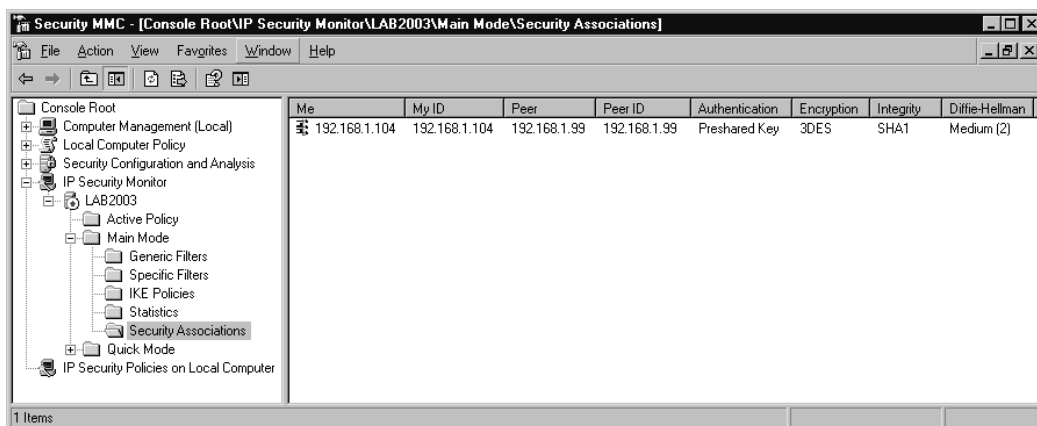
Remember that a logging server can make a very attractive target for a hacker for many reasons. For one, if he gains enough access to see that logs are being sent, attacking the logging server may enable him to delete the logs and erase his tracks. In addition to erasing his tracks, the log server will likely contain a treasure trove of information about your internal systems and what software is running on them. Based on the log data, even more information may sometimes be deduced. As an example, if the hacker sees a large number of connection from a variety of systems to a host in a DMZ on port 443, and then subsequent connection from that host to another host in the internal network on 1433, the attacker might guess that the first host is an e-commerce Web server that makes calls to an internal SQL database, possibly containing customer information such as credit card numbers.

If all that isn't enough, and the hacker cannot compromise the logging server, he could simply attack it to cause a denial of service so that it cannot receive the logs, or try to overwhelm the logging server with so many messages as to cause it to run out of disk space, crash, or at the least make it really hard to sort through all the chaff to find entries related to his attack. For all these reasons, the logging server must be secure like any other high-risk system.

30. In the MMC of the syslog client (not the server), select **IP security policies on local computer** in the left pane.
31. Right-click and select **All Tasks | Export Policies**.
32. Choose a name and location to save the policies and click **Save**.
33. Open the MMC of the syslog server, and if it isn't already added, add the **IP Security Policies** snap-in.
34. Right-click **IP Security Policies on local computer** in the left pane, and then select **All Tasks | Import Policies....**
35. Browse to the policies you exported previously and click **Open**. This will import the exact policy that was configured on the syslog client.
36. You should see the syslog policy in the list of policies in the right-hand pane. At this time it should list **No** under the **Policy Assigned** column.
37. Right-click the syslog policy and select **assign**.

Now when you send a syslog message it will negotiate an IPsec tunnel. You can verify that the tunnel was established by opening the **IP Security Monitor** snap-in in your MMC. Simply select **IP Security Monitor** on the left pane and click to expand the tree under your server name. Then select **Main Mode**, and finally select **Security Associations**. Your SA listing should look similar to the one shown below in Figure 5.17.

Figure 5.17 Security Associations



If you refer back to Figure 5.9, that was the sniffer output of the test syslog message demonstrating that without encryption, it was possible for anyone to read the contents of the syslog message. In Figure 5.18 you will find a similar capture, with our encryption policy in place.

Figure 5.18 Encrypted Syslog Message



```
Sniffing...
IP HEADER 192.168.1.104 -> 192.168.1.99
-----
IP->version: 4
IP->ihl: 5
IP->tos: 0
IP->tot_len: 136
IP->id: 12545
IP->frag_off: 0
IP->ttl: 128
IP->protocol: 50
IP->checksum: 63412

UNKOWN IP_PROTO 50
----- Begin of data dump -----
d6 46 14 5b 00 00 08 62 5e 34 00 a1 6e 4f ad .F.[...b^4..nO.
13 3d 58 88 9b 8b 41 ef d4 a8 f9 68 b8 d1 6a a8 .=X...A....h..j.
a4 38 72 74 ec bc 6d 56 f8 f4 7e 5a a8 32 92 82 .8rt..mV...~Z.2..
b3 39 10 f1 f6 01 97 4e 3f 82 36 39 8c 3d a8 fe .9.....N?.69.=.?
be f0 7d 60 b3 cd c1 42 8a 85 bf f1 72 fe c0 1c ..}^...B....r?..
ae 6f 5e 6d 4b 95 96 e0 ce 23 89 af 85 72 07 f9 .o^mK....#...r..
04 d1 70 8d 1a 30 00 fa 6e 64 f5 c7 c5 f1 46 05 ..p..0..nd....F.
cd aa 31 ac ..1.
----- End of data dump -----
```

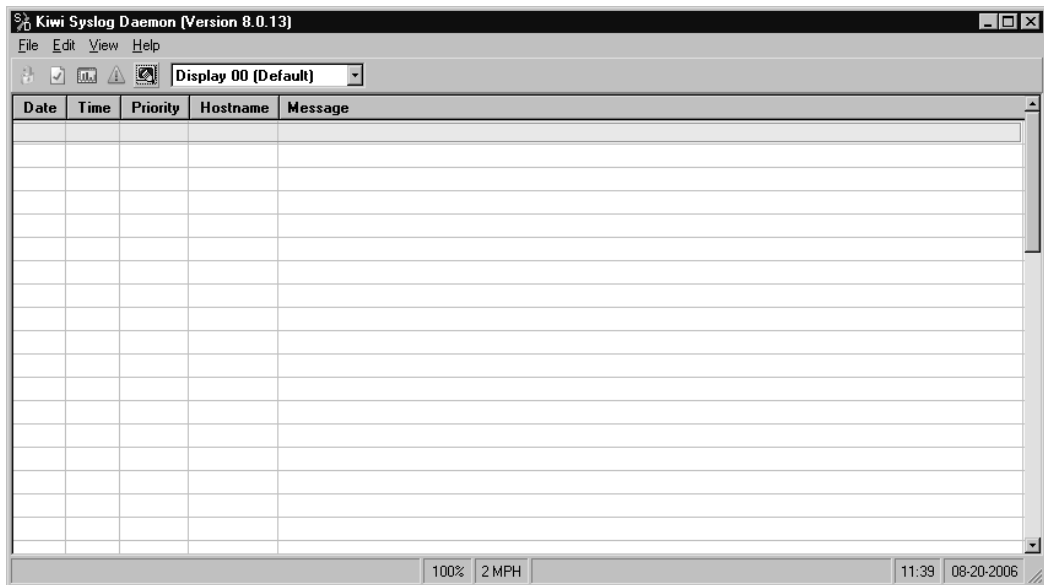
As you can see, the data is now unintelligible to hackers who might be trying to extract any useful information from your syslog messages.

Receiving Syslog Events

When it comes to running a syslog server on a Windows host (to receive syslog logs), there are several options. Far and away the most widely used free solution is KiwiSyslog available from www.kiwisyslog.com. To prepare your Windows host to receive syslog messages follow these steps.

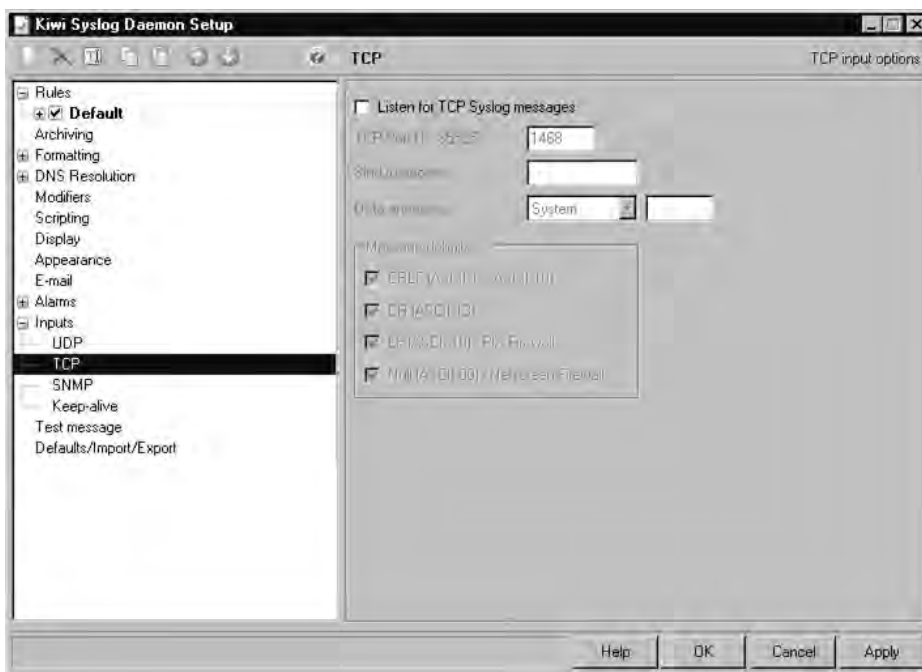
1. Download and install Kiwi Syslog Daemon.
2. When run, it will open up with the default display as shown in Figure 5.19.

Figure 5.19 Kiwi Syslog Console



3. For standard UDP syslog messages, you shouldn't need to do anything else to begin receiving messages. There are some configuration options you can configure by navigating to **File | Setup**. The setup window is shown in Figure 5.20.

Figure 5.20 Kiwi Syslog Setup



4. If you will be receiving any syslog messages over TCP, browse in the left pane to **Inputs | TCP**.
5. Check the box next to **Listen for TCP Syslog messages**.
6. Enter the port you will be receiving the messages on.
7. The other options are probably fine to leave at the defaults. If the messages are not formatted properly, or the priorities are not correct, adjust the message delimiters or Data Encoding respectively.

Kiwi Syslog also comes with some free alarm options. You can set an alarm for a minimum message count, maximum message count, or for a disk space threshold. The value of an alarm for minimum message count may not be immediately obvious. If you set this to one per hour, this means if you don't get any syslog messages in an hour it will trigger an alarm. Unless your environment is very small, or you are sending only high-severity messages, this probably means there is a connectivity problem. The alarm can make an audible sound, execute a program, or send an e-mail.

Another very powerful feature is that you can configure Kiwi to receive SNMP traps. SNMP is Simple Network Management Protocol. Traps can be configured as a sort of alarm message, to be sent when various parameters surpass certain thresholds. When this option is enabled, you can configure which facility and level/severity to use for the SNMP traps. In most cases, if you point your SNMP traps to the server that is hosting your Kiwi syslog, and check the box next to Listen for SNMP Traps, the default settings on this screen should work fine.

Linux Syslog

The Linux daemon that supports syslog is `syslogd`, or on some newer distributions, `syslog-ng`. With current Linux distributions you would be hard pressed to find one that didn't come with a syslog daemon already enabled, but if you need to, you can install it like any other package. There are a few differences between `syslogd` and the newer `syslog-ng`. `Syslog-ng` supports sending syslog messages over TCP, which gives you more choices for encrypting your syslog messages. The feature that will appeal to the most people is the increased granularity when configuring what actions should generate syslog events using the `/etc/syslog-ng.conf` file. This greater degree of control also comes with a price, which is that the configuration file of `syslog-ng` is a little more complex than that of plain old `syslog`.

Generating Syslog Events

Once installed, you can control the behavior of the syslog daemon through `/etc/syslog.conf` or, in the case of `syslog-ng`, through `/etc/syslog-ng.conf`. This configuration file is relatively simple and additional information can be found by viewing the man page for your version of `syslog`. You can typically type **man syslog** at the console to view the MANual for `syslog`. There should also be a man page for `syslog.conf` to explain the syntax of the configuration file. The configuration file is used to tell the syslog daemon what syslog messages to generate. Each line contains a *selector* and an *action* field. The selector tells the syslog daemon what types of events to act on, while the action field tells the syslog daemon what to do with those events. More practically you will see the following format used in the `syslog.conf` file: `<facility>.<severity><action>`. Take the following example:

```
Mail.*          /var/log/maillog
^Selector      ^action
```

This would tell the syslog daemon to log all events from the mail subsystem; that is, the mail facility, regardless of their severity, to `/var/log/maillog`. For examples of valid severities refer to Table 5.1 earlier in this chapter. An asterisk “*” is a special case and can represent several different things. If used in place of the facility or severity, it can represent all facilities *or* all severities respectively. If you place an asterisk as the action, the message will be sent to all logged-on users. The action can also be `/dev/console`, for example, to send it to the console.

Unless a given facility isn’t used or available on your bastion host, you should probably plan a logging action for all of them. The action will typically be to log to a real file, but there are some other options available. The most notable action from a security perspective is the *remote machine* action. This enables you to send the logs to a remote machine, which is a good idea. The syntax for the remote machine action is simply `@host`. The following line in the `syslog.conf` would send all critical events to the system at 192.168.1.11.

```
*.crit          @192.168.1.11
```

If the system holding your log files were compromised by a hacker, any logs on that host become suspect and of less value if legal action were to be taken. If you needed to use logs that were stored on the compromised machine, ensuring their integrity would be nearly impossible. You would have virtually no way to guarantee that the logs had not been modified; in short, the logs would become all but useless. By sending the syslog events to a syslog daemon on another machine, you can make the logs much more secure. The hacker would have to break into two machines in an attempt to modify the log files.

Encrypting Syslog Traffic

To encrypt your syslog messages, you will need to send your syslog messages over TCP and then use another means to encrypt the logs in transit. If the Linux distribution you are using does not include `syslog-ng`, it is available for free from www.balabit.com/products/syslog_ng/ and can be installed like any other package. There are many ways available to encrypt the syslog messages.

Stunnel can be used to send the logs over an SSL-encrypted tunnel. OpenSSH can use its port-forwarding feature in much the same way and encrypt the data using SSH. Finally, IPsec can be used as well. IPsec is the only option that enables you to encrypt the syslog logs even if they are being sent over UDP.

Before beginning the encryption configuration, verify that you can successfully send syslog messages from the Linux host to the syslog server (Linux or Windows). While UDP syslog uses a destination port of 514, TCP syslog messages do not have a standardized port number to use. You can use TCP 514 if you like, but it is officially registered to rshell. For more information, go to www.iana.org/assignments/port-numbers. If you don't have any other software listening on your syslog server on TCP 514, though, you can certainly use that port. The default `syslog-ng.conf` is configured to behave very closely to the original syslog if you don't modify anything. To direct logs to a remote syslog server you will need to edit the `syslog-ng.conf` file.

The default configuration file has all the destinations defined in one area. The default is to name destinations with a name following the format `d_<name>`, but this isn't required. Syslog.conf uses the format of `<facility>.<severity> <action>`. Syslog-ng uses the format `log { source; filter; destination; };`. Each of these fields must be defined in the `syslog-ng.conf` file. To define our remote destination, add the following line just after the last destination that is defined in the `syslog-ng.conf` file.

```
destination d_lab { tcp ("192.168.1.99" port(5140)); };
```

This defines `d_lab` as a syslog server at 192.168.1.99, which is listening on TCP port 5140. In the filter section, you can define any filters you want to apply to the logging. The following filter would trigger on all log messages of severity info (6) through emerg (0), which is basically anything except debug. Using `f_` to precede any filter names is a standard, but it is not a requirement. The following line will define a filter named `f_no-debug`.

```
filter f_no-debug {level(info..emerg); };
```

The source is defined by default as `s_sys`, and we have added definitions for the filter and the destination. You can now use these definitions and configure syslog-ng to send syslog messages. Do this by adding the following line after the last log line.

```
log { source(s_sys); filter(f_no-debug); destination(d_lab); };
```

After you save the changes, you will need to restart the syslog-ng daemon. Considering that we are logging everything short of debug, you should see some messages as soon as you start the syslog-ng service. In the event that you don't see any messages, or if you just want to do some further testing, you can generate a test message on most Linux systems with the logger command using the format *logger -p facility.severity message*. An example is:

```
logger -p cron.emerg test message here
```

This will send the messages “test message here” as facility cron, with a severity of emergency. Assuming you are able to successfully send TCP-based syslog messages, the next step is to add a layer of encryption to secure the syslog messages.

Configuring Stunnel

Stunnel doesn't contain any cryptographic code itself. Instead, Stunnel uses external libraries to perform the encryption. In this case, OpenSSL is used to create an encrypted tunnel. SSL stands for *Secure Sockets Layer*, which is the same well-tested encryption that is commonly used to encrypt Web pages. To demonstrate the operation of Stunnel on both Windows and Linux, we will be using a Linux host to generate the logs (via syslog-ng) and a Windows host to receive them (via Kiwi Syslog Daemon). This should give you a feel for the operation of Stunnel on both operating systems. Follow these steps to configure Stunnel on the Linux host.

1. Add the following destination definition to your syslog-ng.conf file, which will define a new destination called *d_loghost*.

```
destination d_loghost {tcp("127.0.0.1" port(5140));};
```

2. Change the log line in syslog-ng.conf that points to a destination of **d_lab** to point to the new destination **d_loghost** and restart syslog-ng. This tells syslog-ng to send the syslog messages to the localhost using a destination TCP port of 5140, which is where we will tell Stunnel to listen.

```
log { source(s_sys); filter(f_no-debug); destination(d_loghost); };
```

3. If Stunnel does not come pre-installed on your Linux distribution, then download and install Stunnel from www.stunnel.org.
4. Create a Stunnel configuration file called `/etc/stunnel/stunnel.conf`. Add the following text to the file and save the file.

```
client = yes
[syslog-ng]
accept = 127.0.0.1:5140
connect = 192.168.1.99:6140
```

This sets Stunnel in client mode (the default is server mode), and tells Stunnel to listen on port 5140 and send the data back out to 192.168.1.99 on port 6140.

5. Open a terminal window and start Stunnel by typing **Stunnel**.

Now you need to configure the server side of the SSL tunnel. Follow these steps to configure Stunnel on the syslog server and Kiwi Syslog Daemon.

1. Download and install Stunnel from www.stunnel.org.
2. Navigate to the directory where you installed Stunnel and edit the **stunnel.conf** file. Add the following at the end of `stunnel.conf` and save the changes.

```
[syslog_lab]
accept = 6140
connect = 7140
```

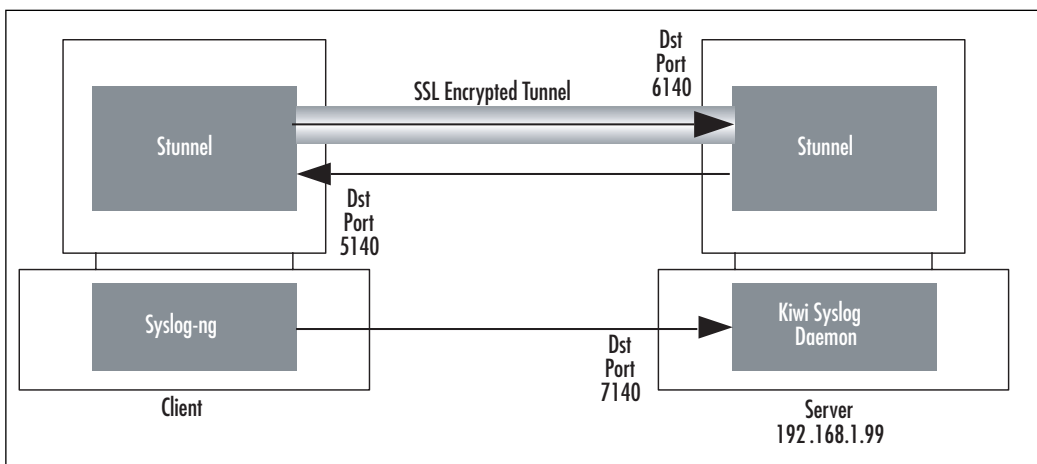
Stunnel will be in server mode by default, and the `accept-and-connect` line simply tells Stunnel to listen on port 6140, and when a connection is made, send the data out to 7140 on the local host. By not specifying an IP address in our configuration file, Stunnel assumes the local host.

3. Ensure that your Kiwi Syslog Daemon is listening for TCP connection on the port you specified in the previous step (7140 in this example). You can configure this by opening the Kiwi Syslog Daemon and navigating to **File | Setup**. Select **Inputs | TCP** in the left pane and ensure that **Listen for TCP Syslog Messages** is checked and 7140 is entered in the **TCP** port field.

4. Navigate to Start | Programs | Stunnel | Run Stunnel.

You should now be receiving log messages on the syslog server over the encrypted SSL tunnel. After you start Stunnel, it will place an icon in the system tray. Right-clicking on this icon will open a very small menu. If you select **Log** on this menu, you can see a running log of the connections that Stunnel has accepted. Because the communication path can be a little confusing, Figure 5.21 shows a graphical representation of the data flow using Stunnel.

Figure 5.21 Stunnel Data Flow



After you have verified that you can receive the encrypted syslog messages, you will need to generate a new server certificate and private key to replace the default one. Stunnel requires a server certificate and private encryption key for the SSL encryption to function. The installation files come with a default certificate and key, combined into a single file called `stunnel.pem`. Because the *same* certificate is distributed with all the installation files, using this certificate in a production environment would be insecure because everyone would have your key. You need to generate a new certificate

and key file for use on the server (you do not need to do this on the client host). The simplest way of generating the new certificate and key is by using the OpenSSL package.

1. Download and install OpenSSL. The source files (to be used on Linux) can be downloaded from www.openssl.org/source/. Pre-compiled binary files (for use on Windows) can be downloaded from www.slproweb.com/products/Win32OpenSSL.html.
2. After you have downloaded and installed OpenSSL, use these commands to generate a new server certificate: **openssl genrsa -des3 -out server.key 1024**. You will be prompted for a pass phrase and it will then generate an initial server key.
3. Enter **openssl req -new -key server.key -out server.csr** to generate a certificate signing request. You will be required to enter the previously assigned pass phrase and answer several prompts with regional information such as your state, country, and company name.
4. Enter **openssl rsa -in server.key.org -out server.key** to remove the pass phrase from the server key. You will be required to enter the pass phrase to complete this step.
5. Enter **openssl x509 -req -in server.csr -signkey server.key -out server.crt** to generate the server certificate.
6. Finally, combine the certificate and key file into a single .pem file that Stunnel will use by entering the following command: **copy server.crt+server.key stunnel.pem**
7. Replace the original stunnel.pem with the newly created stunnel.pem certificate in the Stunnel directory.
8. Restart Stunnel on the server and verify that everything is still working properly.

Now that you are successfully encrypting your syslog messages using Stunnel, you might notice that all of the syslog messages claim to be sourced from 127.0.0.1, which is the local host. This is because technically, that is who sent the message to the syslog server. If you are receiving the syslog messages to syslog-ng (on a Linux host) you can change this behavior by including

keep_hostname(yes); in the options section of the `syslog-ng.conf` file. Stunnel enables you to encrypt virtually any TCP-based connection, not just syslog. You could use Stunnel to encrypt a Telnet session, for example, or even a custom TCP-based application. After you have the Stunnel software installed and the basic configuration set up, adding additional tunnels is as simple as defining additional services in the `stunnel.conf` configuration file.

Configuring OpenSSH

OpenSSH is an open source implementation of the SSH (Secure Shell) protocol. Originally it was intended as a secure alternative to other clear text protocols for remote administration, such as Telnet or rshell, SSH includes a port forwarding option that enables it to function similarly to Stunnel. There are advantages to using OpenSSH over Stunnel, such as the fact that you might already have OpenSSH installed to provide remote access. In those cases it would be one less piece of software that you needed to install and configure. Virtually every Linux implementation will come with OpenSSH installed by default, so you are more likely to already have SSH on your Linux host than you are Stunnel.

We discussed the installation and configuration of OpenSSH in Chapter 2, “Protecting Your Perimeter,” so we will discuss only the configuration of OpenSSH here, under the assumption that you already have OpenSSH installed and working properly to provide remote command-line access. OpenSSH port forwarding, in this scenario, is used like Stunnel. The SSH client establishes a connection to the SSH server and then listens to local traffic destined for a port you specify. When it receives it, it will send the data out encrypted, to an OpenSSH server, which in turn decrypts the data and passes it to the local port the service is listening on. To setup port forwarding using OpenSSH, perform the following steps.

1. On the Linux host, run the following command: `ssh user@192.168.1.99 -L 5140:192.168.1.99:7140`.
2. Generate a sample syslog message and it should be received on the syslog server. If not, ensure that your Kiwi Syslog Daemon is still listening on TCP port 7140.

Although much of the operation of SSH port forwarding is the same as the SSL tunnel that Stunnel creates, there are a couple of differences. By default, the SSH port forwarding is just that. It is the forwarding of a port on an established SSH tunnel. So the encrypted SSH tunnel is established first, using the normal SSH TCP port of 22 unless you specify a different port. This tunnel is set up exactly the same as if you were going to connect for remote command-line access. In addition, SSH will listen for and forward connections on the additional port you have specified. The primary disadvantage of SSH for port forwarding is there is no command line means to specify the password. This is an intentional design choice on the part of the SSH developers in order to increase security. They specifically did not wish to provide a simple means of including passwords in scripts and batch files. This requirement for user interaction makes SSH a better candidate for interactive sessions rather than service-based connections such as syslog. As you can see, however, it is very easy to set up on a system that already has SSH configured properly.

Configuring IPsec

Because IPsec is an industry standard, specifically designed for interoperability between different vendor systems, it is a very popular choice for implementing encryption. Current Linux kernels have IPsec support built in, and there are other packages that provide their own implementation of IPsec as well. For these examples we will assume you are using the Linux kernel native IPsec support. Follow these steps to configure IPsec on a Linux host.

1. If they are not already installed, download and install the IPsec Tools from <http://sourceforge.net/projects/ipsec-tools>. These tools provide a simplified interface to configure the various IPsec settings.
2. Edit the `/etc/racoon/psk.txt` file. This file holds the pre-shared (AKA “secret”) keys. In the previous example we chose “password” as our pre-shared key. (This certainly isn’t a secure password; you should use a high-quality password for a production environment.) The format of the file is `<identifier> <key>`. In our example we would add the following line to the file and save the new file.

```
192.168.1.99 password
```

The previous example for setting up IPsec on Windows used a preshared key of “password.” If you wish to change the preshared key on the Windows syslog server, edit the IPsec policy by following these steps:

3. Open the MMC and select **IP Security Policies on Local Computer** in the left pane.
4. Double-click the **Inbound_Syslog** policy in the right pane.
5. Double-click the **TCP Syslog** security rule and select the **Authentication Methods** tab.
6. Click **Edit** to change the preshared key, and enter the new key.
7. Click **OK, OK, Apply**, and **OK** to accept the changes and exit the policy configuration windows.

The next step is to configure the IPsec policy on the Linux host. With the IPsec tools loaded, this can be done using the setkey utility. The utility can display the current security associations and perform several other configuration changes to your IPsec policy. By creating a setkey configuration file, we will define the security parameters to use. The entire contents of the configuration file are shown in Figure 5.22.

8. Create a configuration file, you could name it `/etc/racoon/setkey.conf`, and enter the information shown in Figure 5.22.

Figure 5.22 setkey Configuration File

```
# Configuration for 192.168.1.105
# Flush the SAD and SPD
flush;
spdflush;

#### ESP SAs using 192 bit long keys (168 + 24 parity) ####
add 192.168.1.105 192.168.1.99 esp 1001
    -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;

add 192.168.1.99 192.168.1.105 esp 1001
    -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
```

```
#### Security policies ####
spdadd 192.168.1.105 192.168.1.99 any -P out ipsec
        esp/transport//require;

spdadd 192.168.1.99 192.168.1.105 any -P in ipsec
        esp/transport//require;
```

All lines beginning with `#` are comments and ignored by `setkey`. The `flush` and `spdflush` tells `setkey` to wipe out the previous configuration. This enables us to start clean and ensures that we are using only what is contained in this configuration file. The next section (`#ESP SAs`) defines the *Encapsulating Security Payload* (ESP) parameters. The first line states that traffic from 192.168.1.105 (the syslog client) to 192.168.1.99 (the syslog server) should use ESP and 3DES for encryption. The long string beginning with `0x` is a key. This is just a sample key used for testing. You should generate your own key for increased security. The following section serves the same purpose for traffic coming from 192.168.1.99 to 192.168.1.105. In this example we used the same key for both, but you certainly don't have to. The final section (`# Security Policies`) defines the IPsec modes to be used. We are configuring transport mode and requiring that traffic matching the policy be encrypted. The line is duplicated with the IP addresses reversed so that our policy will apply to traffic in both directions.

9. Apply the settings in your IPsec policy by entering **`setkey -f /etc/racoon/setkey.conf`**.
10. Edit your racoon configuration file **`/etc/racoon/racoon.conf`**.

Racoon is the daemon on Linux that handles your *Internet Key Exchange* (IKE) functionality. If invoked from the command line with no options, it will automatically be run in daemon mode. For initial testing and setup, we would recommend running it in the foreground, so that you can see the output for troubleshooting purposes. Executing `racoon -F` will run in the foreground, and adding `-d` (for debug) will increase the verbosity level to provide even more information. Figure 5.23 shows the complete `racoon.conf` contents.

Figure 5.23 Racoon.conf File

```

SYNGRESS
syngress.com
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

##### IKE PHASE 1
remote 192.168.1.99 {
    exchange_mode main;
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group 2;
    }
}

##### IKE PHASE 2
sainfo anonymous
{
    lifetime time 1 hour ;
    encryption_algorithm 3des, des ;
    authentication_algorithm hmac_sha1, hmac_md5 ;
    compression_algorithm deflate ;
}

```

The first few path lines are unedited from the defaults and tell racoon where to find the pre-shared key file and any certificates you want to use. The functioning of this configuration file is pretty straightforward. The *remote* entry says that when speaking to 192.168.1.99 we will use main mode and attempt to use 3DES, with SHA1, and a pre-shared key. The second section contains security association information that should be applied to all hosts (due to the anonymous entry). You could instead define different parameters to be used when communicating with different hosts if desired, by

creating multiple entries in the format `sainfo <host>` instead of using anonymous.

11. At a terminal prompt, start `racoona` using `racoona -F` or `racoona -F -d`.

If you already have the Windows IPsec policy defined from the previous example, follow these steps to *change the previously configured IPsec policy* to accept TCP-based connections. If you are not using the same system, refer to the Windows IPsec section of this chapter and follow the steps for configuring the IPsec policy, substituting the TCP port number you want to use for the UDP port 514 that was used in the previous example.

1. Open the MMC and navigate to the **IP Security Policies on Local Computer** in the left pane.
2. Double-click the policy that was created previously; in my example it was called **Inbound_Syslog**.
3. Highlight the syslog filter, which will open the **Edit Rule Properties** window.
4. Click Add and enter a name for the new filter, such as **TCP Syslog**.
5. Enter a description and click **Add**, ensure that the **Use Add Wizard** is checked.
6. Click **Next** in the **IP Filter Wizard**. For source address select **Any** and click **Next**.
7. For Destination Address select **My IP Address** and click **Next**.
8. For a protocol select **TCP** and click **Next**.
9. Leave **From any port** selected, and change the bottom option to **To this port**. Enter the port you want your syslog server to listen on, such as 7140.
10. Click **Next**, **Finish**, and then **OK**.
11. This will place you back at the **Edit Rule Properties** window; ensure that the new rule is selected with the radio button and click **OK**.
12. Back on the Properties window, click **Apply** and then **OK**.

The new policy should now be in effect, which will encrypt traffic it receives on TCP port 7140. You should now be able to receive your encrypted syslog messages over TCP to port 7140. Because IPsec is not protocol dependent, this same type of configuration can easily enable you to encrypt UDP-based syslog from your Linux system to a Windows system as well. To do this, simply substitute the configuration options of TCP 7140 for a UDP port of your choice, such as the default UDP 514.

**TIP**

When you first apply all the IPsec settings, you will probably not see traffic immediately. In most cases there will be a short delay while the initial IPsec connection is being established. This time is being spent agreeing on the encryption parameters and exchanging key information prior to the secure communications being able to take place.

The following is a short summary of the various encryption options for use with syslog.

- **SSL** SSL is probably the simplest to implement. It does require a TCP-based syslog daemon and reconfiguration of syslog to point to the local listening SSL port. Stunnel may or may not need to be installed on your particular distribution.
- **SSH** In almost all Linux systems SSH will be included in the default install. Like SSL, it does require a TCP-based syslog daemon but does not require reconfiguration of the syslog ports and destinations, other than to use TCP. The biggest disadvantage is that SSH is intended for interactive session and requires authentication to establish the SSH tunnel.
- **IPsec** IPsec is both the most functional and flexible encryption option, as well as the most complicated. You need to match various security association settings on both systems and multiple files have to be configured in order for it to work. IPsec's primary strengths are the high degree of flexibility in how it is configured and that it is

protocol independent. You can implement IPsec without making any changes to your TCP-based syslog configuration.

Bear in mind that this is a very minimalist IPsec configuration. The objective is only to secure your syslog traffic. Configuration can become much more complex, particularly if you need to configure different IPsec policies for multiple systems. Refer to www.ipsec-howto.org for some very good documents that walk you through the process in a little more detail. We would also recommend reading the man page for `syslog-ng`, `syslog-ng.conf`, `setkey`, `racoon`, and `racoon.conf`.

Receiving Syslog Events on a Linux Host

Configuring your Linux host to accept syslog messages is fairly simple. For the generation of test events from a Windows host, Kiwi offers a handy tool creatively called Kiwi Syslog Message Generator. It will generate both TCP and UDP messages of your specification, or messages using random parameters. You can download the message generator at

www.kiwisyslog.com/info_sysloggen.htm. If you are using the standard syslog daemon, you will need to edit the `/etc/sysconfig/syslog` file. The line that contains `SYSLOGD_OPTIONS=` tells the system how to run the syslog daemon. Add `-r` to the list of options and restart the syslog service. That's all there is to it. Your Linux system will now be listening on the default syslog port (UDP 514) for syslog messages. If you enter `netstat -A` in a terminal window, you should see syslog as one of the listening processes.

If you are using `syslog-ng`, you will need to edit the `/etc/syslog-ng/syslog-ng.conf` file. If you are only trying to receive syslog messages on the default UDP port of 514, all you need to do is *remove* the comment delimiter (`#`) from the following line in the `syslog-ng.conf` file.

```
# udp(ip(0.0.0.0) port(514));
```

The default configuration file is set up this way to mimic the original syslog daemon, which is set up to *not* listen for syslog messages from the network by default. After editing the configuration file, restart the `syslog-ng` service and you should begin receiving syslog messages on UDP port 514. To configure the Linux host to receive syslog messages on a TCP port (recom-

mended) instead, add the following to the `syslog-ng.conf` file in the source definition section (`s_sys` is the default source definition).

```
tcp(ip(0.0.0.0) port(7140));
```

Replace 7140 with the TCP port you wish to receive syslog messages on. If you wish to configure some type of encryption for your syslog messages to the Linux host, you can use any of the previously discussed examples to accomplish this.

Analyzing Syslog Logs on Windows and Linux

Now that you are receiving all these wonderful logs, there is the question of what to do with them. You are probably realizing that they can create a large volume of data in a relatively short time frame. You need a way to sort through the log files and pick out the most interesting pieces of information, and then do something about it where necessary, preferably all in as automated fashion as possible. There are basically two types of processing you may wish to perform on your log data. One is a *real time* analysis, and the other is an *offline* analysis done at some later point in time after the logs have been generated. Typically, the real-time type of analysis is used for sorting out the more important events from the bulk of “normal” activity and highlighting them or sending alerts to bring these events to the attention of a human for further analysis. This type of analysis and response can take many forms, from a simple color coding as the events scroll across an administrative terminal, to complex rules for sending out automated e-mails or paging people.

The offline analysis is more often used for less time-sensitive investigations, such as identifying trends or spotting issues for follow up. For example, a CPU running a little high is not generally a critical event that would be flagged during a real-time analysis, while a *chronically* high CPU might be something you want the offline analysis to report on. On a similar note, an *excessively high* CPU would be appropriate for real-time notification because it could indicate a production device is no longer able to adequately service requests. Each type of analysis has its merit and uses. What follows is a discussion of some useful tools for log analysis.

Windows Log Analysis

Your first criteria for log analysis is to determine if you are going to be analyzing the logs in their native Evenlog format, or in syslog format on a Windows host. Kiwi offers the Kiwi Logfile Viewer, which will monitor a logfile for changes and color code the entries as they appear in the Logfile Viewer. Kiwi Logfile Viewer can be downloaded from www.kiwisyslog.com/log-viewer-info.php. The install routine holds no surprises, so install the viewer to a directory of your choice and then execute the Log Viewer. When the viewer first opens, it will be blank. You will need to navigate to **File | Open** and browse to the log file. If you are using the Kiwi Syslog Daemon, the default log file is in the directory where you installed Kiwi Syslog Daemon such as `syslogd/logs/SyslogCatchAll.txt`. The main window will then populate with the log entries. These are the same entries that would appear in the Kiwi Syslog Daemon window. In order for the viewer to detect changes to the log file, select **Edit | Tail** from the viewer menu, or click the tail button across the top. You now have the messages displaying in the viewer and it is updating as new messages are received. The next step is to configure some color coding to make the important events stand out.

Navigate to **Options | Highlighting** to access the configuration window. Click the large plus on the right side to create a new highlight trigger. The default it creates will be a red highlight. Edit the **String to Match** field and enter the text you wish to trigger the highlight. An example would be `.emerg` to trigger on all emergency priority messages. You can of course edit the text and background colors that are used for the highlighted message. The Kiwi Log Viewer window is shown in Figure 5.24.

The Kiwi Log Viewer is intended to be running on a console where a human analysis can monitor it throughout the day. It lacks any advanced features or configuration but is extremely easy to install and configure.

Figure 5.24 Kiwi Log Viewer

The screenshot shows the Kiwi Log Viewer application window. The title bar reads "Kiwi Log Viewer - [I:\Internet\Syslog\Logs\SyslogCatchAll.txt]". The menu bar includes "File", "Edit", "View", "Options", and "Help". Below the menu bar is a toolbar with icons for file operations and a status bar showing "Size: 252 KB" and "Format: Kiwi MSWinEventLog YYYY-MM-DD". The main area contains a table with the following columns: "Date Time", "Facility.Level", "Hostname", and "Event Log Type".

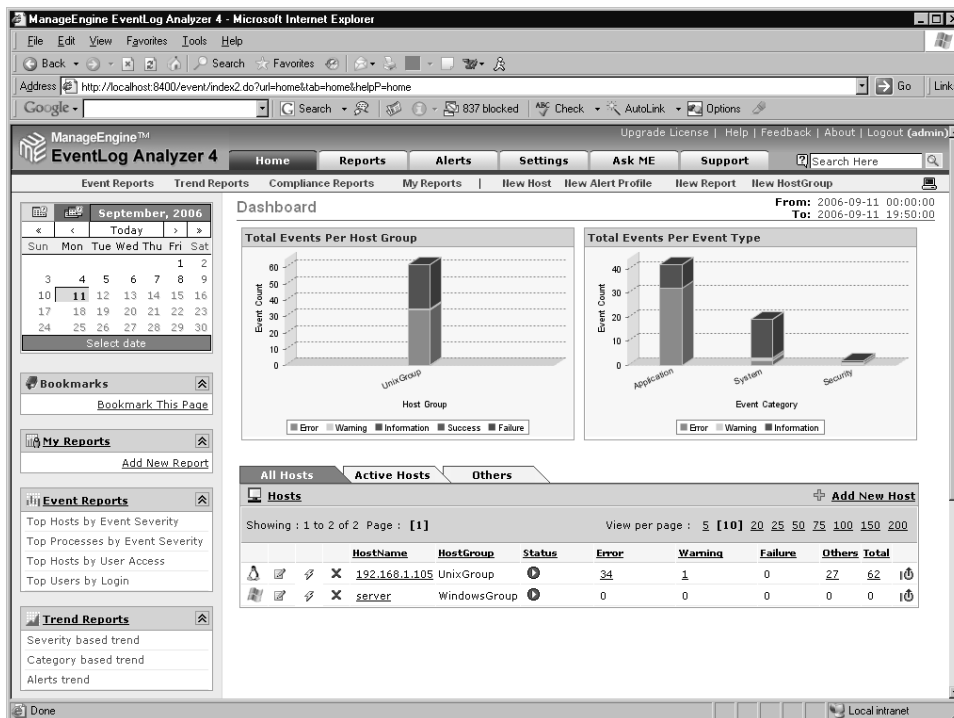
Date Time	Facility.Level	Hostname	Event Log Type
2006-09-11 17:54:28	User.Info	192.168.1.105	Sep 11 17:53:02 localhost (root-2344): Resolved address "xml:readonly:/etc/gconf/g
2006-09-11 17:54:28	User.Info	192.168.1.105	Sep 11 17:53:02 localhost (root-2344): Resolved address "xml:readwrite:/root/.gconf
2006-09-11 17:54:28	User.Info	192.168.1.105	Sep 11 17:53:02 localhost (root-2344): Resolved address "xml:readonly:/etc/gconf/g
2006-09-11 17:54:31	User.Info	192.168.1.105	Sep 11 17:53:06 localhost (root-2344): Resolved address "xml:readwrite:/root/.gconf
2006-09-11 17:55:29	Cron.Emerg	192.168.1.105	Sep 11 17:54:03 localhost root: 12345678901234567890
2006-09-11 18:01:23	Syslog.Notice	192.168.1.105	Sep 11 17:59:57 localhost syslog-ng[1555]: STATS: dropped 0
2006-09-11 18:01:26	Daemon.Critical	192.168.1.105	Sep 11 18:00:00 localhost smartd[1795]: Device: /dev/hda, 2 Currently unreadable (
2006-09-11 18:02:26	Cron.Info	192.168.1.105	Sep 11 18:01:01 localhost crond[2470]: (root) CMD (run-parts /etc/cron.hourly)
2006-09-11 18:08:41	Syslog.Notice	192.168.1.105	Sep 11 18:07:15 localhost syslog-ng[1555]: AF_INET client connected from 192.168.
2006-09-11 18:08:41	UUCP.Error	192.168.1.105	Sep 11 18:07:15 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:41	Cron.Notice	192.168.1.105	Sep 11 18:07:15 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:42	Auth.Error	192.168.1.105	Sep 11 18:07:16 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:42	System3.Info	192.168.1.105	Sep 11 18:07:16 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:42	System2.Error	192.168.1.105	Sep 11 18:07:16 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:42	System2.Alert	192.168.1.105	Sep 11 18:07:17 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:43	System4.Error	192.168.1.105	Sep 11 18:07:17 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:43	UUCP.Critical	192.168.1.105	Sep 11 18:07:18 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:44	News.Alert	192.168.1.105	Sep 11 18:07:18 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:08:44	System2.Alert	192.168.1.105	Sep 11 18:07:18 192.168.1.99 This is a test message generated by Kiwi SyslogGen
2006-09-11 18:11:23	Syslog.Notice	192.168.1.105	Sep 11 18:09:57 localhost syslog-ng[1555]: STATS: dropped 0

EventLog Analyzer 4 is a Java-based application that will analyze both Windows event logs and syslog logs from a Windows host. It can be installed on a Linux host as well, but then it cannot collect event logs from a Windows host. The free version does limit your analysis to only five total hosts. It can be downloaded from <http://manageengine.adventnet.com/products/eventlog/download.html?free>. When you perform the install, you will be prompted to choose from the trial edition or the free edition. In this example we are choosing the free version. After selecting the install directory, you will be prompted to choose a Web server port, which will later be used for viewing reporting data. You will also be able to choose if you want to install EventLog Analyzer as a service (which is the default).

After it is installed, start the program via the start menu. This will open a command prompt and execute the Java application. Once completed, it should open up your default browser to the login page. The default username and password are on the login page. They are username = admin with a password of admin. EventLog Analyzer expects to act as the syslog server process by default. This means you wouldn't need another program to listen for incoming syslog messages. To add hosts, click **Add New Host**. If you

want to receive syslog messages from a Linux host, select **UNIX** for the host operating system and enter the host name or IP address of the system you want to receive logs from. For a UNIX system (syslog), the logging should be more or less constant, but for Windows hosts (event logs) you must configure the logging interval to no shorter than every 10 minutes. The Home tab will display a list of hosts and some basic event summary information as shown in Figure 5.25.

Figure 5.25 EventLog Analyzer 4



EventLog Analyzer does offer some nice features, such as some pre-configured reports sorted by event severity, event process, and some trend reporting options. To configure an e-mail alert to be sent, there are two steps to complete. First, you need to click on the **Settings** tab, and then select **Mail Server Settings**. Enter your POP server and if required, any login information needed to authenticate to the POP server; then click **Save**.

Next, click the **Alerts** tab and then click **Add Alert Profile** on the left side of the window. The next screen shown allows you to configure the trigger for the e-mail alert. You can name the alert profile and choose whether it applies to a single host or a host group. Next, select the facility and severity you wish to trigger the alert. You can also specify a string that the log message must contain for additional granularity. Choose a criticality for the alert e-mail and enter an e-mail address. When completed, click **Add Alert Profile** to save the profile. Figure 5.26 shows the newly created profile on the **Alerts** tab.

Figure 5.26 Alert Profile “Test”

Alert Profile Details										
New Alert Profile										
	Name	Host/Group	LogType	Severity/EventID	Criticality	Occurrences	Interval (min)	Notifications	Delete	A
	test	192.168.1.105	Kernel	Critical	High	1	1	0	X	T

As you can see, we configured this alert to trigger on a *kernel.critical* message. On the Linux host (192.168.1.105), you can generate a test message to trigger the alert using `logger -p kern.crit testmessagehere`. All in all, the EventLog Analyzer has some very nice reporting capabilities. The five-host limit may or may not pose any issues in your environment. On the downside, it seems to be rather resource intensive and it only collects data on a polled interval, so it is not very well suited for time-sensitive alerting or notification.

Kiwi Syslog Daemon is quite simply an amazing piece of free software. We have been using it in several of the examples as a syslog server on a Windows host, but it can do a lot more than just receive the syslog messages. To start with some basic filtering capability, notice the top of the window contains a drop-down box that defaults to Display 00 (Default). By default, all inbound logs will go to display 00; if you choose any of the other displays, they will all be empty. You can follow these steps to configure a custom filter, which will send the events that match the filter to a different display. In this way you could send all the emergency messages to a different display.

1. Navigate to **File | Setup** and right-click **Rules** in the left-hand pane.

2. Choose **Add Rule**, and then enter a name for the new rule. In this case we entered `Crit_Events`.
3. Under your new rule, right-click **Filters**, click **Add Filter**, and then select a name for this filter.
4. With the filter selected in the left pane, the right side now has new drop-down windows at the top. The **Field:** box enables you to choose which field to use for the filter. If you select **Priority**, it will enable you to filter the messages based on both the message facility *and* the message severity. The resultant grid enables you to highlight all combinations of facility and severity you wish to filter on. By clicking the **Crit** column header, and then clicking the check mark at the bottom, it will select all facilities with a severity of critical.
5. Once you are satisfied with your selection, click **Apply**. The Priority filter configuration window is shown in Figure 5.27.

Figure 5.27 Kiwi Syslog Daemon Priority Filter



6. Next, configure the action to take on messages that match the filter. Do this by right-clicking **Actions** and selecting **Add Action**.
7. In the **Action** drop-down box at the top, select **Display**.
8. Choose a display to direct the critical messages to other than the default display and click **Apply**.

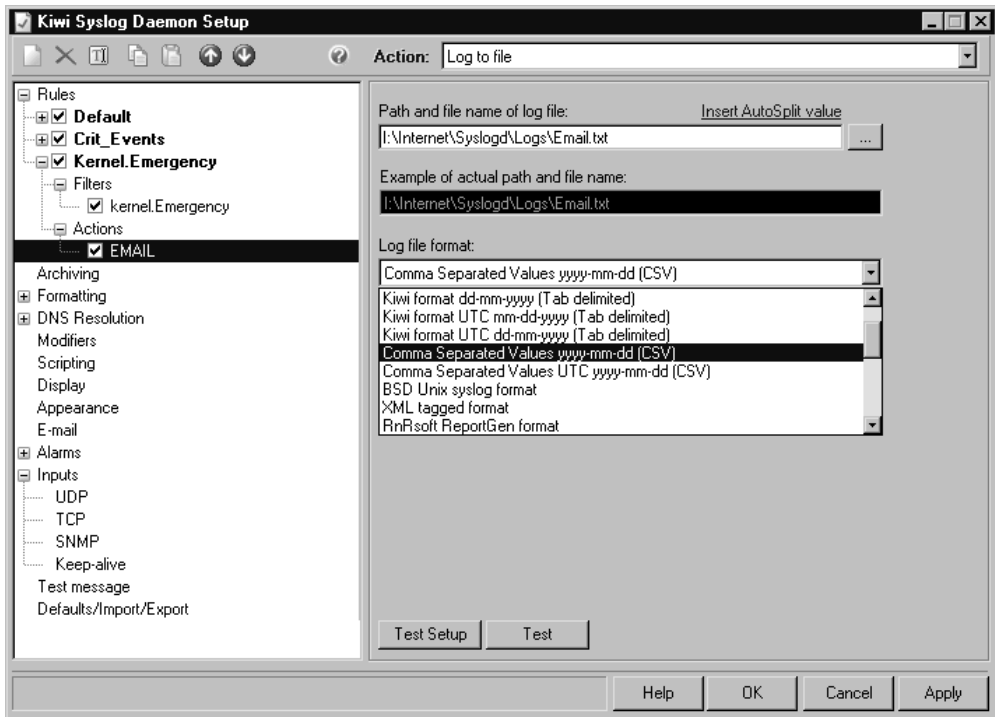
You can change the names of the displays to something more meaningful by navigating to **Display** in the left pane and then using the **Modify Display Names** drop-down box to select the display you want to modify. Enter the new name in the field to the right and click **Update**.

Unfortunately, there are several features that are only available with the registered version, such as sending an e-mail, running an external program, or logging the message to the Windows event log based on message content filter rules. Kiwi Syslog Daemon is still a very powerful tool in its free form. Even given the limitations of the free Kiwi Syslog Daemon you can take advantage of its filtering capabilities and some external utilities to achieve some alerting functionality anyway.

As a simple example, you could configure one of the Kiwi filters to write to a file as its action. In this example, any events matching *kernel.emergency* will be written to a file `\syslogd\logs\Email.txt`. Because the default filter writes all events to `\log\SyslogCatchAll.txt`, the event will be recorded in both files. You will probably want to change the logging format to a comma-separated value to make parsing the file easier. You can change this by navigating to File | Setup and drilling down in the left pane into Rules until you get to Actions. With your custom action highlighted you will see the Log File Format drop-down box shown in Figure 5.28. There are many formats to choose from; in this case, we chose Comma Separated Value `yyy-mm-dd (CSV)`.

After this is done, we will generate a separate *email.txt* file for all *kernel.emergency* messages that will be easy to work with in a batch file. The following batch file will demonstrate how to easily look for syslog messages and send them in an e-mail.

Figure 5.28 Kiwi Log File Format



```
@echo off
if exist email.txt goto :mail
exit

:mail
ren email.txt working.txt
FOR /F "tokens=*" %i in (working.txt) do (
echo %i > lastline.txt
)

FOR /F "tokens=1-6 delims=," %i in (lastline.txt) do (
blat working.txt -to recipient@any.com -f syslog@any.com -s "Syslog [%%j]
^from host %%k" -server smtp.any.com
)

del working.txt
del lastline.txt
```

As you can see, this type of powerful functionality doesn't take a lot of work to get working. The first few lines are just checking for *email.txt*. If Kiwi hasn't found any *kernel.emergency* messages, *email.txt* will not exist, in which case we just exit. If *email.txt* is present, we know we need to send an e-mail so we jump to *:mail*. Our first task is to rename *email.txt* to *working.txt* so that if more messages are generated while the batch file is running, it won't cause any problems. The *email.txt* file could contain multiple syslog messages, so we iteratively pipe each line of *email.txt* to a temporary file called *lastline.txt*. Because we are sending each line to the same file, when it's all finished, we will only be left with the last syslog message in *lastline.txt*. We then use *FOR* to grab the facility and severity from the *last* syslog message in *lastline.txt* and place that in the subject line of our e-mail. In the body of the e-mail we use *blat.exe* to send *working.txt*, which will have all the syslog messages generated at the time the batch file was run. As part of the final clean up, we delete *working.txt* and *lastline.txt*.

Blat.exe is a free command-line SMTP program available for download from www.blat.net. Note that the line is wrapped at the caret “^” due to space limitations in the book. Those two lines should all be one line in the batch file. As for *BLAT*, *-to* is used to specify the e-mail recipient and *-f* will be displayed as the e-mail sender. *-s* specifies the subject line and, finally, *-server* is used to specify the SMTP server to send the e-mail to for processing. There are, of course, many ways to accomplish the same thing. The batch file also uses only the filenames, meaning they all need to be in the same directory (the Kiwi log directory); otherwise, you should use full paths for all the file references. This batch file could be scheduled to run every minute or every five minutes to provide relatively fast e-mail alerts. The Internet contains a wealth of programs for many purposes that are free to use. To provide a few examples, here are some high-quality free command-line utilities.

- **Blat** Command-line SMTP— www.blat.net
- **Vmailer** Command-line SMTP—www.virdi-software.com/vmailer/
- **Bmail** Command-line SMTP—www.beyondlogic.org/solutions/cmdlinemail/cmdlinemail.htm

- **sendEmail** Command-line SMTP—
<http://caspian.dotconf.net/menu/Software/SendEmail/>
 sendEmail has the advantage that it is available for both Windows and Linux, so you only need to learn a single set of syntax across both operating systems.
- **Libwww** Suite of utilities for command-line HTML
 Read about it here www.w3.org/Library/Distribution.html.
 Download it here www.idm.ru/content/view/9/8/lang,en/.

Linux Log Analysis

When it comes to log analysis, there are many options available for Linux systems. What you are looking for in a log analysis utility will determine which one is the best one for you to use, and in most cases no single utility will be able to do everything you might want it to. You probably can, however, find a small subset of tools that can do 99% of what you need. Two robust general-purpose tools are *swatch* and *logwatch*. These are general purpose in that they are not specific to a particular log file format. Of the two, *swatch* is the more lightweight, being both easier to set up but also having fewer options. *Swatch* is intended to parse the logs in real time and to act upon what it finds inside the logs according to the configuration you specify. *Logwatch* has a slightly different role. Its focus is on analyzing and reporting on log files, but not in real time.

Configuring Swatch

Because *swatch* (short for simple watcher) is relatively focused in its purpose, the setup and configuration are pretty simple. The *swatch* home page can be found at <http://swatch.sourceforge.net/>. Some *swatch* behaviors can be set from the command line, but the rules to match must be in a configuration file. An example command line to invoke *swatch* is `swatch -c /etc/swatch.conf -t /var/log/syslog`, which tells *swatch* to use the configuration file (`-c`) at `/etc/swatch.conf`. If you don't specify which file to watch using the `-t` option, *swatch* will default to `/var/log/messages` or `/var/log/syslog`, in that order. If you don't specify a configuration file it will echo everything to the console. You can use `-f <file>` to examine a file once, instead of it running continuously with the `-t` option.

The real meat of what swatch can do is defined in the configuration file. If you had the following lines in your configuration file, it would cause any line containing “denied” or “Denied” to echo to the console as yellow text and sound the bell once. Everything else would be caught by `./ */` and get echoed to the console as normal text.



```
watchfor      / [dD] enied/
echo yellow
bell 1
```

```
watchfor /.* /
echo
```

As you can see, configuring swatch is not difficult. Refer to the swatch man page for specifics on some of the coloring options and other functionality of swatch. Some of the key commands for security considerations and their use follow:

- **—script-dir=<path to directory>** Used on the command line, when swatch runs, it creates a temporary watcher script, which by default is written to the user’s home directory. You should redirect watcher script to a secured directory where it cannot be edited. A hacker with access to this temporary script could control what swatch reports and cover his tracks.
- **exec command** Used in the config file, this will cause matches to execute another command. This could be as simple as a program to send an e-mail, for example, or it could run a custom script to lock down the netfilter firewall automatically.
- **Throttle hours:minutes:seconds,[use=message | regex | <regex>]** This is especially valuable because it controls how often duplicates of a given message will be acted on. This way a brute force password cracker being run won’t overload swatch with a non-stop scrolling message, possibly filling up your logging partition.
- **Threshold events:seconds,[repeat=no | yes]** This is another very important one. This enables you to ignore certain matches until they surpass a given threshold; for example, threshold 4:60 will not

perform any action unless the pattern is matched four times within a sixty-second window. This is very useful for things such as incorrect passwords. You don't want all sorts of alarms going off because the admin mistypes a password once, but many incorrect attempts in a short time frame may be a sign of a hacker trying to compromise the account.

NOTE

Although the man page includes `threshold`, this option doesn't seem to work as written. Various help forums have stated that it works on Debian-based systems while others have said that it just plain doesn't work. The `threshold` option did *not* work on my Fedora test system.

Now that you have the basics down, create an example configuration. Suppose you have a mission-critical server with an IP address of 192.168.1.100. You want `swatch` to send you an e-mail for any message containing that IP address. You don't want to get 500 e-mails in five minutes if the server crashes, so you want a throttle such that you won't generate more than one e-mail every five minutes at most. The following lines at the beginning of the `swatch.conf` would accomplish this.

```
SYNGRESS
syngress.com
Watchfor      /192.168.1.100/
exec sendmail -f Syslog@yourcompany.com -t recipient@some.com -u ALARM -s
MAILsrvr.com -m "Problem with 192.168.1.100"
throttle 00:05:00
```

The `-f` option is the *from* address; `-t` is the *to* address. `-u` specifies the subject line and `-s` specifies the outbound SMTP server to use. The option `-m` is to specify the message body. The variable `$0` will be expanded by `swatch` to include the entire log entry as the e-mail body. You could also use `-o message-file=/var/logs/somelog.log` to create the body from a file you specify.

Suppose you want to know anytime anyone is denied access to anything. You can color any message with “access denied” appearing in it with blue text. To make sure it gets your attention, you can also have each occurrence ring the system bell twice. Use the following lines to configure this.

```
watchfor      / [Aa]ccess [Dd]enied/
echo blue
bell 2
```

Finally, you have an entire range of development servers. These servers are constantly generating log entries that would be alarming if they weren't on development boxes. Fortunately, these systems are all located in the same building, with most of the development servers residing on the 192.168.123.0 segment. Further, the development team has a server naming convention that all servers start with the name DEVEL. You can use the following lines to tell swatch to ignore a message that contains both "192.168.123." and "DEVEL" in it.

```
Ignore      /192.168.123.&DEVEL/
```

Finally, use the following line to log everything else to the terminal using normal text.

```
watchfor      /.*/
echo
```

If you read the man page, there are many color options available for highlighting text, but not all of them will work on all systems, so you will have to do a little testing to see which colors you really have available to use in your terminal. *Watchfor /.*/* is the default, which will trigger on everything, and is used if you don't specify any configuration file at all. If you want to see less on the swatch screen, you could configure only the lines you want to see, and then place an *ignore /.*/* at the end. If you read the man page you will notice that swatch supports using *mail* to send an e-mail natively. In order to use mail you need to have a local *mail transfer agent* (MTA) configured, such as sendmail for example. If you need to process e-mail on the host in question, you will probably need an MTA configured anyway, but if log alerts are the only e-mail you will be sending, a command-line utility such as sendEmail might be the simplest way to go.

You'll notice that for the swatch filtering we are using syntax of */somethingtosearchfor/*. Swatch (along with *many* other tools/languages) uses regular expression (*regex*) for the filtering syntax. In our examples the filters were very simple but *regex* commands can look intimidating if you need complex filtering. A good introductory primer on *regex* syntax and a wealth of addi-

tional information can be found at www.regular-expressions.info/reference.html.

Configuring Logwatch

Logwatch is intended to be more of a reporting tool than a live monitor. You can access the home page for logwatch at www2.logwatch.org:8080/. Basically, you run logwatch and specify the log file you wish to analyze. Logwatch has the capability to perform some limited formatting of the output for you. As is often the case, an example should help make logwatch's role clear. The output of *logwatch* —*service sshd* —*print* is shown in Figure 5.29.

Figure 5.29 Output from Logwatch



```
[root@localhost ~]# logwatch --service sshd --print

##### LogWatch 7.1 (11/12/05) #####
      Processing Initiated: Mon Jul 10 23:19:05 2006
      Date Range Processed: yesterday
                           ( 2006-Jul-09 )
                           Period is day.
      Detail Level of Output: 0
      Type of Output: unformatted
      Logfiles for Host: localhost.localdomain
#####

----- SSHD Begin -----

SSHD Killed: 1 Time(s)

SSHD Started: 1 Time(s)

Illegal users from these:
  192.168.1.108: 4 times

Users logging in through sshd:
  root:
    192.168.1.108: 7 times
```

```
----- SSSHd End -----
##### LogWatch End #####
```

The commands told logwatch you wanted it to report on the *sshd* service and to print the results of the analysis to the standard output (that is, the terminal window). At first glance this might seem trivial, but logwatch can be invaluable in helping to sort through very large logs and extract the meaningful information in an easy-to-understand format. Another command-line example is the *logwatch* `—service sshd —print —archives —range all` option. This tells logwatch to parse not only the logfile specified, but *all* archived files of that log family. For example, if used with the option `—logfile messages`, logwatch parses `/var/log/messages` in addition to `/var/log/messages.*` variations, such as `/var/log/messages.1`.

Logwatch is a *generic* log analysis tool in that it can be configured to analyze other types of log files in addition to the ones it understands by default. If you attempted to run *logwatch* `—service customapp —print`, you would get an error that “logwatch does not know how to process service: customapp.” You can, however, add the capability for logwatch to report on *customapp*. First of all, you need to understand where logwatch stores its configuration information. It uses these directory structures:

/usr/share/logwatch/default.conf/ holds the original configuration files shipped with logwatch.

/usr/share/logwatch/dist.conf/ holds the configuration files shipped with your specific distribution.

/etc/logwatch/conf/ holds any configuration files specific to your particular system.

Under these directories there are several subdirectories, all of which may not be present under all directories.

/services/ is used to define the configuration files for each service. The presence of a given service file is used by logwatch to determine if it understands a service in order to report on it. For example, if you have a `/services/customapp.conf` file in this directory, logwatch will not give you an

error when you ask it to report on `customapp`. The `customapp.conf` file in this directory will specify a log file *group*, which in turn will specify which log files to inspect.

`/logfiles/` contains log file *group* configuration files. Basically, these files define which log files fall into a given group. These files also define where to look for the archives of a given file. So you could use the `/logfiles/custom.conf` to specify which log files should be inspected when reporting on `customapp`. These could be multiple individual log files, such as `/var/logs/custom1.log` and `/var/logs/custom2.log`.

The exact filters and configuration files you need will vary from application to application, so we could not possibly provide instruction on your specific environment. Refer to the instruction located at www2.logwatch.org:8080/tabs/docs/HOWTO-Customize-LogWatch.html for customizing your logwatch for your specific application. There is also a community of logwatch users that may have already generated the needed customization for your application if it's a commonly available product. You can find a lot of help and discussion on logwatch on the logwatch mailing list; you can sign up at www2.logwatch.org:8080/tabs/lists/.

Securing Your Event Logs

In much the same way that having event logs no one reviews is of little value, having event logs you can't trust is also of very little value. If you are trying to reconstruct a set of events, you need to be sure that the logs have not been tampered with. Erasing the logs is typically one of the first steps a hacker makes after gaining access to a system. In this way, not only are legal repercussions more difficult to enact, but securing the system to prevent the break-in becomes all the more difficult if you don't know how access was obtained. Most concerns around log reliability will be related to one of two areas: chain of custody and log integrity. We will look at ways to ensure both, because they are heavily interrelated, and in the process make your network more secure.

Ensuring Chain of Custody

Chain of custody simply means knowing who has had possession of an item at all times. These same principles apply to physical objects as well as to data. In the case of a physical object, it means who had possession of the object itself, or who had physical access to said object. In the case of digital data such as log files, chain of custody includes physical access to the media that stores the logs, as well as logical access to the log data. The objective is to be able to prove who had custody (or access) to the data. If you ever need to pursue legal actions against a hacker, ensuring chain of custody becomes especially important. If your log server is sitting on the network and everyone has access to the log files, the logs will likely be of minimal value in a court case.

The ways one can ensure chain of custody with data are many and varied. Much like the concept of security in depth, a multitude of controls will serve better than relying on a single control or method. All log data should only be accessible to those who need it. These restrictions can be accomplished through file-level access controls, limiting which user accounts can log in to the logging server, and even network-level firewall and isolation. At the physical layer, you need to limit who can physically gain access to the logging server, through door locks, security guards, ID badges, and the like. This primarily addresses who has access to the log data. You will also want to create an audit trail for when the data is accessed in an authorized fashion.

All access to log files (authorized or not) should be logged. These logs should be stored in a different location that the normal log administrators do not have access to. In all reality, you would rarely need to inspect these logs, but it is certainly better to have them available than not to. In the event that a hard drive or other storage media needs to be changed, there should be some means to document who changed it, when, and why. This is often accomplished with a chain of custody form (similar to what is used with physical evidence in legal proceedings), which accompanies the physical medium as it is moved around. All of these measures work collectively to ensure you know who had access to the data at all times. The next step is ensuring that the data has not been altered at any point in time.

Ensuring Log Integrity

The next best thing for a hacker to do if he can't delete the logs, is to alter them. While you might think it would take something extreme, all he has to do is alter them in any obvious way, and they will lose most of their value for legal proceedings. This is because, as soon as it is demonstrated that any part of the logs have been altered, the integrity of the entire log cannot be guaranteed. There are a lot of ways to ensure the integrity of a log file. Some people have advocated printing the logs to create a hardcopy and storing those in a secured location. This has mostly gone out of practice because the volume of logging information has increased dramatically. Another option is to write the logs to a write-once medium, such as a recordable CD-ROM or DVD. This has the advantages that the storage capacity is reasonable and the physical media is relatively small, making secure storage cost effective. The simplest and most common means of ensuring a file's integrity is to generate a hash value at the time the file is generated and to store this hash value on a read-only medium.

A hash is the result of a mathematical computation performed on a set of data. This result, called a hash, hash value, or message digest, is (ideally) unique and reproducible for a given input. In practical terms this is similar to a digital fingerprint for a set of data. If you were to generate a hash using the entire book "War and Peace" as input, you would get a particular hash value. Anyone else who used the same hash algorithm with "War and Peace" as input would also generate the same hash value. If any single character were altered in the entire book, such as a period being changed to a comma, the hash value would be different. A hash function is a one-way computation, meaning there is no way to derive the original input from a known hash value. In the case of a log file, this means if you were to review the log files at a later date, you could recompute the hash value using the log file as input and compare this with the original hash value that was generated when the log file was created. If the two hashes match, the log file has not been altered.

There are many utilities for generating a hash value and many different algorithms that are widely used. Some algorithms are more "secure," in that the odds of two different inputs producing the same hash value are smaller. SHA (secure hash algorithm) and MD5 (message digest 5) are very commonly used

algorithms that are considered to be secure enough for most uses. Remember that the more automated your processes are, the less likely an error will occur, and the more reliable the log data will be in a legal setting. There are many utilities available; we will review the use of a few of the best ones below. If you are collecting your logs to a Windows server, an excellent utility is `fsum` from <http://www.slavasoft.com/fsum/index.htm>. `Fsum` is freeware and can be purchased for commercial use. The license agreement allows `fsum` to be run on only one computer at a time. We would recommend, as with all free products, that you review and understand the license agreement yourself. `Fsum` will run on Windows 9x, NT, 2000, and XP and can generate 13 different types of hashes. In the following example of the hash values for several common hashing algorithms, we removed some of the redundant text from subsequent examples to conserve space.



```
C:\>fsum input.txt -md5
```

```
SlavaSoft Optimizing Checksum Utility - fsum 2.5
```

```
Implemented using SlavaSoft QuickHash Library <www.slavasoft.com>
```

```
Copyright (C) SlavaSoft Inc. 1999-2003. All rights reserved.
```

```
; SlavaSoft Optimizing Checksum Utility - fsum 2.5 <www.slavasoft.com>
```

```
;
```

```
; Generated on 11/21/06 at 12:29:09
```

```
;
```

```
345dd07cc1cf8ba6b9da0ffa7886e2cd *input.txt
```

```
C:\>fsum input.txt -sha1
```

```
; SlavaSoft Optimizing Checksum Utility - fsum 2.5 <www.slavasoft.com>
```

```
; Generated on 11/21/06 at 12:30:30
```

```
345dd07cc1cf8ba6b9da0ffa7886e2cd *input.txt
```

```
C:\>fsum input.txt -crc32
```

```
; SlavaSoft Optimizing Checksum Utility - fsum 2.5 <www.slavasoft.com>
```

```
; Generated on 11/21/06 at 12:30:34
```

```
345dd07cc1cf8ba6b9da0ffa7886e2cd *input.txt
```

There are plenty of GUI-based hashing utilities as well, though generally the terminal-based ones are simpler to automate. If you are using Linux as

your log collector, you are likely to have `md5sum` or `sha1sum` (preferred) already installed. You can also use the `openssl` suite of software to generate message digests using a wider variety of algorithms. Both `md5sum` and `sha1sum` are special-purpose programs that only generate hash values using their respective algorithms. `OpenSSL` can generate the following hash types: MD2, MD4, MD5, rmd160, SHA and SHA1. Here are some examples of `md5sum` and `sha1sum` approaches.



```
# md5sum /input.txt
3ecb68cc0a0f5bff183bbe4d53cfe522 /input.txt
# sha1sum /input.txt
afd7e214ec0c04d07c27b0f3412c477406a012e1 /input.txt
# openssl md5 /input.txt
MD5(/input.txt) = 3ecb68cc0a0f5bff183bbe4d53cfe522
# openssl sha1 /input.txt
SHA1(/input.txt) = afd7e214ec0c04d07c27b0f3412c477406a012e1
```

A regularly scheduled process to generate the hash values can be automated to coincide with the rotation of the log files themselves. For example, if you rotate your logs at midnight each night, you can create/edit the process to generate the hash of the log file once it is no longer being actively modified by the logging process. This hash *can* be stored with the log file, as long as it is *also* stored in a secure location that is separate from the log file. You don't want anyone who has access to the log files to be able to simply generate a new hash and replace theirs for the original; this would circumvent the entire purpose for generating the hashes. As an added bonus, if desired, `openssl` can also encrypt the log files from the command line, which, again, makes scripting this process relatively simple.

Applying Your Knowledge

So now you know how to generate log events for custom applications and processes. You can centrally collect the log files using encryption if needed, and you can ensure that the logs files have not been tampered with. You are also armed with the tools to analyze the log files, and you can do all this for free. You might be wondering just what you should do with all this newfound knowledge or what the next step for applying it should be. Whether you are

the only person who needs to approve of a new log management policy, or whether you need to get your manager to sign off on it, the process remains largely the same—start with a plan. The tasks required to apply these strategies fall into two broad steps, develop a log management policy, and develop an implementation strategy. Both of these will likely be developed concurrently and each will affect the other.

Your *implementation plan* will need to include an analysis of what hosts will be included in the log management scope. The numbers of hosts, their physical location, and the volume of log data they are likely to produce will all impact how you implement your logging policy. You will then need to make a decision on when and where to centrally collect the log files. The location should be strategic, accounting for communications links and other performance metrics. The frequency will include both how often to collect the logs and how frequently to run any analysis on the logs. Perhaps most importantly, you will need to decide what format the log files be processed in, Windows event log format or syslog format. You will need to choose which log management tools to use, which will likely involve some testing in a lab environment. You will need to plan for such administrative concerns as how to preserve the confidentiality of the log data both in transit and in storage.

Your *log management policy* will be a high-level outline of how logs should be managed. This document probably won't go into detail on how to accomplish specific objectives; it will specify only what those objectives are. The policy should address the requirements for ensuring confidentiality of log data. Remember to address confidentiality of log data both in transit and while in storage. Provisions may need to be made for off-site storage much like with server backup tapes. Bear in mind that in some cases there may be external regulations that apply to these requirements. In some cases, log file integrity may need to be addressed as well. This is especially likely if there are regulations that affect your organization, such as HIPAA. You will need to develop a strategy around event alerting as well. This could include notifying the security group if the log messages indicate a possible intrusion attempt or a different person/group if the log messages indicate a key server is having problems. The alerting hierarchy will need to be documented and the configuration of the monitoring software configured to reflect what is documented. As you can see, the policy you define will need to include management sign-off.

Summary

We hope you now have a good understanding of the role of system logs and the value of centralized consolidation and monitoring. There are a lot of tools available covering many aspects of log file collection, generation, import, export, and analysis. A vital Web site for anyone who is interested in log file analysis topics is www.loganalysis.org by Tina Bird. In addition to many informative articles on log file analysis, there are many links to a wide variety of utilities as well. After a little research, you can make an informed decision and generate a plan for your organization's log file archival and monitoring strategy.

Solutions Fast Track

Generating Windows Event Logs

- ☑ If you wish to process the event logs in syslog format, consider using any of several products that are available to export your Windows event logs to a syslog server in syslog format.
- ☑ Microsoft provides several command-line utilities to create and manipulate event log entries. These will enable you to create event log entries for custom applications or batch files.

Generating Syslog Event Logs

- ☑ Syslog is the mostly widely implemented logging format and is natively understandable in virtually every non-Microsoft operating system.
- ☑ The use of TCP syslog messages will enable a wider range of encryption options as compared to UDP-based syslog messages.

Securing Your Event Logs

- ☑ Remember to maintain adequate chain of custody. You need to be able to say beyond a doubt who has had access to your log files during all portions of their life cycle.
- ☑ Generate a hash value of all log files when they are generated and store these in a secure read-only location. This will enable you to prove that the logs have not been tampered with if you need to in the future.

Applying Your Knowledge

- ☑ Your logging strategy should include considerations for centralized logging, log analysis and alerting, and maintaining confidentiality of log data.
- ☑ SSL, SSH, and IPsec can all provide encryption of log messages while in transit between hosts.
- ☑ Live log analysis typically aims to filter and highlight output to limit the data to only the most-relevant information so that a human can judge what events are significant.
- ☑ Offline log analysis is typically used for reporting metrics, incident investigation, or fine tuning the live analysis configuration.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: I have many hosts at a remote office and I want to encrypt the UDP-based logs from all of them back to a central logging server; is there a simpler solution than enabling IPsec on all my hosts?

A: Sure, you could accomplish this any of several ways. You could collect all the logs to a single syslog server at the remote office and then configure the secure IPsec encryption from that single server back to the central office. You could also configure a gateway device to communicate securely with the central office in tunnel mode, so that all traffic from all hosts that pass through that gateway will be encrypted. Using either option, only one system needs to be configured with IPsec.

Q: What is the simplest way to generate log entries for testing purposes?

A: For generating syslog events from a Windows host, Kiwi SyslogGen is indispensable. You can download it from www.kiwisyslog.com/info_sysloggen.htm. For Linux, you can simply pipe data into the log files directly, such as `echo kernel.emergency >> /var/log/messages`. If, for example, you had `swatch` running, it would then be able to trigger on `kernel.emergency` (assuming it was watching `/var/log/messages`, which it will by default).

Q: How long should I keep my log files?

A: That will depend on your business and environment. If you are in an environment where risk of a security incident is higher, such as an ISP, that will surely affect how long you should keep your logs. If you have government regulations, the retention period might be mandated to you. If your logs contain any information that would fall under HIPPA, then an entirely different set of requirements becomes relevant. When deciding

on rotation and retention periods, work closely with your legal department to ensure that you are meeting all your obligations.

Q: Which hash algorithm is the best?

A: As with many security questions the answer is, “it depends.” The more secure hashing algorithm tends to be more computationally intensive. This may not be noticeable with a small input file, but with a very large file, the time it takes to generate the hash can be significant. Thus, a more practical answer would be, “use the most secure hashing algorithm that you can without adversely impacting the system processes.” From a security perspective, the “risk” with a hash algorithm is that the same message digest can be produced from two different inputs. Newer algorithms and longer hashes decrease this risk. So, in order of preference, use SHA-512, SHA-256, SHA1, SHA0, and MD5. Perform some testing and see which ones operate at an acceptable speed for your application.

Testing and Auditing Your Systems

Solutions in this chapter:

- Taking Inventory
 - Vulnerability Scanning
 - OSSTMM
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Sooner or later you will need to identify all the systems on your network. Despite the most stringent of usage policies, sometimes undocumented systems may be added to the network. Sometimes these systems are “test” systems that were never decommissioned. At other times you may find “rogue” systems whose mere presence on the network violates policy. There may be instances where the system is managed by a third party as part of a vendor’s service offering. The value of a full network discovery is even more apparent if you are dealing with an environment that you are not familiar with, such as a newly acquired company, or if you are new to your position. If the network has few enough hosts, this task isn’t much of a challenge. If the network is large, or spread across multiple locations, and visiting them all isn’t practical, an automated discovery may be much more practical. We will look at some generic discovery/scanning tools, as well as some that are targeted at specific services.

After you have identified all the systems on your network, the next logical step is to determine the security posture of those systems. Several automated security scanning tools are available that can check for a large list of known vulnerabilities and can make this task easier. We will demonstrate the configuration and operation of some automated vulnerability scanners. We will also discuss the Microsoft Baseline Security Analyzer, which simply checks a Microsoft system and reports on any known security issues it finds. Finally, there are some formalized security testing methodologies that you can use to assess the security of a system, beyond simply running a vulnerability scanner.

Taking Inventory

In a perfect world, you would have 100 percent accurate and complete documentation encompassing every system that is connected to the corporate network. No one with access to the network would ever connect a system to the network without all the proper documentation and approvals to do so. Well, we all know “perfect” doesn’t exist. Perhaps you have a specific reason to do the network discovery, or maybe not. A periodic discovery is a good idea anyway, even if you don’t have any specific reason to do one. It can provide

assurance that policies are being followed when you can successfully produce documented approval for all devices on your network. A host inventory can also demonstrate that your documentation matches the true state of the network and that routers and switches are where they are supposed to be. Given the fact that systems can be very hard to locate physically, especially given the increasingly smaller size of wireless access points, a network-based discovery is often more fruitful than a physical one.

Locating and Identifying Systems

There are two primary steps to performing a network inventory. The first step is simply to identify the existence of a system. There are a number of ways to do this; typically a combination of methods will result in the most accurate inventory. Pinging entire blocks of IP addresses will identify most systems. If the system is configured not to respond to a ping, however, it will of course be missed. This occurs most often when a personal firewall is running on the host that is blocking network pings. Even in cases where a system will not respond to a ping, the host is usually listening on *some* port. A more comprehensive TCP-based port scan will often reveal the presence of systems that a ping scan will not. Further, by capturing the initial output for each port you can often gather more information, which can be used to identify the listening software or host. For example, if you connect to TCP port 21, and it responds with HTML headers, you could probably conclude that the system is running a Web server on the port normally used for FTP. You can inspect the DHCP scope on the DHCP servers in an attempt to identify a system that is not authorized to be on the network. Wireless systems can be identified relatively easily due to the fact that they must transmit a signal in order to communicate. Depending on the size of the network, you may even be able to take an inventory of the ports used on switches and routers, or for those with a lot of time on their hands, by cross-referencing the ARP tables of the switches with a list of known hosts. In 99 percent of the cases, however, a simple ping scan of all the network IP addresses combined with a TCP and UDP scan of a few key ports will provide a very good inventory of the hosts on the network.



A well-secured network will hinder exactly the types of inventory-building activities you will be performing. The same techniques that stop a hacker from mapping out your network will also hinder you as an admin. If you are not able to see the results you are expecting, remember that firewalls, VLANs, IPsec, and other security measures may skew your results.

After you have identified the systems that exist on your network, the next step is more time consuming: determining *where* the system is physically located. In some cases, maybe you don't need to, particularly if they are authorized systems, or if you can identify a means to contact the person responsible for the system in order to make the system "legal." If you do find a rogue system, however, you will want to see where it is located and perform other information-gathering steps in an attempt to get it removed from the network or complete the needed procedures for the system to have *authorized* access to the network. Sometimes this process is relatively simple, such as when the system is using a host-naming convention that tells you its location and maybe even the server role, such as DALLASWEB01.somecompany.com. In other cases you may need to use the IP address and *traceroute* to track down the physical location based on the subnet combined with a good network map (we'll go over an example in the next few paragraphs). In the case of a wireless system (host or access point), locating the rogue system can be particularly challenging.

Remember that a network device inventory is a living document. It will take time to perform an IP scan, track down any devices that you weren't familiar with, and verify network access approval or seek approval for all devices. By the time you're finished, it will probably be time to start the process over. Because the network is rarely a static entity, this type of discovery should be performed on a regular schedule. You may have local policies that dictate how frequently the discovery should be. If these policies are not present, you should develop a process and make it a part of your normal business operations. In this way, rogue systems can be located in a minimal amount of time and you can minimize any security risk that these systems may pose.

The contents of your inventory documentation will vary according to your needs, but there are some common elements. At a bare minimum you will want to know the IP address, host name, and contact information for the person(s) responsible for administering the device. You could get as detailed as including hardware specifications (manufacturer, model, memory, etc.), MAC address, administrative contacts, emergency contacts, operating system type and version, and much more. Ultimately you will want to customize the documentation to your business needs. Perhaps deploying biometric authentication is a priority, in which case you might want to include a column indicating which devices have fingerprint scanners attached to them.

Nmap

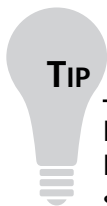
Nmap is the most widely used general-purpose network scanner. It is available from <http://insecure.org/nmap/> for Windows, Linux, MAC OS X, Sun Solaris, and several other operating systems. The operation of Nmap is largely the same whether you are running it on Windows or Linux. The most notable exception is that you will need the Windows packet capture driver, WinPcap, if you are running Nmap on Windows.

NOTE

The latest version of Nmap supports raw sockets, which means that if you are using Windows 2000, Windows XP, or Windows 2003 Server, you don't need the WinPcap drivers. For older versions of Windows you will still need WinPcap.

Nmap can scan for open ports using a variety of standardized TCP packet options, as well as using some of the options in non-standard ways. There are a large number of command-line options, which can sometimes appear confusing, but the Nmap documentation and support on the Internet are both very good. Periodically, a GUI front end will come and go, but currently there are no Windows front ends for Nmap being actively developed. NmapFE is a GUI front end for Linux and it is actively maintained by the creator of Nmap. The GUI has the benefit of enabling you to check boxes for

various options instead of requiring you to know a more complex command-line syntax.

**TIP**

Be aware of the underlying network topology that you are working with. If you are scanning a host on the other side of a firewall it will likely severely alter your results. In some cases, even an ISP will filter out certain ports. Although this prevents those ports from being available over the Internet, they might still be available locally, and possibly still pose a security risk.

If you have the Windows packet capture driver (WinPcap) installed and working properly, all that is needed to install Nmap on Windows is to extract the contents of the Zip download to a directory and run the Nmap executable. On Linux you can download and compile the source code, or install it as an RPM. When you run it with no options, you will see a lengthy help screen with a few examples. For the real treasure trove of helpful information, refer to the Nmap man page located at <http://insecure.org/nmap/man/>. If you are comfortable working on Linux or Windows, Nmap functions almost identically on either. There is, however, one difference that can be significant, which is speed. Nmap runs much faster on Linux than Windows. In a small network this may not be a consideration, but if you are scanning a large number of hosts, or ports, the difference in scan times can be significant.

Let's go through some examples of how you could make use of Nmap. Let's suppose you want to do an initial scan of your entire company network. If your company is using the private address space 192.168.0.0 or some portion thereof, you could scan the entire class B network, sending only a ping to see if the system is "alive" with the following command line.

```
nmap -v -sP 192.168.0.0/16
```

This would perform the most basic type of scan, which is a ping scan only, as specified by the use of the `-sP` option. You can see more information by using the `-v` option, which tells Nmap to be more verbose; in most cases

you will find the extra information informative. This option can also be used multiple times for even more information, so `-v`, and `-vv` are both valid. Because it is fairly common for a personal firewall to block ping attempts, you may have better luck if you run the scan without the `-sP` option. If you don't specify a scan type, Nmap will default to a TCP SYN scan (same as `-sS`). The normal TCP *three-way* handshake consists of the initiating system sending a packet with the SYN bit set. The target host responds with a packet with the SYN and ACK bit set. The original system then sends an ACK packet back to the target. In this fashion a TCP session is established, which is followed by the desired communications. The SYN scan (`-sS`) will send the initial SYN packet, but when the target host replies with a SYN ACK, Nmap never completes the three-way handshake to fully establish the session. This method is so fast and efficient that it is the default scanning method Nmap uses.

If you do not specify which TCP *ports* to scan, Nmap will scan all TCP ports defined in the `nmap-services` file, which at the time of this writing is 1680 of the most common ports. So let's suppose during your ping scan of the entire network a system was identified that you didn't recognize (192.168.1.106) and you want to find out more about it. After the ping scan you could perform an Nmap scan with no options and see which of the most common ports are open. The output of `nmap 192.168.1.106`, being a typical single-host scan with no other options specified, is shown in Figure 6.1.

Figure 6.1 Nmap Results



```
C:\Apps\Nmap>nmap 192.168.1.106
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2006-09-17 14:54
Central Standard Time
```

```
Interesting ports on 192.168.1.106:
```

```
Not shown: 1676 closed ports
```

```
PORT      STATE SERVICE
```

```
135/tcp   open  msrpc
```

```
139/tcp   open  netbios-ssn
```

```
445/tcp   open  microsoft-ds
```

```
5101/tcp  open  admdog
```

```
MAC Address: 00:08:02:32:8A:4C (Compaq Computer)
```

```
Nmap finished: 1 IP address (1 host up) scanned in 2.172 seconds
```

From these results you can see that the system has TCP ports 135, 139, and 445 open, most likely indicating a Windows host. Just to confirm your suspicions, you could use Nmap's operating system fingerprinting feature. Any given system on the network was likely programmed slightly differently, resulting in slightly different ways of responding to network traffic. Nmap can use these subtle differences in responses (such as TCP ISN [initial sequence number] sampling, TCP options support and ordering, IPID [IP ID] sampling, and the initial window size) as clues and compare them with Nmap's nmap-os-fingerprint database. If it finds a match in the database, there is a good probability that the actual OS can accurately be identified. An example of the OS fingerprinting using the `-O` option in action is shown in Figure 6.2.

Figure 6.2 Nmap OS Fingerprinting



```
I:\HackApps\Nmap>nmap 192.168.1.106 -O
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2006-09-17 15:00
Central Standard Time
```

```
Interesting ports on 192.168.1.106:
```

```
Not shown: 1676 closed ports
```

```
PORT      STATE SERVICE
```

```
135/tcp   open  msrpc
```

```
139/tcp   open  netbios-ssn
```

```
445/tcp   open  microsoft-ds
```

```
5101/tcp  open  admdog
```

```
MAC Address: 00:08:02:32:8A:4C (Compaq Computer)
```

```
Device type: general purpose
```

```
Running: Microsoft Windows 2003/.NET|NT/2K/XP
```

```
OS details: Microsoft Windows 2003 Server or XP SP2
```

```
Nmap finished: 1 IP address (1 host up) scanned in 2.813 seconds
```

Nmap identified the system as either Windows 2003 Server or Windows XP with service pack 2. Further, you may notice that Nmap has identified the system as a Compaq based on the MAC address. With all this information you have a pretty good idea of what type of system this rogue PC is. The next step would likely be to find out where it is physically located. Assuming you don't recognize the subnet as belonging to a specific location, *traceroute* will

use ICMP to try to trace each router between you and the target host. An example of *traceroute* output is shown in Figure 6.3.

Figure 6.3 Traceroute Output



```
I:\HackApps\Nmap>tracert 192.168.1.106
```

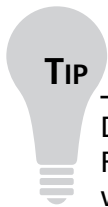
Tracing route to 192.168.1.106 over a maximum of 30 hops:

```

 1      2 ms      2 ms      2 ms  192.168.102.1
 2     11 ms     14 ms     10 ms  10.10.10.1
 3     12 ms     10 ms     11 ms  router1.houston.your-co.com [10.10.20.1]
 4     14 ms     12 ms     12 ms  router2.austin.your-co.com [10.10.30.1]
 5     14 ms     10 ms     13 ms  router3.dallas.your-co.com [10.10.40.1]
 6     20 ms     18 ms     17 ms  router4.orlando.your-co.com [192.168.2.1]
 7     19 ms     20 ms     17 ms  192.168.1.106

```

Trace complete.



TIP

Different systems may have different commands to do the same thing. For example, on Windows systems the *traceroute* command is *tracert*, while on Linux systems it is *traceroute*.

I have edited the actual IP addresses and host names, but you can try the *traceroute* command to a few hosts in your network. Because it is very common to include some indication of the geographic location in the naming convention for routers, often this will tell you where the host is located. In Figure 6.3, hop #6 would lead me to believe the host was in Orlando, Florida. Assuming you had a *managed* switch in Orlando, you could then Telnet to the switch (in this example a Cisco 2900XL switch) and view the table of MAC addresses. Referring to our previous Nmap scan, we know the MAC address of our mystery system is 00:08:02:32:8A:4c, so we can use the following command to filter the MAC table to show only the MAC address we are interested in:

```
SWITCH#Show mac | incl 0008.0232.8A4C
0008.0232.8A4C      Dynamic      1      FastEthernet0/2
```

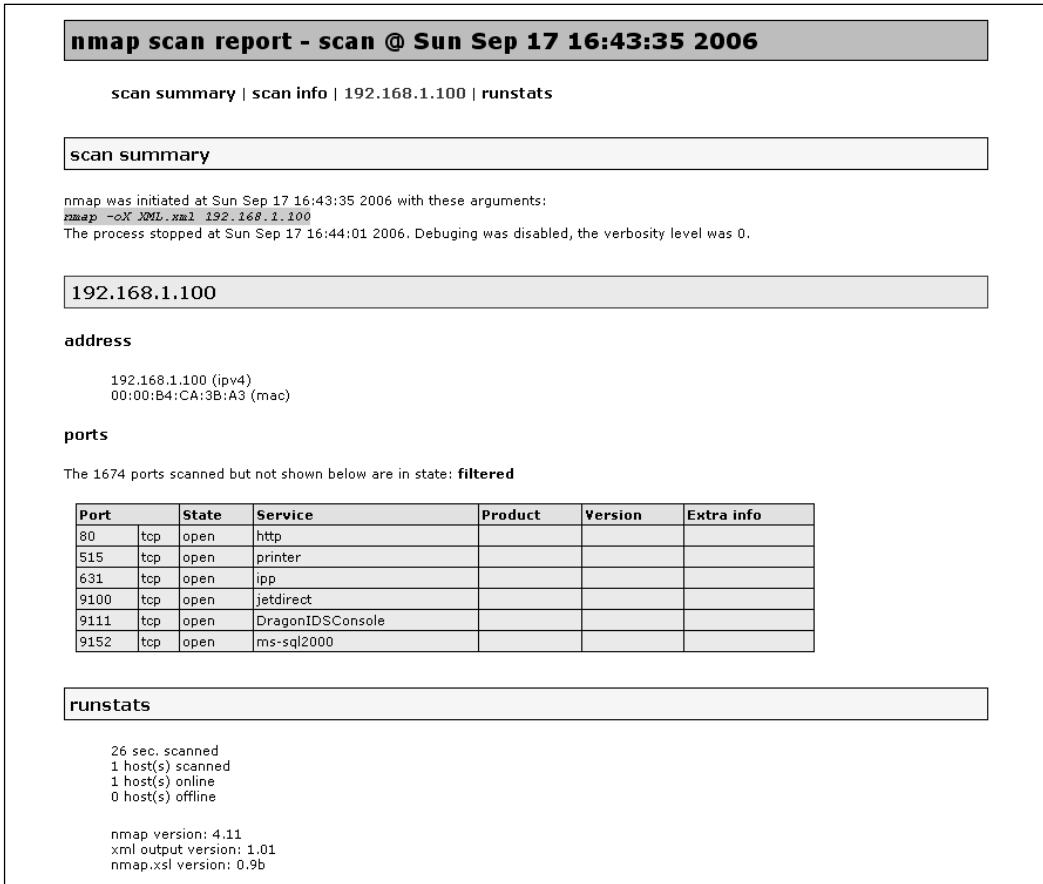
We could now provide an exact network port (port 2 on the switch) for someone who has local access to trace the cable and find the mystery machine. As you can see, Nmap has a lot of features. There are a large number of options that focus on avoiding IDS detection. There are many additional options that manipulate the TCP packets in far more unusual ways. Although these options aren't for everyone, even if you don't need to use these special options yourself, it is good to be familiar with them as a security professional. There are also options that specify the timeout period to be used when attempting to connect. The defaults are usually adequate, but you can use more aggressive timing if you want to speed up the scans. Although the Nmap man page is practically a necessity if you are going to be doing much scanning, Table 6.1 highlights some of the most useful command-line options, as a sort of tip sheet.

Table 6.1 Nmap Options

Nmap Options		
Option	Example	Notes
--exclude	nmap 192.168.1.1-254 --exclude 192.168.1.106	These are especially important when scanning large blocks of IP addresses so you can avoid certain critical servers.
--excludefile	nmap 192.168.1.1-254 --excludefile file1.txt	
-sP	nmap 192.168.1.1-254 -sP	Performs an ICMP ping scan only.
-sV	nmap 192.168.1.1-254 -sV	Attempt to determine service/version on open ports.
-sT	nmap 192.168.1.1-254 -sT	Performs TCP scan using 3-way handshake for each port
-p	nmap 192.168.1.106 -p135,136,137 nmap 192.168.1.106 -p U:514,T:514	Scan only the ports you specify, using TCP or UDP. U:<UDP ports>, T:<TCP ports>
-PO	nmap 192.168.1.1-254 -PO	Treat all hosts as online. Without this, Nmap will not scan the host if it fails to respond to a ping.
-O	nmap 192.168.1.1-254 -O	Perform OS detection.
-A	nmap 192.168.1.1-254 -A	Determine OS and Version info, same as -O and -sV
-oN <file>	nmap 192.168.1.1-254 -oN normal.txt	Sends the same output you would see on screen to a file.
-oX <file>	nmap 192.168.1.1-254 -oX XML.xml	Sends the output in XML format for web viewing.
-oG <file>	nmap 192.168.1.1-254 -oG grepable.txt	Sends the output in a more easily grep'ed format. Grep is the *nix command line filtering utility, similar in functionality to the Windows find utility.
-v or -vv	nmap 192.168.1.106 -v nmap 192.168.1.106 -vv	More verbose output providing more detail about what actions Nmap is performing.

Nmap is a good general-purpose scanner that can perform a wide variety of scans. The available output formats can be very useful if you should need to provide reports of your scan results. You could even schedule a scan and have the output written to a file in XML, which you could then distribute via e-mail or view on a Web site. Figure 6.4 shows part of the XML output of a sample scan of 192.168.1.100.

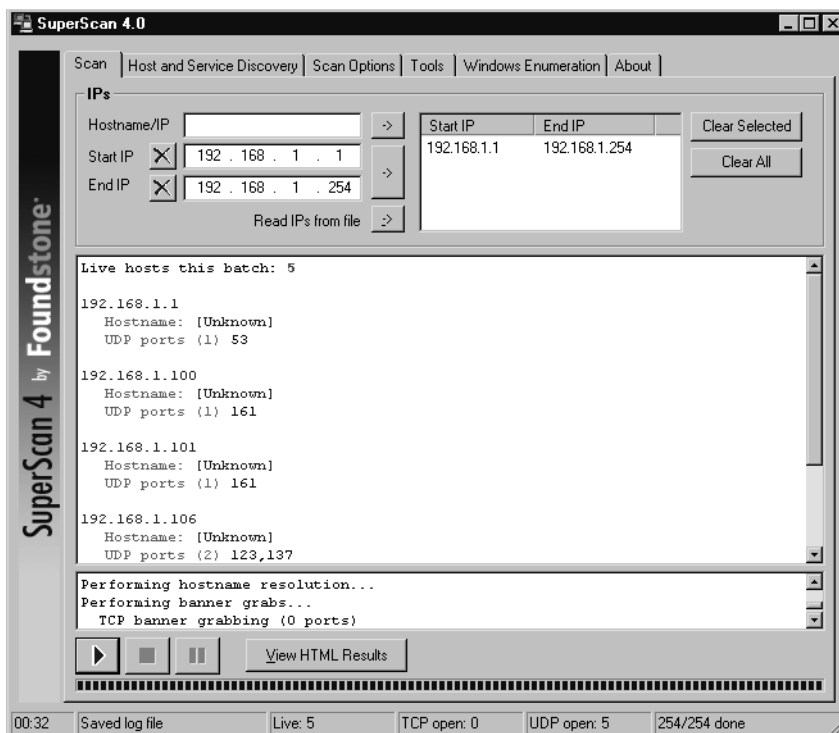
Figure 6.4 Nmap XML Output



Super Scanner

Sometimes you want something simpler than Nmap, or maybe you want to use something that doesn't require the WinPcap drivers to be installed in order to run it on Windows. SuperScanner doesn't require the WinPcap drivers and doesn't even require a setup program. All you need to do is download the program from www.foundstone.com/resources/proddesc/superscan.htm, extract the executable from the Zip file, and run it. The latest version (version 4) will run on Windows 2000 and Windows XP. The main window is shown in Figure 6.5 with some results from systems it found.

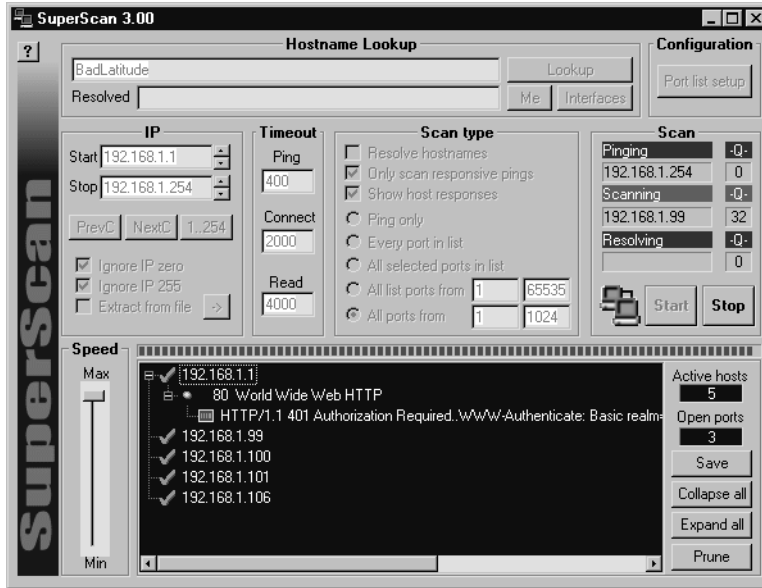
Figure 6.5 SuperScan V4



The operation of SuperScan is rather straightforward. To scan a subnet, simply enter the starting and ending IP address and click the -> button to add it to the scanning queue. If you enter the same IP address and starting and ending IP, you can scan a single host. A third option is to click the -> button next to **Read IPs from file**, which will enable you to browse to a file that contains a list of IP addresses. After selecting the IPs or range of IPs to scan, click the start button at the bottom, which looks a lot like a traditional play button. The authors of the programs suggest using version 3 (shown in Figure 6.6) if version 4 doesn't work properly for you. I have included both versions because, while version 4 offers many more options, my experience has been that version 4 often returns no results after a scan while version 3 works much more reliably. One of the primary reasons to use V4 over V3 is that version 3 and earlier versions support scanning only a single class C network at a time. Version 4 also offers several additional features over version 3, specifically the capability to scan noncontiguous IP address ranges,

additional control of some of the scanning parameters, and some special options aimed specifically at enumerating Windows hosts. As you can see, the improvements in version 4 are significant, so if version 4 does work for you, it would probably be the preferred version to use.

Figure 6.6 SuperScan V3



As you can see, the interfaces between version 3 and version 4 are substantially different. The button used to start the scan is actually a little more clearly labeled in versions prior to version four. When you click the start scan button, the scan will commence. The scan could take some time if you have a large number of hosts or ports to check. There will be a small plus next to systems that have open (i.e., listening) ports. You can expand the plus symbol and see a list of open ports. Expanding the next plus will show any responses the scanner received when connecting to that port.

SuperScan version 4 offers more control over the scanning options that are used, but the biggest difference between version 3 and version 4 is the enumeration options available for Windows hosts. Let's take a minute to talk about what the Windows enumeration SuperScan 4 can do, and special-purpose enumeration tools in general. A definition of enumerate is "to make a concise list of the relevant points." We can refine that definition to fit in a network

security context as “building a list of objects or data points pertaining to a given network host.” This could include things like running services and applications, file shares that are accessible, users and groups on a host, and so on. When it comes to Windows hosts, there is a lot of information gathering that you can do and lists that can be generated. For example, if you wanted to enumerate all the shared folders on a single host, you could use the following command:

```
net view \\192.168.1.108
Shared resources at \\192.168.1.108
```

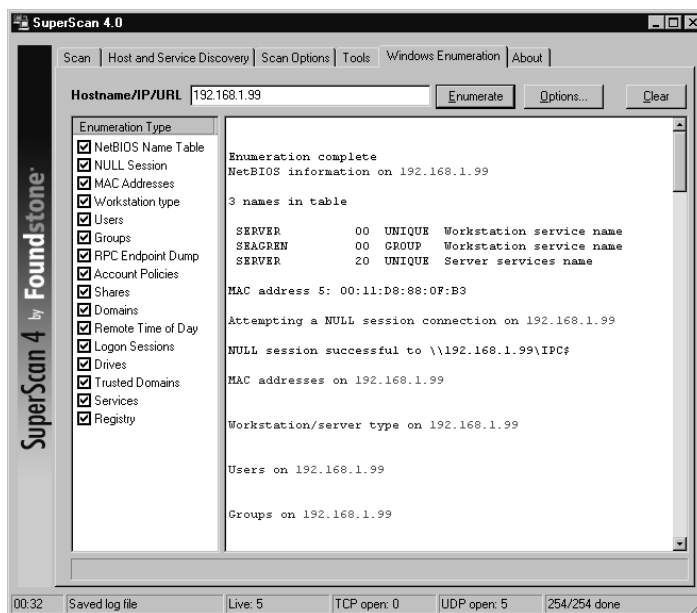
```
Share name  Type  Used as  Comment
-----
```

```
SharedDocs  Disk
```

The command completed successfully.

Many similar processes are automated for you by SuperScan. When you select the **Windows Enumeration** tab (shown in Figure 6.7), you are presented with various checks you can perform in the left pane. After you choose the option you wish to use, click **Enumerate** and the results will populate in the right pane.

Figure 6.7 Windows Enumeration in SuperScan V4



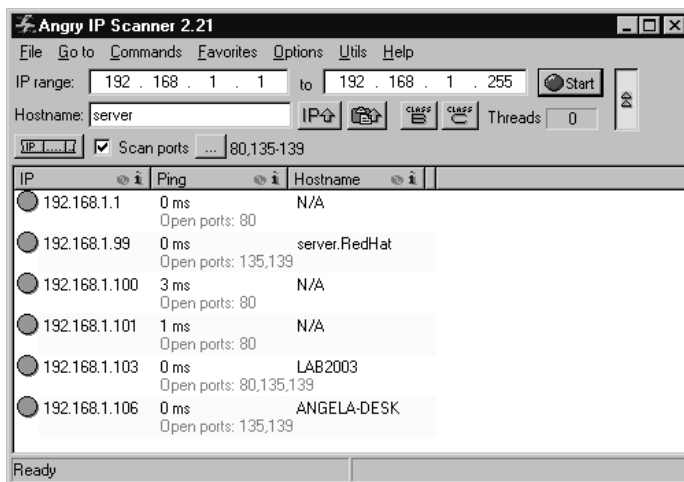
The process of enumerating the shares can be done for all hosts on the entire subnet instead of a single host by checking the **Shares** check box on the **Windows Enumeration** tab, and is just one of the enumeration options SuperScan can use. By default SuperScan will perform all of the enumeration using no credentials, but if you click **Options** on the **Windows Enumeration** tab, you can enter specific account information that should be used for the connections. The NetBIOS Name Table enumeration type is the same information you would get by using `nbtstat -A 192.168.1.108`. This shows the NetBIOS machine name (which can be different than the host name, though it rarely is), and the workgroup/domain the machine belongs to. Depending on how securely the system has been configured, you may be able to get a lot of information from these enumeration techniques. If the system is very secure you will get very little information. In general these checks carry little risk to the target system but as is always the case, if a service disruption is not acceptable, you should avoid running these types of checks because there is always *some* risk involved.

SuperScan has fewer features than Nmap with the exception of the Windows enumeration options, but it is easier to use, and does not require running a Setup Wizard, any registry entries, or special network drivers. Because SuperScan doesn't require any installation per se, and makes no changes to the registry, it can be very useful to have on a pen drive or shared network drive. This type of low-footprint tool can be very useful at times.

Angry IP Scanner

A final GUI scanner that is rapidly on the rise in popularity is Angry IP Scanner (www.angryziber.com/ipscan/). It lies somewhere between SuperScan 3 and 4 in functionality and does not require any installation. Angry IP Scanner also has the advantage that it does not need the WinPcap drivers. Although Angry IP Scanner uses a deceptively simple interface (shown in Figure 6.8), it packs a lot of features into its small file size.

Figure 6.8 Angry IP Scanner



If you select an IP by highlighting it, and then navigate to **Commands | Open Computer**, you are presented with a list of handy Windows options such as **In Explorer**, **Web Browser**, **Ping**, **Telnet**, and **Traceroute**. There are two interesting features of Angry IP Scanner: it is an open source project and it is the only scanner covered that supports being run from the command line *in addition* to running from a GUI.

Scanline

If you happen to be looking for something even *more* lightweight, there is an excellent product available. Scanline, which is also available from Foundstone in its free tools section (www.foundstone.com/resources/proddesc/scanline.htm), is a command-line-only TCP scanner. This can be especially valuable if you do not have remote GUI access to the system you want to scan *from*. If you have only command-line access, such as from an SSH session, Scanline would be a perfect fit. Using Scanline is rather simple. Entering simply *sl* from the Scanline directory will result in the help screen shown in Figure 6.9.

Figure 6.9 Scanline Help

```
C:\Pers\Apps\Scanline>sl
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com
```



```

sl [-?bhijnprstUvz]
    [-cdgmq <n>]
    [-fLlO <file>]
    [-tu <n>[,<n>-<n>]]
    IP[,IP-IP]

```

- ? - Shows this help text
- b - Get port banners
- c - Timeout for TCP and UDP attempts (ms). Default is 4000
- d - Delay between scans (ms). Default is 0
- f - Read IPs from file. Use "stdin" for stdin
- g - Bind to given local port
- h - Hide results for systems with no open ports
- i - For pinging use ICMP Timestamp Requests in addition to Echo Requests
- j - Don't output "-----..." separator between IPs
- l - Read TCP ports from file
- L - Read UDP ports from file
- m - Bind to given local interface IP
- n - No port scanning - only pinging (unless you use -p)
- o - Output file (overwrite)
- O - Output file (append)
- p - Do not ping hosts before scanning
- q - Timeout for pings (ms). Default is 2000
- r - Resolve IP addresses to hostnames
- s - Output in comma separated format (csv)
- t - TCP port(s) to scan (a comma separated list of ports/ranges)
- T - Use internal list of TCP ports
- u - UDP port(s) to scan (a comma separated list of ports/ranges)
- U - Use internal list of UDP ports
- v - Verbose mode
- z - Randomize IP and port scan order

Example: `sl -bht 80,100-200,443 10.0.0.1-200`

This example would scan TCP ports 80, 100, 101...200 and 443 on all IP addresses from 10.0.0.1 to 10.0.1.200 inclusive, grabbing banners from those ports and hiding hosts that had no open ports.

The same scan we performed earlier of the 192.168.1.0 network could be performed with the following command line:

```
sl 192.168.1.1-254 -n
```

The `-n` options tells Scanline to ping only and not to do a port scan. If you simply specify an IP address or range, with no other options, Scanline will scan its internal list of default ports. This behavior is the same as Nmap. The Scanline list of default ports currently includes the following ports.

UDP ports

```
7 9 11 53 67-69 111 123 135 137 138 161 191 192 256 260 407 445 500 514 520
1009 1024 1025 1027 1028 1030 1033 1034 1035 1037 1041 1058 1060 1091 1352
1434 1645 1646 1812 1813 1900 1978 2002 2049 2140 2161 2301 2365 2493 2631
2967 3179 3327 3456 4045 4156 4296 4469 4802 5631 5632 11487 31337
32768-32790 43981
```

TCP ports

```
7 9 11 13 15 19 21 22 23 25 43 49 53 66-68 70 79 80 81 88 89 98 109 110 111
113 118 119 135 139 143 150 156 179 256-259 264 389 396 427 443 445 457 465
512-515 524 540 563 587 593 636 691 799 900-901 1024-1031 1080 1100 1214
1243 1313 1352 1433 1494 1498 1521 1524-1525 1529 1541 1542 1720 1723 1745
1755 1813 1944 2000 2001 2003 2049 2080 2140 2301 2447 2766 2779 2869 2998
3128 3268 3300 3306 3372 3389 4000 4001 4002 4045 4321 4444 4665 4899 5000
5222 5556 5631 5632 5678 5800 5801 5802 5900 5901 6000 6112 6346 6347 6588
6666-6667 7000 7001 7002 7070 7100 7777 7947 8000 8001 8010 8080-8081 8100
8383 8888 9090 10000 12345 20034 27374 30821 32768-32790
```

The entire help file, or manual, is located on the same Foundstone page where you download Scanline. The sample output shown in Figure 6.10 shows the results of a simple scan using the following syntax: `sl 192.168.1.115`.

Figure 6.10 Scanline Results



```
C:\Pers\Apps\Scanline>sl 192.168.1.115
```

```
ScanLine (TM) 1.01
```

```
Copyright (c) Foundstone, Inc. 2002
```

```
http://www.foundstone.com
```

```
Scan of 1 IP started at Wed Aug 30 21:17:06 2006
```

```
-----
192.168.1.115
```

```
Responded in 0 ms.  
0 hops away  
Responds with ICMP unreachable: Yes  
TCP ports: 22 135 139 427 1025  
UDP ports: 137 138 500
```

```
Scan finished at Wed Aug 30 21:17:14 2006
```

```
1 IP and 267 ports scanned in 0 hours 0 mins 8.34 secs
```

Special-Purpose Enumerators

The scanning utilities we have discussed have been general-purpose scanners even if some included specialized enumeration techniques. Although SuperScan 4 includes some special Windows enumeration options, previous version of SuperScan, Nmap, and Scanline do not. In some cases you may want to scan for very specific responses. One example would be to scan for machines infected with the Back Orifice Trojan (BOPing) or to scan for SNMP-enabled devices (via SNScan). Nbtscan gathers NetBIOS information on a network for all devices. Both BOPing and SNScan are available from Foundstone, but there are many more examples of special purpose enumerators available on the Internet. The intended purpose of these special enumerating scanners may vary from legitimate security tools to scanning for systems to launch denial of service attacks from. As is always the case, use caution when downloading such tools from the Internet and research the source of the tool to ensure that you are not introducing a Trojan or virus into your environment. The general purpose scanners are *usually* intended for finding responsive systems and determining what ports they are listening on only.

Are You Owned?

A Word of Caution

Perhaps you have been fortunate enough, or cautious enough, to never have downloaded any malicious software accidentally. One thing you will discover when searching for security software on the Internet is that it is precisely security software that is most often a security risk. Countless Internet sites like to offer up security tools to discover Trojans (or even more commonly, supposedly control the Trojans), to clean a virus, or otherwise protect you when in fact the programs you are downloading are infected with a virus, Trojan, or other malicious software. You must exercise extreme caution when scouring the Internet for security tools or you will become the next victim of unscrupulous people.

My advice is to download your security tools from only the major security researchers. In this way you can be fairly sure that the software will only do what it's supposed to, without any hidden payload. There may be times when you simply cannot find what you are looking for from one of the more mainstream security sites, in these cases you may have to visit some less well-known sites. In these instances I would recommend downloading the software to an isolated test system and only running the software after extensively testing it with a variety of anti-virus and anti-spyware programs. These steps should help minimize the chances of falling prey to malicious software.

Table 6.2 highlights the primary features of each scanner covered in this chapter.

Table 6.2 Scanner Features

	Utility Feature Matrix				
	Nmap	Superscan Version 3	Superscan Version 4	Angry IP Scanner	Scanline
Command Line Support	✓	-	-	✓	✓
GUI Interface Support	-	✓	✓	✓	-
Non-Contiguous Target IP Ranges	✓	-	✓	-	-
Access to non-standard TCP scan types	✓	-	-	-	-
Special Purpose Enumeration	-	-	✓	✓	-
WinPcap Required	✓	-	-	-	-

Locating Wireless Systems

Some of the most difficult systems to locate are ones with no physical connection to the network, such as systems that rely on wireless connectivity. There are many reasons for doing a wireless site survey. If your company uses wireless technology you will probably want to learn what the effective network coverage is. Perhaps you don't want the building across the street to be able to use your wireless access point. On the flip side, you could perform a site survey to map out where your coverage is weak and needs to be redesigned. Or perhaps an employee has installed a wireless access point or repeater and such "rogue" devices are not permitted according to your company policy. If any of these are true you will want to identify that the device exists, and, if necessary, attempt to locate the physical device. Physically locating the system is more of an art than a science. A directional antenna and a little triangulation can help you get pretty close to a wireless device. A directional antenna can have as small as a 15-degree reception arc and when it comes to triangulating, the smaller the reception arc, the better.

The pastime of taking a laptop computer and driving around with a wireless network card and a wireless scanning utility such as NetStumbler is called *war driving*. This term is derived from an even older technique in which you use a modem to dial large blocks of telephone numbers to see if

any computers answer the call, which is called *war dialing*. With the ever-increasing portability and wireless access points becoming so cheap and prolific, you no longer need a car to locate them. Simply walking around with your laptop looking for wireless signals has become known as *war walking*. There are Web sites, and groups of people, who make it a pastime to locate unsecured (i.e. publicly accessible) wireless access points and map them so that others will know where to find free wireless access. This is one such Web site www.wigle.net/gps/gps/Map/onlinemap/?state=TX&s=Show+State where if you zoom in enough, it shows the individual SSIDs of the wireless access point.

Network Stumbler

NetStumbler, which is short for Network Stumbler (www.netstumbler.com), is a tool to detect wireless using 802.11a, 802.11b, and 802.11g. In addition to simply passively listening for indications of wireless devices, NetStumbler will send out various types of traffic in an attempt to solicit additional information from the device. In practice, NetStumbler is very easy to use. The only real concern is making sure you are using a wireless card that NetStumbler supports. Although there are no guarantees, typically sticking with cards that use the Lucent Orinoco chipset, or Cisco cards will provide good performance and compatible hardware. Senao also offers a higher power card with excellent sensitivity that I have used myself very successfully. Unfortunately, there is no comprehensive list of supported cards so a little research before buying can really pay off here. The NetStumbler site does contain some useful information on supported cards, though. When you start up NetStumbler you will see a screen similar to the one shown in Figure 6.11.

If everything is working properly, NetStumbler will start up in scanning mode and hopefully produce a list of detected devices. If you aren't getting any results and think you should be, navigate to **Device** at the top of the window and see if the proper network card is selected. You can change the selected card without stopping the scanning. The number of results you get will vary greatly depending on the quality of the wireless card and antenna you use.

Figure 6.11 NetStumbler 802.11b

MAC	SSID	Name	Chan	Speed	Vendor	Type	Encryption	SNR	Signal+	Noise-	SNR+
00152B782A00	hhonors		1	54 Mbps	(Fake)	AP		12	-81	-100	19
00152B783080	hhonors		1	54 Mbps	(Fake)	AP		14	-86	-100	14
00152B783081			1	54 Mbps	(Fake)	AP	WEP	17	-83	-100	17
00152B782A03			1	54 Mbps	(Fake)	AP		16	-84	-100	16
001195D627E3	default		6	54 Mbps	(Fake)	AP		18	-52	-100	48
00152B782363			6	54 Mbps	(Fake)	AP		21	-78	-100	22
00115020DB55	MSHOME		1	54 Mbps	(Fake)	AP	WEP	12	-85	-100	15
00152B783083			1	54 Mbps	(Fake)	AP		16	-84	-100	16
001195D62757	default		6	54 Mbps	(Fake)	AP		25	-74	-100	26
00152B782360	hhonors		6	54 Mbps	(Fake)	AP		15	-77	-100	23
00152B782362			6	54 Mbps	(Fake)	AP	WEP	18	-79	-100	21
00152B782A01			1	54 Mbps	(Fake)	AP	WEP	16	-81	-100	19
001195D6264B	default		6	54 Mbps	(Fake)	AP		26	-73	-100	27
001195D627AD	default		6	54 Mbps	(Fake)	AP		35	-49	-100	51
001195D626C7	default		6	54 Mbps	(Fake)	AP		34	-49	-100	51
00152B782361			6	54 Mbps	(Fake)	AP	WEP	19	-81	-100	19
001217C616F4	linksys		6	11 Mbps	(Fake)	AP		28	-62	-100	38
001195D625F2	default		6	54 Mbps	(Fake)	AP		20	-79	-100	21
00152B783082			1	54 Mbps	(Fake)	AP	WEP	17	-81	-100	19
00152B782A02			1	54 Mbps	(Fake)	AP	WEP	16	-82	-100	18
001195D625E7	default		6	54 Mbps	(Fake)	AP		29	-50	-100	50
000C416E773D	AWLAN01		3	54 Mbps	Linksys	AP	WEP	17	-82	-100	18
001195D627EF	default		6	54 Mbps	(Fake)	AP		28	-54	-100	46
001195D62616	default		6	54 Mbps	(Fake)	AP		48	-49	-100	51
001195D62701	default		6*	54 Mbps	(Fake)	AP		42	-44	-100	56

Ready 25 APs active GPS: Disabled

A little research on the Internet can help you choose a good wireless network card. The Seattle Wireless (www.seattlewireless.net) Web site has many excellent articles that can help you make an informed decision. You will need to choose which wireless card to get, whether or not you want to use an external antenna (versus the standard built-in antennas) and if so, whether you want to use an omnidirectional antenna or a directional one. If you are going to be trying to triangulate to find the devices in question, a directional antenna will make the job much easier. Also, remember when making your hardware selection that signal loss is the enemy, and for each connector between the antenna and the wireless device you are sacrificing some small amount of signal clarity. Because of this, having the proper connector on your antenna is preferable to using a “pigtail” as an adapter cable between the antenna and the wireless device.



When trying to triangulate the location of a given wireless device, bear in mind that wireless signals can be reflected off nearby objects, such as buildings. So you might get a stronger signal from a wireless access point in the room next door from the direction of the building across the street. This could happen if the walls between you and the wireless device were very well insulated and the building across the street was particularly well suited to reflecting the wireless signal.

Many wireless card manufacturers also offer a utility to monitor the signal strength of an access point. These will often show a graph of the signal strength and or signal quality. Their utilities generally require you to be associated with the access point in question, however, so their use as a war walking utility will be limited. NetStumbler packs a lot of information on its results page. The general quality of the signal is indicated by the color of the circle on the left (green is good, yellow not so good). The circles with a lock symbol indicate that the wireless device is using some form of encryption. The type of encryption is shown in the Encryption column. If NetStumbler detects an access point (green or yellow circle) but is no longer receiving any signal from the device, the circle will change to gray. The device could be gray due to ambient conditions such as whether causing a weak signal to no longer be detected, or the device could have been simply turned off.

The tree view (left-hand pane) of NetStumbler includes some handy ways to sort your results. The Channels entry can be expanded to list all the channels with signals on them. Each channel can be expanded again to see what devices are using those channels. This can be useful if you are getting interference from nearby devices on the same channel you are using. This will let you quickly see what other devices are using the same channel. The Filters entry also contains some handy ways to filter the results. One of the more useful filters is the Encryption Off, which, simply enough, shows a listing of all the devices that are *not* using encryption.

One feature you might find yourself searching for is a way to tell NetStumbler to connect to a given wireless device. Don't look too hard because NetStumbler does not include this feature. To connect to any of the

access points you discover, you will need to use the operating system utilities or another software program. Boingo is one such program for Windows (www.boingo.com). While it is not at all full featured, and it is not very good at displaying accurate signal strength, you *can* highlight an AP and click Connect and it will attempt to connect for you. Another tool of note is Aircsnort, which can be used to passively collect encrypted packets and eventually decrypt the keys used for wireless communications.



WARNING

Remember to use sound judgment when dealing with wireless devices. After connecting, the owner of the device could be sniffing all of your traffic, looking for vulnerabilities to exploit, or blatantly attacking your system. Once connected, you should not make any additional connections through that wireless device or you could expose your credentials to whoever controls the access point. Even initiating an encrypted connection to a trusted device *through* the device would be ill advised because they could be acting as a man in the middle and intercept your credentials. None of these warnings even touch on the potential legal ramifications of using another's wireless bandwidth. The legality of such activities may vary from one locality to another. For this reason you should seek the legal guidance of your employer before connecting to an unknown wireless device.

Documentation

Documentation is frequently one of the most overlooked aspects of network engineering and design. Most people don't like to generate network diagrams and related documents. Many IT staff consider documentation a poor use of their time and would rather spend it doing "real" work. The fact is there are a lot of reasons why you need to have this documentation, and it is important that the documentation remain accurate and up to date. The types of documentation that is important will vary based on your specific business needs, but the following list represents some of the most important types of documentation from a security perspective.

- Network Topology Maps
- Access Request Forms
- Business Continuity and Disaster Recovery Plans
- IT Security Policies/Standards/Procedures

An additional consideration that applies to *all* your critical documentation is one of availability. If the network is unavailable and all your documentation is stored on a file server, you're going to have a hard time accessing the network documentation that you need to help repair the network. Many times individuals fail to account for this during an emergency and discover they cannot access their critical documentation. This could include not being able to access your business continuity plan, network diagrams, and other critical documents. Typically, copies of the documentation should be printed out and stored in a safe location offsite, possibly at the same location where you store your offsite data backups.

Network Topology Maps

Most people are probably familiar with network topology maps or network diagrams. The idea is to show a graphical representation of how the various network components are connected. Oftentimes this type of documentation is generated when the initial network is installed but it is not maintained like it should be. An *inaccurate* network map can actually cause more problems than not having one at all, because someone may assume things are configured one way, when in fact they are configured differently. Accurate network diagrams are critical. Their real value is apparent when there are problems and you need to troubleshoot the network. When you don't know how things are put together, any problem solving has to be preceded with an information-gathering exercise that only adds unneeded delays.

Or maybe the network is small enough that you know every device that is connected to it like you know the back of your hand, and you don't need a diagram. This might work adequately most of the time but if you ever have to bring in outside help, they probably *don't* know the network as well as you do, and now you have to pay for the consultant's time just to learn how things work, before they can even begin to do the work you actually hired

them to do. In this situation the lack of accurate documentation is costing you or your company real dollars, not only to pay for the outside help to learn the lay of the land, but possibly in lost revenue while the solution is delayed. These costs can get outrageous quickly if you're not careful and these are the sort of things management will take notice of.

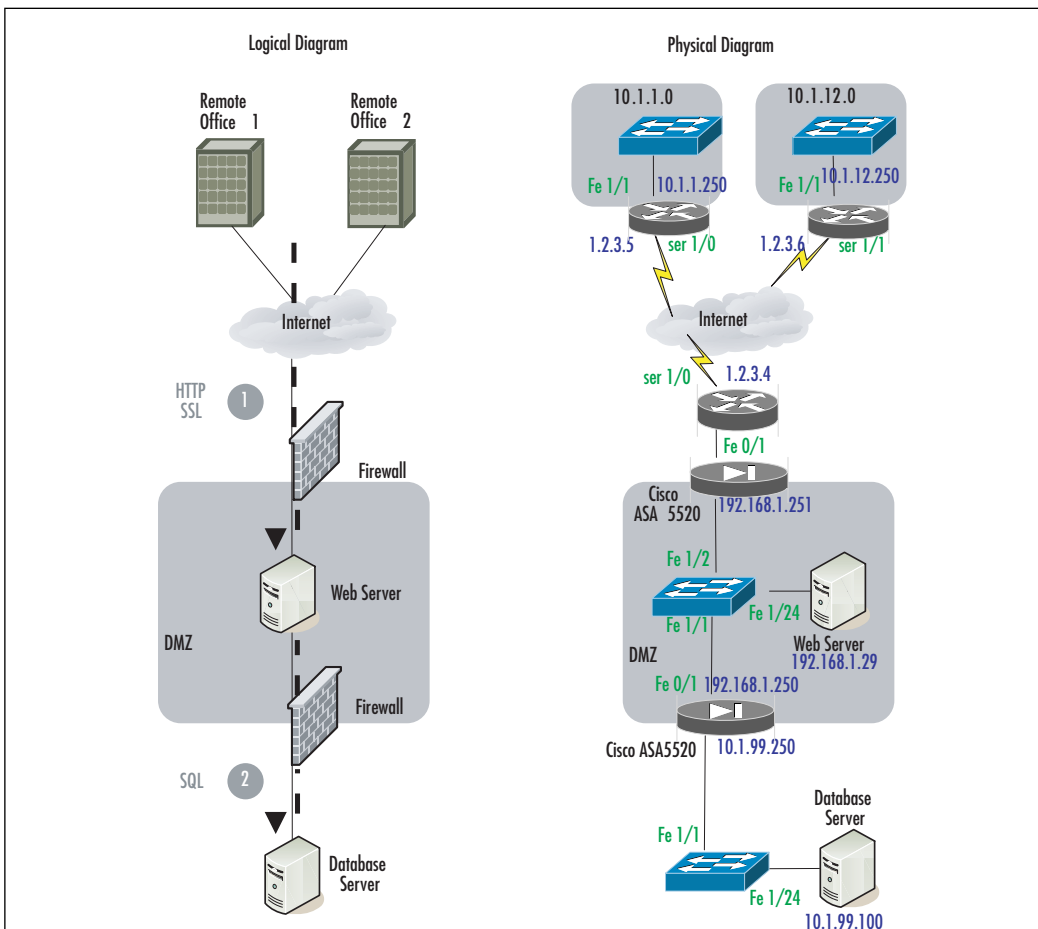
Yet another situation where quality documentation can really be valuable is for an audit. There are the obvious types of audits, Sarbanes-Oxley, SAS70, and related types of business audits. You might not be impacted by these types of regulations and requirements. There are other audit-like scenarios that may affect you. If you are looking to partner with another business entity in such as fashion that it will require network connectivity between the two entities, the other business partner will likely require documentation related to your network infrastructure. They will want to review it to make sure it is a secure configuration. If you cannot provide them with the documentation they requested it could impede the business venture, again causing a loss of revenue.

Okay, so I've harped on the value of good network maps enough. The next question is what exactly constitutes "good" network maps? At its most basic form, simply having all the important data in one place and it being accurate is all that is required. Beyond that, there are other characteristics that are nice to have, such as consistency. A consistent look and feel will go a long way for being able to quickly look at the diagram and understand the information it contains. This could mean a consistent set of icons or symbols, and consistent placement of key information, like who the document owner is and version information. While different people generating the diagrams will have a stylistic impact on their work, if these get too disparate you can end up having a lot of difficulty sorting out one document from another.

You should also consider developing a stance on when it is appropriate to use logical diagrams or physical diagrams. Logical diagrams tend to be more high level and show the overall data flow and devices' general connectivity, while a physical diagram typically includes specifics on cables, ports, and so on. Each type of diagram has its place. A physical diagram is generally of more use when it comes time to troubleshoot a connectivity issue, while a logical diagram often is clearer for nontechnical staff, such as project managers and upper management. To some degree it may just come down to personal preferences; neither type is "wrong," and either or both types may be appropriate

depending on your needs. When it comes to the aesthetics of documentation consistency will likely prove to be an asset. An example of the same diagram in both a logical and physical view is shown in Figure 6.12.

Figure 6.12 Logical and Physical Sample Diagram



Access Request Forms

Another key piece of documentation is the access request form. You will want documentation to demonstrate that a user formally requested access to the network, or a particular network resource, such as a server. This documentation will also serve as a record for who approved the recourse, and for how long. This type of documentation will most often be useful for audits, to

demonstrate which systems and users have approved access to use the network. This category can also include the signed IT security policy (which may be a requirement to approve network access). Either of these could be important if HR needs to follow up on a matter of network usage policy breach. As with the network diagrams, these types of documents could be useful for demonstrating best practices and instilling confidence in potential business partners.

Business Continuity and Disaster Recovery Plans

While not purely a network security document, there are many security considerations surrounding business continuity (BC) and disaster recovery (DR) plans. For one, they will typically contain a log of highly sensitive information in the plans themselves. For this reason, access to these documents should be limited to only those personnel who require access. This documentation will also serve as your first guide to walk through the processes that are outlined, and the infrastructure that is in place, and to look for any security risks. Oftentimes people neglect to secure their DR servers or leave backup tapes containing sensitive information lying around without securing them. In the end, this documentation will hopefully never be useful or needed, but if it is, these are the documents that can make or break a company after a disaster occurs.

IT Security Policies/Standards/Procedures

Because this subject is the cause of much confusion, it's worth summarizing what each of these types of documents should contain.

- **Policies** Policies are broad statements that are general in nature. These documents should not change often. For example, a policy statement could be “data classified as confidential or higher must be encrypted when traversing an untrusted network.” These documents rarely contain sensitive information, and one company's policies will often look very much like another's.
- **Standards** These specify what method should be used to conform to policy. They are more specific than policies. An example of a standard would be “acceptable encryption protocols are 3DES, AES(128),

and AES(256).” The information in standards may be useful to a hacker, such as what encryption you are using, but this information is typically of marginal value.

- **Procedures** Procedures are the most detailed documents. A procedure outlines exactly how to perform a given activity. These are very specific and include exact instructions such as “click here” or “run this program using these options.” Because of the level of detail, procedures often make use of numbered steps and include specifics such as IP addresses and possible access accounts and passwords. While not every process will have procedures written for it, these documents often contain highly sensitive information and should be safeguarded appropriately.

Because some of the documentation in this category can contain sensitive information they should be handled with care. Processes need to be in place to ensure the information is available, and that the confidentiality of the data is maintained. The integrity of the data is sometimes overlooked but is of equal importance. Only authorized individuals should have access to modify this documentation.

Vulnerability Scanning

After locating all the hosts on your network, and hopefully removing or performing remediation on the unauthorized ones, you should determine the security status of all your systems. One of the most efficient ways of doing this is with an automated vulnerability scanner. These types of scanners typically work using varying levels of *invasiveness*. At the safest end of the spectrum, the devices only look for settings that *might* indicate a vulnerability, but they do not actually exploit the vulnerability. This approach can result in some false positives, but it is also the safest type of scanning because it carries the lowest risk of causing a service disruption on the target machine. At the opposite side of the spectrum, the scanner can actually attempt to exploit the vulnerability. Because many of the vulnerabilities are expressly designed to disrupt service, this type of scanning obviously carries a high risk along with it. It does, however, result in very few false positives and provide a very accurate indicator of the overall security of the system in question.

It's worth pointing out that a vulnerability scanner is just that, a scanner with no real intelligence. Some unscrupulous security companies will run the same scanner you can run for free, print the results, and present them to their customers along with a bill. The vulnerability scanner has to depend on the human user to configure it intelligently. And even when something *is* found, it is a human that must make a judgment call as to whether or not that item is truly a risk in your current environment. In some cases, further investigation will be required to determine if the findings are valid and if they represent a true risk. What follows is a discussion of Nessus, one of the best free vulnerability scanners available today.

Nessus

Nessus has been around for a long time, since 1998, in fact. It is available for Linux, FreeBSD, Solaris, MAX OS X, and Windows (2000, XP, and 2003). The Windows product is currently listed as beta, but it ran fine with no issues for me. You can download it from www.nessus.org/download/. While free, the most current version (Nessus 3) is no longer open source. The Nessus system comprises two components: a server and a client. The server process does the actual scanning, while the client is used to configure and run scans and to view the results of a scan. Nessus is a very feature-rich application, which can perform more than 10,000 different types of checks via downloadable plug-ins. The licensing is relatively generous, but there are some circumstances whereby you must purchase a license. In short, you can always scan your own personal systems but for scanning third-party networks some additional licensing will be needed. There are also license options for installing Nessus on an appliance to be provided to customers and for providing Nessus as an OEM product. For full details on the licensing of Nessus, refer to the licensing FAQ located here: <http://nessus.org/plugins/index.php?view=faq>.

You should periodically scan your hosts for vulnerabilities according to the requirements of your IT security policy. You should also perform a vulnerability scan any time significant changes are made to your network or hosts because this could inadvertently create a security risk, either due to human error or due to an interaction between the changes and the existing security controls. A significant change could include adding a new feature like enabling terminal services, performing an upgrade, or installing a new service

pack. Basically, if significant changes have been made, you want to ensure that those changes haven't created a security vulnerability.

Running Nessus on Windows

The installation of Nessus is straightforward. You will need to select the installation target directory and accept the license agreement. For an excellent reference on Nessus see *Nessus Network Auditing* (Syngress Publishing, 2004).

After you have successfully installed Nessus, simply click the **Tenable Nessus** icon, and you will see the startup screen, as shown in Figure 6.13.

Figure 6.13 Windows Nessus



To begin, simply click **Start Scan Task** and then enter the target host to scan. You can enter a single IP address, a list of IP addresses separated by commas, a hostname, or even a network range. After you enter the target(s), click **Next**. The next screen enables you to select the plug-ins you want to

use. The default will be to run all plug-ins except the “dangerous” ones. The dangerous plug-ins are those that are more invasive and that run a higher risk of causing a service disruption. If you want more granular control over which specific plug-ins are executed, click **Manage Policies** (refer back to Figure 6.13). When you are satisfied with your plug-in selection, click **Next**. The next screen enables you to choose where to scan *from*. You can use the system you are on as a client, and instruct a different machine to do the actual scanning, or you can use the system you are on as both the client and the server, in which case it will perform the scan from the same system you configure the scan. After you have entered the information for the server to use, or left the default of **localhost**, click **Scan Now**.

Running all the plug-ins will take a little time to complete. After the scan has completed, your Web browser will open and display the results of the scan. A set of results from a default (that is, no dangerous plug-ins) scan of a Windows XP Professional system is shown in Figure 6.14.

If you want more control of how the scan is performed, you can select **Manage Policies** in the left pane, and then click **Add a new policy** in the right pane. You will need to choose a name for the policy and click **OK**. After this is done, the policy will appear in the list. You can choose either **Edit Settings** or **Edit Plugins**. The settings option enables you to configure various parameters about the scan, such as whether or not the dangerous plug-ins should be used, credentials to use during the scan, ping options, etc. The **Edit Plugins** option is simply that, it enables you to pick and choose which plug-ins you want to use. The plug-in selection window has two panes, the one on the left is for a plug-in *family*, such as all plug-ins related to FTP, while the pane on the right lists the individual plug-ins within the selected family, as shown in Figure 6.15.

Figure 6.14 Windows Nessus Results

Tenable Nessus Security Report

Start Time: Mon Sep 04 16:57:50 2006 **Finish Time:** Mon Sep 04 17:00:19 2006

localhost

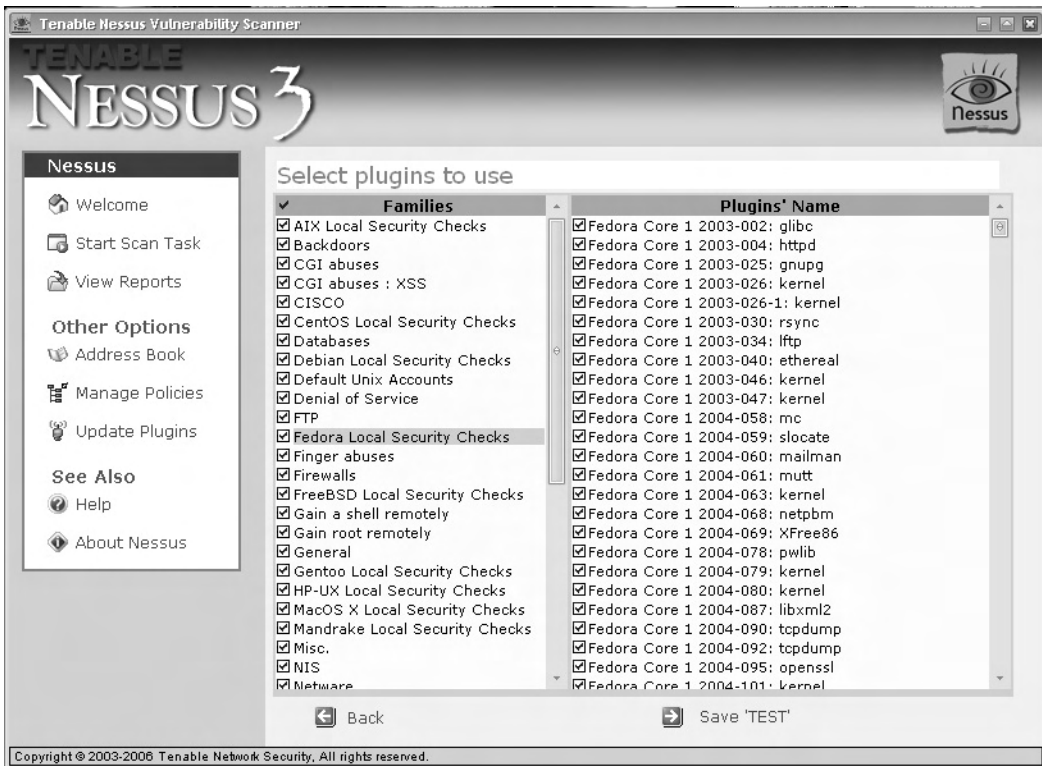
5 Open Ports, 18 Notes, 2 Warnings, 2 Holes.

127.0.0.1 [\[Return to top\]](#)

talarian-tcp (5101/tcp)	<ul style="list-style-type: none"> Port is open Plugin ID : 11219 Yahoo Messenger is running on this machine and listening on this port. Yahoo Messenger allows a user to chat and share files with remote entities. <p>Solution: Ensure that the service is required within your environment.</p> <p>Risk Factor : Low Plugin ID : 11993</p>
ssh (22/tcp)	<ul style="list-style-type: none"> Port is open Plugin ID : 11219 An ssh server is running on this port Plugin ID : 10330 Remote SSH version : SSH-2.0-OpenSSH_3.8.1p1 Plugin ID : 10267 The remote SSH daemon supports the following versions of the SSH protocol : <ul style="list-style-type: none"> . 1.99 . 2.0 SSHv2 host key fingerprint : 97:2d:c8:2b:18:4f:29:0a:4b:91:54:25:c2:5e:3d:1d Plugin ID : 10881
epmap (135/tcp)	<ul style="list-style-type: none"> Port is open Plugin ID : 11219
microsoft-ds (445/tcp)	<ul style="list-style-type: none"> ✘ Synopsis : Arbitrary code can be executed on the remote host due to a flaw in the 'server' service. Description :

Done My Computer

Figure 6.15 Nessus Plug-Ins (Windows)



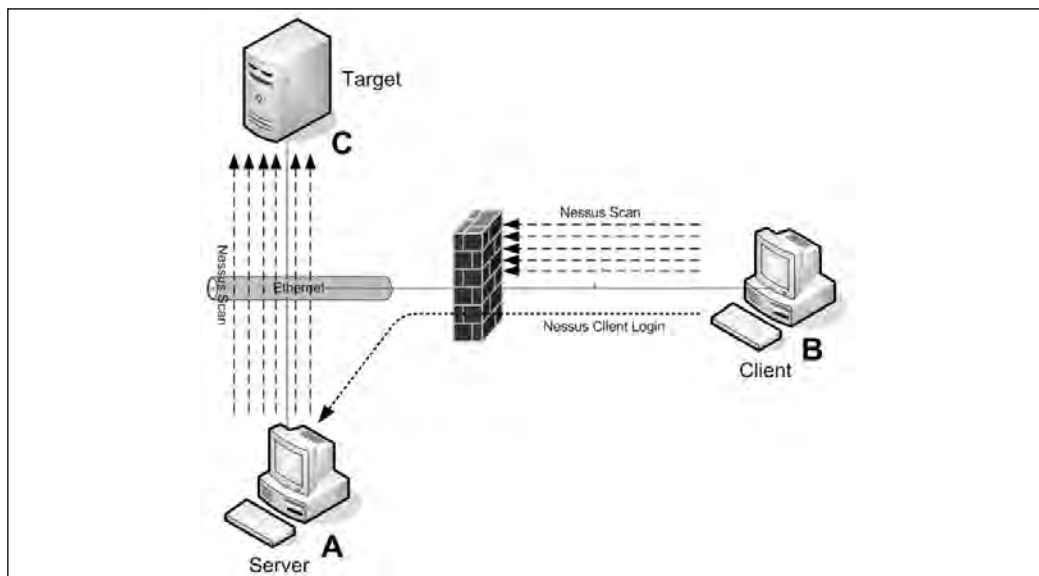
If you know you are using only Windows hosts, you could de-select entire families of plug-ins such as the highlighted Fedora Local Security Checks. Running these plug-ins when you know they will not be applicable to the system(s) being scanned will only increase your scan time, and in the case of the “dangerous” plug-ins, possibly increase the chances of causing a service disruption.

Running Nessus on Linux

Nessus operates a little differently on Linux than it does on Windows. On a Windows system, when you start Nessus, by default the client (which configures the scan parameters and views the scan results) is the same system that is doing the actual scan (the Nessus server). NessusWX is the Windows client that will enable you to connect to a remote Nessus server. In Linux, this does not have to be the case. The Nessus *server* process, *nessusd*, can be running on

machine A, while you use the Nessus *client* on machine B to configure a scan against target host C. Machine A will then perform the scan and send the results to machine B for viewing. Because of this, the installation files for the client and server will need to be installed individually. This configuration provides additional flexibility in case you need to perform the scan from a different system than the client. One scenario where this could be useful is illustrated in Figure 6.16. In this case, there is a firewall between your Nessus client system and the target system, but there is no firewall between the Nessus server and the target you want to scan. If you were to use system B as the client and server, many of the checks that need to be performed would be blocked by the firewall, producing inaccurate results at best. With host B acting as a client only, and host A acting as the Nessus server, the firewall only needs to pass the Nessus client session (default TCP port 1241) and then the server could perform the checks on system C without the firewall interfering.

Figure 6.16 Nessus Client-Server Operation



After the install is complete, you must create a user for the Nessus server. This is done by executing `/opt/nessus/sbin/nessus-add-first-user`. Enter a username and then select the authentication method (the default is to use a password). If you selected password, enter the password twice for confirmation.

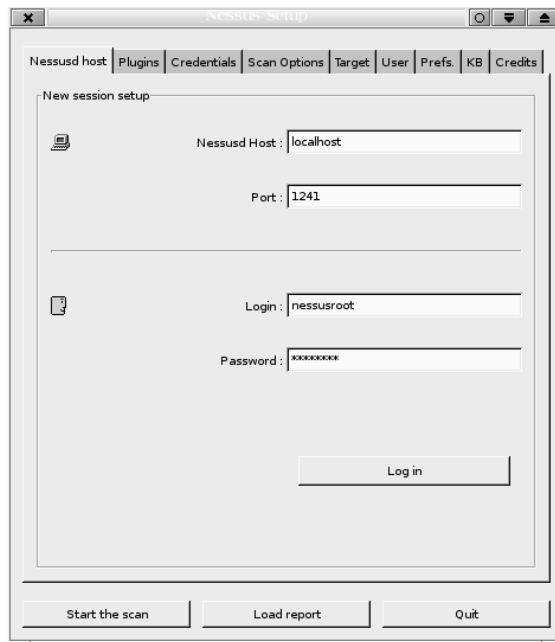
You will then be asked to enter any user rules. The user rules serve to restrict what hosts this user can scan with Nessus. To permit a given user to scan any hosts in 192.168.1.0 and the Nessus client system they are using but nothing else, you would enter the following rules:

```
accept 192.168.1.0/24.
accept client_ip
default deny
```

See the man page for **nessus-adduser** for more detailed examples. Simply type **CTRL+D** to exit if you do not wish to apply any rules at this time, and then press **y** to verify your choices. After this is completed, enter **/opt/nessus/sbin/nessusd -D** to start the Nessus server daemon.

The Nessus client interface looks significantly different than it does on a Windows system, as shown in Figure 6.17. You will use the Nessus client to log in to the Nessus server even if both the client and the server happen to be on the same machine. Enter the Nessus user and password and click **Log in**. In the figure, the client and the server are the same system; this is why the **Nessusd Host** field contains the value **localhost**. You could also easily enter a remote hostname to use the Nessus server on a different system.

Figure 6.17 Linux Nessus Login Screen



After you are logged in, you have access to several tabs. Select the Plugins tab to choose which types of checks to perform. Similar to the Windows interface, the top pane lists the plug-in family, and the bottom pane lists the individual plug-ins. Enable a given plug-in or plug-in family by placing a check next to them. At the time of this writing there were a total of more than 11,000 plug-ins as part of the default Nessus package. Use the Target tab to specify which machine(s) to scan. The Credentials tab is used to provide login information for SSH, SMB (Windows), and Kerberos. In some cases, Nessus will need to authenticate in order to perform some of the checks. The **Prefs** tab enables you to configure various options such as SNMP community strings, HTTP logins, scan verbosity, and many more variables. When you are satisfied with your choices, simply click the **Start the scan** button.

NOTE

Several types of checks will not be fully tested by default. These are types of scans that run a higher than normal risk of causing an undesirable response from the target host. This behavior is the same as the Windows Nessus, in that the more “dangerous” plug-ins are not used by default. To enable these plug-ins, you must remove the check next to **Safe Checks** on the **Scan Options** tab.

After the scan is completed, a report window should open with the results. The report window opens in a Nessus window, not in your Internet browser. The interface enables you to go from pane to pane and drill down into your results. By selecting the Subnet, you are presented with a list of hosts for that subnet in the Host pane. When you select a host, the Port pane populates and enables you to drill down on the results of a specific port. When you select a specific port you can then choose which results for that port you wish to see. The specific nature of the vulnerability will be explained in the largest pane in the lower right, as shown in Figure 6.18.

Figure 6.18 Linux Nessus Scan Results

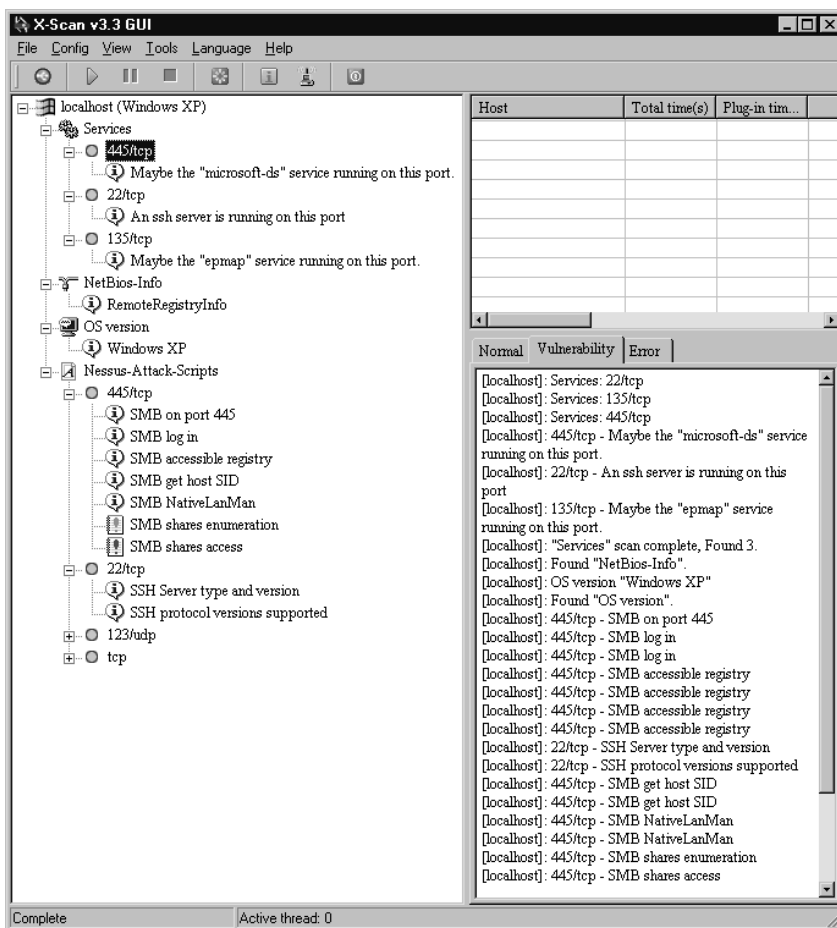
After having looked at the large number of plug-ins available, you can see that Nessus is a very powerful tool for determining what vulnerabilities your systems might have. With the large number of supported operating systems and ability to check for a wide range of vulnerabilities, Nessus is a valuable tool for testing the security of your systems. As is often the case, Nessus will run faster from a Linux system than it will from a Microsoft system, but if you don't need to scan many systems, the difference is probably not significant. The primary advantage to subscribing to Nessus is that you will receive updated plug-ins as soon as they are released. With the free “registered” version you are able to get the updated plug-ins seven days after the subscription users. If you choose not to register your Nessus software, you can get the updated plug-ins only with each new release of Nessus, so with the free version, there is some delay in getting the newest plug-ins. As a paying subscriber you can also configure scanning policies that Nessus will check; these policies can include a wide array of system settings. Even with these limitations, however, it is an excellent free vulnerability scanning tool.

X-Scan

Nessus isn't the only game in town when it comes to vulnerability scanning. There are many offerings but most of them are commercial products. Another free vulnerability scanner is X-Scan from www.xfocus.org/programs/

200507/18.html. X-Scan is a Windows-only scanner that supports a couple of interesting features such as OS detection and weak password checking. Using X-Scan does not require an installation, all you have to do is decompress the files to a location of your choice and run the executable. The main window (with some scan results already populated) is shown in Figure 6.19.

Figure 6.19 X-Scan Results



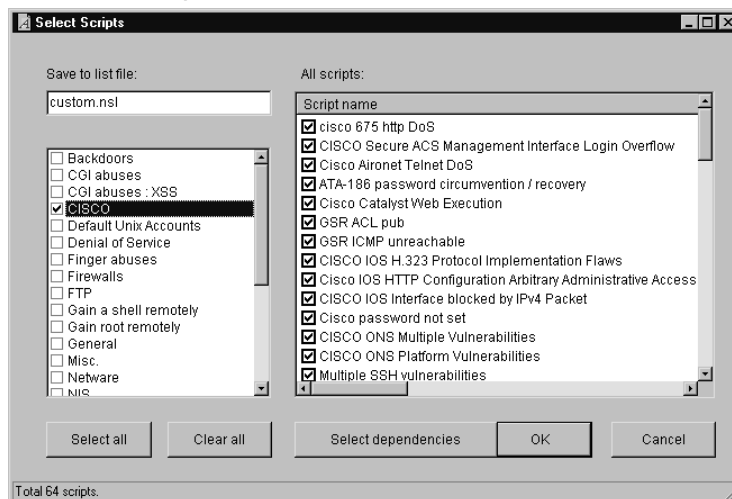
To perform a scan of the local host, you can simply click the green “play” triangle, or navigate to File | Start. If you want to scan a different host, navigate to Config | Scan Parameter. This will open the Config window. The left pane contains a hierarchy of configuration options. Select **Scan range** in the

left pane and enter the target host(s) in the right pane in the **IP address range:** field. If you want to configure some of the checks that are specific to X-Scan, open the Config window and navigate to Global Options | Modules in the left pane. The center pane will display a list of modules, while the right pane explains what the module does. These modules include some OS fingerprinting and a large number of weak password checks.

By selecting **Global options | Report** in the left pane, you can choose what format to use for the scan report. The default is HTML, but you can also choose XMS or text file. If you select **Global options | Others**, you will find a couple of useful options. The default is to *Skip host when failed to get response*. By selecting Scan always, X-Scan will perform the scan even if the system is not responsive to a ping. This can be useful in cases where you know the target host will not respond to a ping but the system is up. You can also see that the default is for the OS fingerprinting to be performed by Nmap. The default list of ports can be edited by selecting **Plug-in options | Port**.

If you want to choose specific plug-ins to apply, navigate to Plug-in options | NASL. If the term NASL sounds familiar that's because it stands for *Nessus Attack Scripting Language*. X-Scan uses the same plug-ins that Nessus uses. The default is to select all plug-ins excluding the “destructive” ones, which in Nessus are called “dangerous.” If you uncheck the Select all box, you can click Select at the top. The Select Scripts window enables you to choose which scripts to run in a familiar format, as shown in Figure 6.20.

Figure 6.20 X-Scan Script Selection



The left pane lists broad categories for the scripts, which is the same as the Nessus plug-in *family* lists. The right pane lists the individual scripts. After making your selection be sure to click **Select dependencies** so that any scripts that are needed to support the ones you selected are enabled. After you are satisfied with your selection, click **OK**. The **save to list file** field enables you to save your lists for future use. It might be useful to save a list that applies only to Windows hosts because many of these checks are for other types of systems. After making all your configuration changes, click **OK** and then run the scan. By default, X-scan will open your browser to view the report results when the scan is completed. A sample HTML report is shown in Figure 6.21.

Bear in mind you will need the WinPcap drivers installed in order to use X-Scan, though like many such utilities X-Scan will install WinPcap if you don't already have it installed. When it comes to the actual checks being performed, X-Scan uses the same Nessus plug-ins, but X-Scan also has the capability to perform some additional checks and OS fingerprinting. Like Nessus, X-Scan also supports command-line operations, which could be a plus depending on your environment. On the down side, X-Scan does not support the client-server architecture that Nessus does. If that type of functionality is needed, you could use some type of remote access functionality to run the scans from a system more appropriately located, but this would require third-party software. If this capability is needed, X-Scan may not be the best tool and Nessus might be a better fit.

Figure 6.21 X-Scan HTML Report

X-Scan Report

This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scanning time
9/19/2006 3:26:13 PM - 9/19/2006 3:29:03 PM

Scan Result

Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0
Number of security notes found	17

Host List

Host(s)	Possible Issue
192.168.1.104	Security notes found

Host Summary - OS: Linux 2.4.X|2.5.X|2.6.X; OS details: Linux 2.4.0 - 2.5.20, Linux 2.4.18 - 2.6.7; PORT/TCP: 22, 111
[return to top]

Analysis of Host: 192.168.1.104

Address of Host	Port/Service	Issue regarding Port
192.168.1.104	sunrpc (111/tcp)	Security notes found
192.168.1.104	ssh (22/tcp)	Security notes found
192.168.1.104	portmapper (111/tcp)	Security notes found
192.168.1.104	RPC/status (34807/tcp)	Security notes found
192.168.1.104	portmapper (111/udp)	Security notes found
192.168.1.104	unknown (32768/udp)	Security notes found
192.168.1.104	RPC/portmapper (111/tcp)	Security notes found
192.168.1.104	RPC/portmapper (111/udp)	Security notes found

Security Issues and Fixes: 192.168.1.104

Type	Port/Service	Security Issues and Fixes
Informational	sunrpc (111/tcp)	Services Maybe the "sunrpc" service running on this port. NESSUS_ID : 10330
Informational	ssh (22/tcp)	Services An ssh server is running on this port.

Microsoft Baseline Security Analyzer

The Microsoft Baseline Security Analyzer is a tool for checking the baseline security of supported Microsoft products. In this instance *baseline* means that

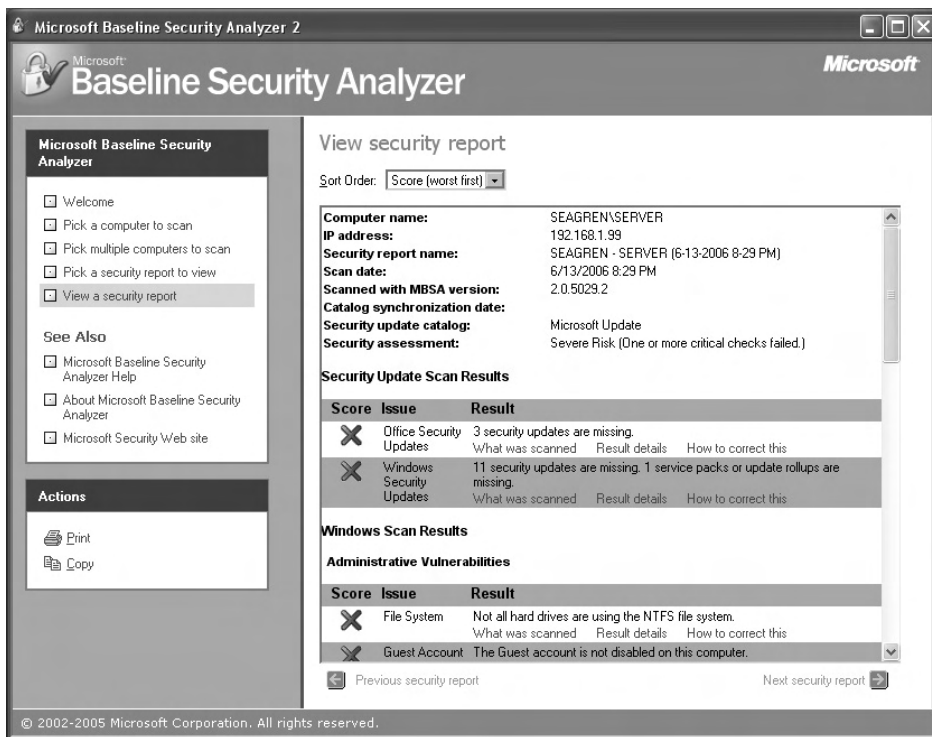
the minimum security patches have been applied (MBSA uses the updater service to check patch levels) and the minimum security settings have been checked. The MSBS is not a general-purpose vulnerability scanner like Nessus. MSBS is instead a way to check your Microsoft hosts for *weak* security settings, not necessarily vulnerabilities. The primary page for MBSA is www.microsoft.com/technet/security/tools/mbsahome.msp. There are different versions of MBSA, each supporting different platforms. MBSA 1.2.1 is for users who have Office 2000 and Exchange 5.0 or 5.5. MBSA 2.0 supports Windows 2000 SP3 or later, Office XP, Exchange 2000, and SQL Server 2000 SP4 or later. A more complete listing of supported products can be found in the article located at <http://support.microsoft.com/?scid=kb;en-us;895660>. The older MSBA 1.2.1 supports only a limited set of software. You can use MBSA 1.2.1 combined with EST (Enterprise Scan Tool) to obtain fairly comprehensive scanning coverage of older legacy applications. The software is relatively small and lightweight, at less than 2 MB. The installation process is simple and quick. The MBSA interface, shown in Figure 6.22, is also very straightforward.

Figure 6.22 Microsoft Baseline Security Analyzer



To scan a single computer, simply click **Scan a computer**, enter a computer name or IP address, and then click **Start Scan**. The scanner will report on any settings or options on the target that are suboptimal from a security standpoint. The results of a sample scan are shown in Figure 6.23. The MBSA has marked a red “X” for any security issues it finds, including in this example not having all the disk partitions formatted as NTFS. NTFS is Microsoft’s file system format that allows for configuration access controls on files and folders. Fat32 is an older Microsoft file system that has no file security.

Figure 6.23 Microsoft Baseline Security Scanner Results



By default, the scan results are stored in %USERPROFILE%\SecurityScans as .mbsa files. MBSA also includes a command-line version, called `mbsacli.exe`. There are several useful options that can be used on the command line. Basic usage would be `mbsacli /target 192.168.1.99`, for example. You could also use `mbsacli /r 192.168.1.1-192.168.1.254` to scan a range of IP addresses. If you omit the target completely, MBSA will default to

the localhost as the target. By default, `mbsacli` this way produces text results directly to the console and creates an `.mbsa` report in the `%USERPROFILE%\Security Scan\` directory. The text output can be redirected to a file, and while the output formatting *is* conducive to parsing the results programmatically, it is *not* a very good format for human viewing. Unfortunately, there isn't much out there to help you manipulate the findings. Microsoft does offer the Microsoft Office Visio 2003 Connector for the Microsoft Baseline Security Analyzer (MBSA) 2.0. This tool enables you to see the results in Visio when clicking on a Visio icon. You can download the Visio connector from www.microsoft.com/technet/security/tools/mbsavisio.mspx.

So let's suppose you wanted to test all your systems regularly using MBSA and report on critical updates that are not installed. You could use the command-line version and the job scheduler for the scheduling part. Then take your *text* output files and the *find* command, such as *find* "Missing" `<MBSAout.txt>`, to list all missing updates. You could further pipe this into *find* a second time to output only the missing critical updates with *find* "Missing" `<MBSAout.txt>` | *find* "Critical". To summarize, MBSA is a very good tool but its most glaring weakness is the lack of a good reporting mechanism. If you have only a few systems to test, however, MBSA can be a very useful tool.

OSSTMM

Let's suppose that you can run vulnerability scanners and perform network discovery, but you want to take your security assessments to the next level. If you begin to think about all the security testing that *could* be done, covering such broad topics as wireless security, physical security, employee education, incident response, and much more, you might feel a bit overwhelmed. The task is so large that simply figuring out where to start could be difficult, and if you do that, there is always the possibility that you might miss something critical. The OSSTMM (Open Source Security Testing Methodology Manual) is exactly what it sounds like. This is a free manual on how to perform a security assessment. It is very detailed (version 2.0 is 120 pages) and can be downloaded from www.isecom.org/osstmm/. Even if you choose not to perform

all the testing that is outlined in the manual, it is an invaluable resource to help guide you on proper testing procedures and practices.

In addition to covering such pretesting tasks as defining scope of the testing and the rules of engagement, it breaks the actual areas to be tested down into sections and subsections called *modules*. Here are summaries of the sections covered.

- **Information Security** This broad category covers such tasks as scouring the Internet for publicly available information that can provide clues about nonpublic information. An example of such indirect disclosure would be a job posting that specifically requires experience with F5's BIGIP products, which would indirectly tell people what type of load balancers the company is using. It also covers the secure handling of confidential data, including personal data.
- **Process Security** This section includes testing verifying your procedures and attempting to gain unauthorized access via an e-mail or phone call, either through improper process (such as not verifying your identity before resetting an account) or while impersonating someone else. This section also includes luring authorized users to an external location (typically a virtual location such as a Web page) whereby their credentials can be compromised or other information gathered.
- **Internet Technology Security** This section focuses more on the underlying technologies, and examines such things as the network, packet loss, routes, and route control, ISP, and bandwidth of the target organization. This section is really where the build of security testing activity takes place. This section also includes performing a network survey and initial investigation via IP scanning and port scanning. Some indirect disclosure issues will also be touched upon here as you learn some things about the underlying network. The handling of confidential information will again be reviewed at the network level concerning encryption protocols and related technologies. This section also does the work of application vulnerability testing, route testing, access control testing, IDS testing, and testing of anti-Trojan

and antivirus systems. Finally, this section includes modules to address password cracking, denial-of-service testing, and a review of the organization's security policy.

- **Communication Security** This section includes testing the PBX and other communications methods such as voice mail, modems, and fax machines.
- **Wireless Security** Because of the complexity and expertise needed to thoroughly test wireless security, wireless has its own section in the OSSTMM. This section includes such esoteric modules as testing electromagnetic radiation (EMR), which can enable a person to read what appears on a CRT monitor from outside the building, based on the EMR that is projected beyond the display screen. It also covers the more mundane testing against the wireless network itself, including both 802.11x and Bluetooth networks. The broad category would even include wireless headsets and the security of the conversations over them, wireless hand scanners (such as in a retail store), RFID devices, and other wireless/cordless devices.
- **Physical Security** This section includes evaluating perimeter security, security monitoring, and access controls methods, such as gates, doors and locks. It also includes alarm response (all types of alarms, including fire, environmental, and a security incident alarm). The geographical location and ramifications thereof are also reviewed in this section.

The end of the document contains multiple templates that can be used for your actual testing, such as a Firewall Analysis Template and a Password Cracking Template. These templates are not procedures but are rather the type of documentation you would include with your testing, detailing exactly what was tested and how. These templates can be valuable for both ensuring that you are documenting your testing adequately and helping ensure that you do not miss any vital steps, because many of the steps have explicit sections of the template to record the specifics.

The business model for the Institute for Security and Open Methodologies (ISECOM) is basically that they provide the OSSTMM for

free. However, it is a peer-reviewed and updated document. As best practices change and the manual is updated, those changes are made available to “gold” and “silver” subscribers before the general public. Typically, the time delay for free access is a few months. Because this is a testing methodology, a few months’ delay probably will not pose any significant problems for those who want the OSSTMM for free.

Summary

Taken as a whole, the tools and utilities covered in this chapter should empower you to locate the systems on your network using a variety of methods. The best of class utilities presented offer a broad spectrum of choices in complexity and features for discovery scanning. After all the systems are located, you can begin testing them with a vulnerability scanner to determine what their current security posture is. This enables you to build a complete and accurate picture of just how secure the systems are. The Microsoft Baseline Security Analyzer takes this one step further and reports on *weak* security settings for Microsoft operating systems, rather than vulnerabilities. All of this collectively gives you the information you need to complete the first step of securing your network, which is information gathering.

Solutions Fast Track

Taking Inventory

- ☑ Taking an inventory of the devices on your network must be repeated regularly to ensure that the inventory remains accurate.
- ☑ Nmap has more features and option than any other free scanner. Familiarize yourself with not only the options you need to use, but also the ones you might encounter as a hacker attempts to collect information on your network.
- ☑ Because the various scanners have different strengths and weaknesses, you should familiarize yourself with all of them and choose the appropriate one for the task.
- ☑ Identifying that the wireless devices exist should be simple; it's determining the devices' physical location that is often difficult.
- ☑ To attempt to triangulate and locate the physical devices, you will need a scanner that displays an accurate signal strength and a directional antenna.

Vulnerability Scanning

- ☑ Be cognizant of the invasiveness of the scans you are running and of the risks that the scan poses to the target host(s).
- ☑ Consider the legal ramifications to any wireless activities you pursue and ensure you have adequate backing from your employer.
- ☑ MSBA is for Microsoft products only and reports on weak security settings, all of which do not necessarily represent a security vulnerability per se.

OSSTMM

- ☑ OSSTMM is a manual to guide you through the process of performing a security assessment using the peer-accepted best practices.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: Does a simple port scan pose any risk to my target hosts?

A: It definitely *can*. A simple ping scan to see which systems are alive shouldn't pose any risk, but a more involved port scan *might*. If the target host has significantly more resources than the target system, you could exhaust the resources of the target system and result in an inadvertent denial of service attack. Of even higher risk are some of Nmap's more unusual scanning options. Some of the specialized TCP flag manipulation scans carry a definite risk to the target host. Because the flag combinations can be illegal (according to the TCP specifications) the target host might not be coded well enough to handle them. Granted, in this day and age this shouldn't happen, but these types of scans still carry a risk.

Q: Why are wireless access points such a big security concern? Why should I care if my users want to use someone else's Internet bandwidth instead of mine?

A: There are many reasons. If a user who is connected to your corporate network also connects to an open wireless access point and his machine is attacked and compromised, the attacker has an open backdoor into your corporate network which probably bypasses all your firewalls and security measures. Even if you set aside all the security issues, the user in question now has an outside connection that you cannot easily monitor. You no longer have visibility if that user is trafficking in trade secrets or otherwise transmitting confidential information. When the user is using your company Internet connection, you have the capability to use an IDS, collect traffic statistics, take advantage of a firewall you control, and apply other security policies.

Q: Can I write my own custom "plug-ins" to perform special security checks using NASL?

A: You can. There is a large body of plug-ins already available and odds are good the check you're looking for is already available unless it is very customized. You can search the available plug-ins at <http://nessus.org/plugins/index.php?view=search>. The search results may include plug-ins that are not available yet except for the *direct feed* or *registered feed* customers. You can also create your own plug-ins from scratch and if they might be useful to others you can share them with the Nessus community. X-Scan also enables you to create your own plug-ins and uses the same NASL plug-ins as Nessus.

Network Reporting and Troubleshooting

Solutions in this chapter:

- Reporting on Bandwidth Usage and Other Metrics
- Collecting Data for Analysis
- Understanding SNMP
- Troubleshooting Network Problems

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

It is an unfortunate fact of life that network security is only glamorous to geeks. For everyone else, seeing an IDS purr away, or watching swatch grind through gigabytes of log messages is pretty dull, and more importantly, meaningless. There will inevitably be occasions where you need to demonstrate the state of your network to a less-technical audience. In some cases you need to justify a recent expense, in other cases you may need to provide support for a proposed expense. It is at these times that some useful tools to help turn the bits and bytes into graphs can go a long way. We will discuss several such tools in this chapter, and how they can be useful in your day-to-day activities.

Reporting on Bandwidth Usage and Other Metrics

If you've ever been in a position to request approval to upgrade your Internet bandwidth, one of the first questions that often come up is, "What are we using the bandwidth for now?" You don't want to have to admit you don't have any idea. In these cases, some type of reporting mechanism on network traffic would come in really handy. Or maybe the Internet responsiveness is slow because your Internet connection is being saturated and you want to know what it's being used for. A report based on the protocols and ports being used would do the job nicely. There are administrative uses for traffic statistics, but where does security fit in? Maybe the entire network has come to a crawl and you need to know why... fast. There are a lot of ways to determine the cause, but a nice graph showing that a particular workstation is generating all the traffic could help. If your reports clearly showed a particular workstation is uploading large amounts of data over a file sharing network, there could definitely be security implications. When it comes down to it, there are a number of metrics that could be useful for administering and securing your network.

There are many commercial products to provide various levels of insight into your network data flows. There are also a large number of products, both commercial and free, to collect more-focused pieces of data (such as Web server statistics). The following list provides a brief summary of some of the

best general-purpose free offerings, with additional instructions on how to install and configure the products provided later in the chapter.

- **Multi Router Traffic Grapher (MRTG)** When it comes to generic network statistics using free software, MRTG is one of the most widely used. You can download it for free from <http://oss.oetiker.ch/mrtg/>. It will run on Unix/Linux, Windows, or Netware systems and is incorporated into many third-party applications. It derives its figures and graphs from simple network management protocol (SNMP) information, so you will need to support SNMP on your devices to use MRTG or figure out some other means to get MRTG the data it needs. We will discuss SNMP concepts in more detail in the next section, so feel free to jump ahead if you are not comfortable with SNMP. MRTG uses Perl (www.perl.org) on the back end for the real work, which is freely available and easy to install.
- **MZL & Novatech TrafficStatistic** TrafficStatistic (www.traffic-statistic.com) works a little differently than MRTG does; it gathers its data by sniffing all the network traffic. Much like an IDS, for traffic analyzers that work this way, placement within the network will be crucial to collecting the data you want to see. In a small environment, this should not be too difficult. TrafficStatistic offers only very minimal reporting data, consisting of total throughput (in, out, and combined) and a top-10 talkers (in, out, combined), and top-10 protocols (in, out, combined). If you need anything more than that, you can pay for additional *plug-ins*. Some of the plug-ins are rather affordable. You can download the free version from www.trafficstatistic.com/pages/basemodules.html. TrafficStatistic might be a good option if you want something that's very easy to install and get running and only provides the most basic of reporting data.
- **PRTG Traffic Grapher** This is probably the best free offering available at the time of this writing based on functionality and ease of use. You can download the free version from www.paessler.com/download/prtg. PRTG Traffic Grapher is one of

the move versatile offerings and can extract statistical data from the NetFlow protocol, SNMP, and traffic sniffing. The setup is pretty painless and the graphs are well constructed by default. The limitation for the free version is that you can collect data from only three sensors, which is generous and will probably be plenty for a small environment.

- **ntop** Ntop (www.ntop.org) is a very powerful Web-based utility to analyze network traffic. You can run ntop on FreeBSD, Linux, Solaris, SGI IRIX, AIX, and Microsoft Windows systems. Ntop does not natively include alarm and notification mechanisms; its sole purpose is the collection and reporting of traffic statistics, which it does very well. The level of detail offered by ntop exceeds that of any other utility reviewed here. Ntop is also completely free, with no restrictions or limitations.

Collecting Data for Analysis

As we discuss the various offerings for data analysis, a key consideration is how these products collect their data. The methods that are used will have a significant impact not only for what metrics are available for analysis, but also the analysis host's placement within the network and resource requirements. What follows is a brief explanation of what data-collection methods are most common, along with some of their strengths and weaknesses.

Sniffing data is one of the simplest methods of collecting data. Without any special configuration, sniffing the data means listening to all network traffic as it passes through the segment the host system is connected to. This technique is typically the most robust because in sniffing the traffic, the host has the capability to see every single packet. What is done with all this data is up to the analysis engine, but the focus here is that you are not grabbing select pieces of information, you are collecting all the data, and then sorting through it. This method will be more processor intensive than most other methods, especially if there is a high volume of traffic.

This method also requires precise placement of the host that will be collecting data, because it has no way to see the data unless it passes through the segment the host is on. Because of this, the physical location of the data-

collection system will likely be dictated by the network topology and location of traffic you wish to analyze. Besides resource requirements, the biggest drawback to this method is that it will collect data at the network level, with no regard for product-specific metrics. Although some analysis platforms can attempt to remedy this and perform analysis on some higher-level information contained in the packet, you will not be able to get the same level of upper-layer information as you will with the other methods.

SNMP is a protocol that is designed specifically to accommodate the management of network-enabled devices. Although this management can include making changes, in a data analysis context, SNMP is really only used to retrieve information. When used this way, a network host requests certain information from the SNMP-enabled device, which then sends the desired metrics in response. Alternatively, the SNMP device can be configured to send the metrics as a sort of alarm when they surpass a configured threshold. The information collected is limited in that it is very focused. You can only ask a device for the specific statistics that it supports. While sniffing collects network layer data, SNMP can collect higher-layer, product-specific data that sniffing would not easily be able to gather. An example of product-specific counters is the *currentAnonymousUsers* and *currentNonAnonymousUsers* values from an IIS 6 server. Attempting to build in the logic for a sniffer to track each connection to the IIS server and monitor if that connection used authentication would be very burdensome. Instead, SNMP can provide these metrics directly from the IIS server, which is already tracking these things.

SNMP can also be a chatty protocol in a large environment, contributing to network congestion. In a small environment this may not be an issue, but it's something to be aware of. The primary benefit that SNMP has going for it is that you do not need to place your data collector in the path of the data. You can place the system anywhere and then it will reach out and poll the devices (using a *Get*) for the desired data points. You can also configure an SNMP-enabled device to send the metrics to a collector when they reach a preconfigured threshold (via a TRAP). SNMP and sniffing provide different information, which enables the two to complement each other's capabilities. Both SNMP and sniffing will require forethought and planning to implement mainly due to the fact that they each collect their statistics differently.

NetFlow is a specially designed protocol for collecting network traffic statistics. NetFlow is primarily supported on Cisco devices, but some other manufacturers implement similar technologies, which exhibit varying levels of interoperability. NetFlow is similar in behavior to SNMP traps in that once a NetFlow-enabled device has been configured, it will then send traffic statistics back to the data collector. The difference is that while SNMP targets very specific metrics that must be supported by the SNMP device, NetFlow targets a very small subset of network traffic data. NetFlow gathers information based on source and destination IP address, source and destination port number, the protocol being used, the type of service settings, and the device interface. These metrics lend themselves to gathering data on bandwidth utilization and network top talkers. This may sound like just what the doctor ordered; however, NetFlow is not supported on all devices, particularly the more economical models. This may mean that NetFlow is a less viable option for data collection in a small networking environment. If you do have network devices that can support it, a little research would be advisable to see if you can take advantage of NetFlow data. You can read more about NetFlow from here: www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

NOTE

RMON stands for *remote monitoring*, which is yet another network management protocol. RMON is a standard, described in RFC2819, which uses SNMP for its underlying functionality and an extensive set of new MIB objects for its data collection. Because it uses SNMP, RMON is vendor neutral, and RMON also takes steps to reduce network traffic where possible. While RMON support on enterprise class network analysis devices is good, it is virtually non-existent on free network analysis solutions.

Understanding SNMP

As you can see from the brief summaries above, a lot of network devices rely on SNMP to gather information. This makes good sense, because SNMP is

one of the most widely used architectures for managing systems in a centralized fashion. Some basic vocabulary you should be familiar with includes the *Management Information Base*, or MIB. The MIB is basically a hierarchical tree-like structure, serving as a catalog of settings that can be read or changed on the target system. The MIB consists of some portions that will be the same across all devices that support SNMP, and other portions that can be defined by individual vendors. Any specific object in the MIB can be referred to by its *Object Identifier* or OID, which is a numerical map to find the object in the MIB. As an example, the first highest level object in the MIB could be referred to as 1 and each of 3 objects one level lower would be 1.1, 1.2, and 1.3. This pattern is continued, and due to the large number of objects, a typical OID would be 1.3.6.1.2.1.2.2.1.14.1. In addition to using the numerical form to reference an OID, each subtree also has a corresponding name. Using the previous example, 1.3.6.1 could also be referenced with .iso.org.dod.internet. The full named form is used less often but you may encounter it. A *managed device* can be almost any network device that runs an *agent* that can translate device-specific management information into an SNMP-compatible format. The *Network Management Station* is the device doing the managing, and can be referred to by many names depending on where you are reading.

The basic operations the management station can perform on the MIB objects are *Get* and *Set*. There are some variations such as *GetNext*, or *GetBulk*, but suffice it to say that what it all boils down to is using a *Get* to read a value in the MIB, and using *Set* to set the value. In the case of graphing network throughput all we are doing is a *Get* for relevant MIB objects. An example would be .1.3.6.1.2.1.2.2.1.14, which is interface inbound errors. Another key feature of SNMP management is the *Trap*. While the *Get* or *Set* operations are initiated by the management station, acting as the client to the managed device, the *Trap* is initiated by the managed device. An SNMP trap is basically an alert of some preconfigured condition, much like the notifications available in PRTG Traffic Grapher. Traps are sent from the managed device to the management station that was specified as a sort of alarm.

If all this seems a bit confusing, don't be alarmed. If you follow the examples below, a little hands on should help clear things up. There are also several free tools to browse a device MIB. Using one of these will probably be useful

for understanding what the OIDs mean and how they are used. I would suggest using *GetIf* from www.wtcs.org/snmp4tpc/. It is a free MIB browser that is pretty easy to use. Once you get it installed, simply enter an IP address or host name in the **Host name** field, ensure that you have a read community string entered, and then click **Start**. I would recommend only entering a read community string to prevent you from accidentally being able to change any settings on the target device. If *GetIf* populates the various fields, then it is connected via SNMP. Click the **MBrowser** tab and look around. By expanding the plus symbols next to **iso | org | dod | internet**, you can browse to the desired OID. After you select the desired OID, click **Start** (to send a *Get*) and query the value. Try to locate 1.3.6.1.2.1.2.1.5 to see the network interface speed, for example.

One final consideration concerning SNMP is that it is considered a “chatty” protocol. This is due to the process of querying multiple OIDs and receiving the responses. If you have a large SNMP infrastructure, and you are reading or writing a lot of SNMP MIB objects, the network traffic that can be generated can be significant. In a small environment, the SNMP traffic should be minimal, but it is a consideration to keep in mind.

Tools & Traps...

SNMP Security

Be aware that SNMP has been around for a long time and as such there are three major versions. SNMP version one and version two have some considerable security flaws. The foremost is that they send their data unencrypted, which could include a whole host of data you would rather not be viewable by just anyone. A second consideration is that the limited authentication capabilities rely on a community string only to determine not only who can read the SNMP data, but who can set the SNMP values, effectively granting access to configure the SNMP-enabled device.

SNMP version three mitigates many of these issues; however, it is not widely supported at this time. While SNMP v3 is often supported on the more prominent enterprise class platforms, support using free tools is practically non-existent. Given these facts, the best way to secure SNMP

Continued

traffic is with a combination of access lists (where applicable) and some form of encryption, such as IPsec.

Configuring Multi Router Traffic Grapher

To get started using Multi Router Traffic Grapher (MRTG) you will need to download the appropriate version for your operating system from <http://oss.oetiker.ch/mrtg/>. Another version, which is packaged with a few useful SNMP tools, is available from www.openextra.co.uk/products/mrtg-extra.php. MRTG also requires Perl to function. Perl is a versatile scripting language that is in wide use. Perl is very rich in features and has modules to accomplish a variety of useful tasks. You can download ActivePerl from www.activestate.com/store/freedownload.aspx?prdGuid=81fbce82-6bd5-49bc-a915-08d58c2648ca. After you have both of these downloaded, follow these steps to configure them.

1. Run the installation file for Perl.
2. Choose the target installation directory.
3. When prompted, allow the Installation Wizard to add Perl to the PATH environment variable and create the Perl file extension association. This way your Perl scripts can be executed without having to explicitly provide the full path to the Perl executable.
4. Uncompress the MRTG Zip file to a directory of your choosing.
5. From the `\mrtg-2.14.7\bin\` directory, run **perl mrtg**. It won't really do anything yet, because we still need to create the configuration file. This test is just to establish that Perl is in the PATH and can execute mrtg.
6. From the `\mrtg-2.14.7\bin\` directory, enter the following command: **perl cfgmaker <SNMP STRING>@<SNMP DEVICE IP> —global “WorkDir: C:\www\webroot” —output mrtg.cfg**. This will create an initial configuration file. You can always use a different working directory. The working directory is where MRTG will place the HTML files, so it is typically in the directory structure of a Web server. If everything works properly, you will

receive no output on the command line, but an `mrtg.cfg` file will have been generated in the `\bin\` directory.

If you get an error it probably means the community string is incorrect or the SNMP security settings are not enabling MRTG to connect. In either of those cases you can follow these steps to adjust the SNMP security settings on Windows XP.

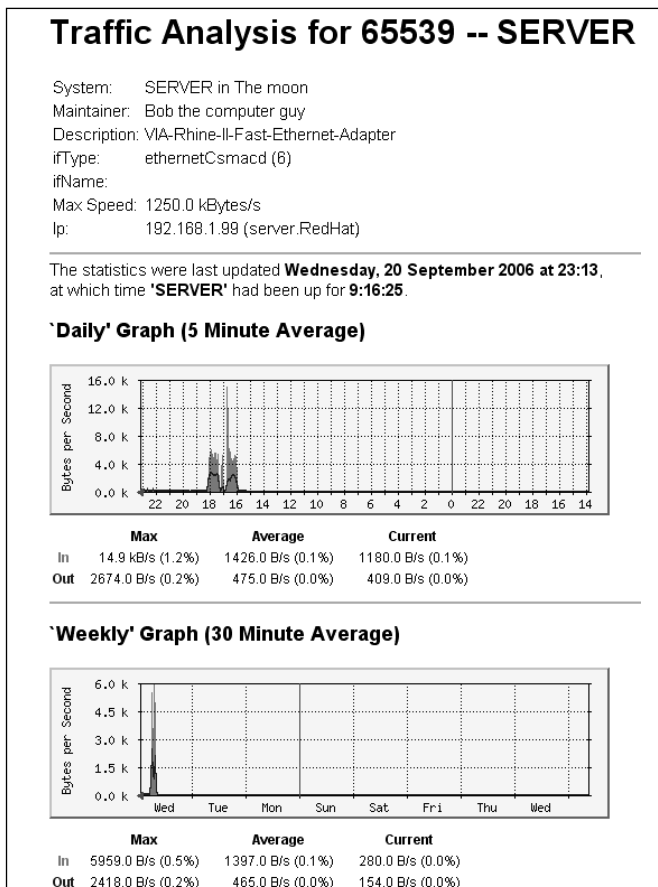
1. To edit the SNMP properties on a Windows XP system, open the Services plug-in (**Start | Run | services.msc**).
2. Locate the SNMP service and double-click it. Click the **Security** tab.
3. Ensure that the SNMP community strings are set correctly.
4. If you do not wish to restrict which machines can use SNMP to communicate with the SNMP device, select the **Accept SNMP packets from any host** radio button. Otherwise, ensure that the system running MRTG is listed in the **Accept SNMP packets from these hosts** section and click **OK**.

At this point you should have the `mrtg` file successfully created using `cfg-maker`. This file is only the starting point and will still require some manual editing before it's really ready to use.

1. You can now run MRTG using the following command from the `\bin` directory: **`perl mrtg mrtg.cfg`**. The first time you run the command you will get some warnings about missing log files. This is normal the first time it is run.
2. Several files will be generated, the primary one being an HTML file beginning with the target host's name/IP address. If you open this in a browser it will show the *bytes in* and *bytes out* traffic statistics.

A sample of the MRTG HTML output is shown in Figure 7.1.

Figure 7.1 MRTG HTML Output



By scheduling MRTG to run regularly, you can build a history of data points that are used to populate the graphs. MRTG can be run in *daemon* mode. In this mode, once you start MRTG it will not exit; it will stay running and continue to collect data. To do this requires two steps. First, edit the configuration file and add the line **RunAsDaemon: Yes** in the **Global Properties** section. Second, start MRTG using the following command from a command prompt: **start /b perl mrtg mrtg.cfg**. Yet another way to accumulate the needed data is to run MRTG as a Windows service using the *SRVANY.exe* and *INSTSRV.exe* Microsoft utilities. Given the wide range of options, there should be a way to execute MRTG that suits your environment.

MRTG has the benefit that there are no limitations on the number of systems you can collect data from. Although throughput is the default metric, if you know the OID of the metric you wish to monitor, MRTG can collect and record historical data for that as well. Some good examples might be disk space, CPU utilization, network errors, and available memory. As an example of specifying the target OID, here is a target specification:

```
Target [RTR] : 1.3.6.1.2.1.2.2.1.14.1&1.3.6.1.2.1.2.2.1.20.1:pass@192.168.1.25
```

The OID format is *<OID for first line in graph>&<OID for second line in graph>*. In this example, you would be specifying a target device at IP address 192.168.1.25, using the SNMP community string of “pass.” The OID to read would be 1.3.6.1.2.1.2.2.1.14.1 (input error counts) for one line on the graph, and 1.3.6.1.2.1.2.2.1.20.1 (output error counts) for the other line. The last number is the interface number. So the input error count on interface #2 would be 1.3.6.1.2.1.2.2.1.14.2. The label in the brackets (in this example “RTR”) would be displayed on the graph page as a device name.

As you can see, getting simple throughput graphs in an HTML page using MRTG is pretty painless. With a little work, you can configure MRTG to graph a wide range of useful information. I would highly recommend doing some reading on the MRTG page with third-party documentation (<http://oss.oetiker.ch/mrtg/3party.en.html>). This page has many articles describing how to use MRTG in various circumstances. There are also links to many graphical utilities to help you manage your configuration files. For increased granularity in your graphs (as small as one-minute intervals) and more aesthetic graphs, check out RRDtool at <http://oss.oetiker.ch/rrdtool/>.

Configuring MZL & Novatech TrafficStatistic

While TrafficStatistic isn’t the most full-featured solution available, it is very easy to get running. TrafficStatistic collects its information from network sniffing only, so you will need to place the host that is running TrafficStatistic in a location where it can see the network traffic you want to report on. Follow these steps to get TrafficStatistic running on an appropriately located host.

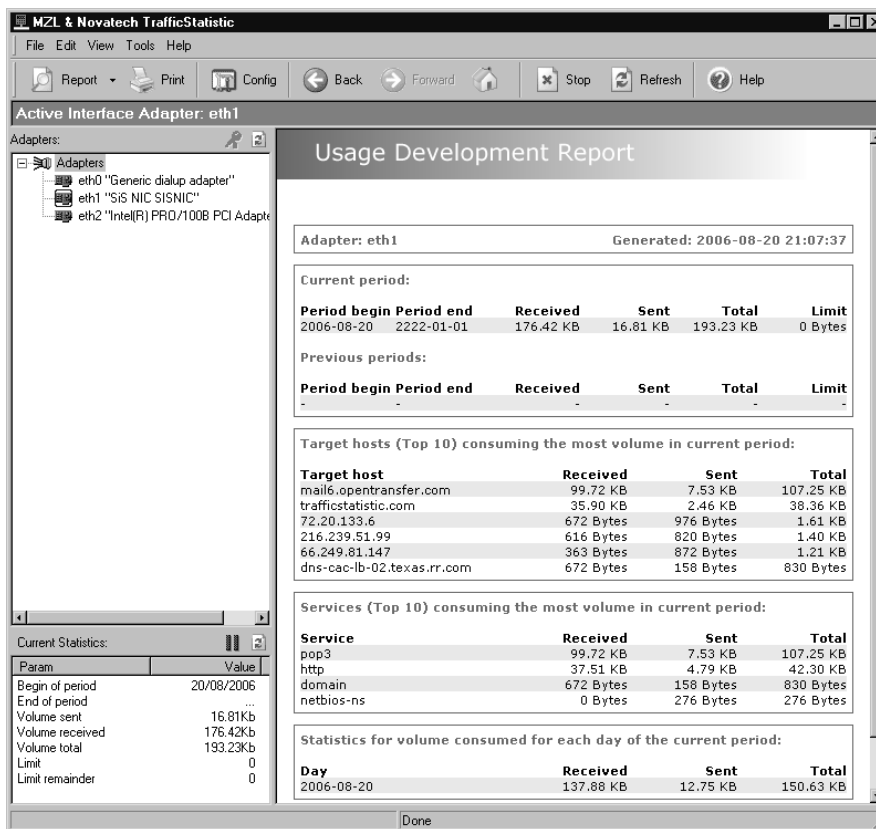
1. Download and run the installation file from www.trafficstatistic.com.

2. During the setup, take note of the HTTPSRV Service port (Default 7777) and the RunCPM Service Port (7778), altering them if desired. Other than that, you can simply accept the defaults.
3. When the setup completes, allow it to Start the IP capture service, HTTP server service, and run the GUI monitor.
4. If you have more than one network adapter listed in the Adapters pane, you will need to right-click the adapter you wish to collect data from and select Configuration.
5. Ensure that **Listen** is checked and click **OK**. You have the option of specifying billing periods at this time as well. These are targeted for ISPs and other providers who need statistics for certain billing periods to charge back to customers, but you could also configure them for your own reporting purposes. By default, the billing period will be from the current date with no end date.
6. After you click **OK** you will be prompted to verify your adapter selection. Click **Yes** to confirm your choice.

You should now be collecting data on the selected interface. The window should show several links in its default state. Click **Create report**, and then select **Usage Development**. In the next window, select the network interface you want to report on, and then click **OK**. The resultant report is shown below in Figure 7.2.

If you are reporting on addresses that are not “private” addresses as defined in RFC1918, you might find you aren’t really interested in traffic that is local to your network, and a top-talker list that is not using Internet bandwidth may not be useful. In this case you have the capability to define the local address and tell TrafficStatistic not to use them in the reports it generates. This list doesn’t truly have to be “local,” but it does enable you to configure TrafficStatistic to *not* report on certain addresses. This could be handy if there are certain hosts that are high traffic as part of normal behavior and you want to exclude them from the reports altogether. To configure the list of local addresses and exclude them from the reports, follow these steps:

Figure 7.2 MZL & Novatech TrafficStatistic Report



1. Ensure that you have the proper network interface selected in the Adapters pane.
2. Right-click the adapter and select **Configuration**.
3. Check the box labeled **Exclude traffic between local addresses**.
4. Click the button labeled **Edit LAL** (which stands for *local address list*).
5. Click the **Add Host**, **Add Range**, or **Add Net** button and enter the appropriate information.
6. When finished, click **OK** and **OK** again to completely finalize your changes.

Note that these changes will not retroactively affect current data that has been collected.

The business model that is used by the makers of the MZL & Novatech TrafficStatistic is to provide the basic monitor for free. If you want additional functionality, you can buy add-on plug-ins. The price of plug-ins varies greatly. The “Multi Optional Report” plug-in provides more-granular bandwidth reports that enable you to actually drill down and see what ports a particular host was using to generate the network traffic; it costs 50 Euro (approximately \$66). If MZL & Novatech TrafficStatistic provides adequate reporting for your needs, and you have a spare machine you can place appropriately to sniff the network traffic, this might be a good fit.

Configuring PRTG Traffic Grapher

PRTG Traffic Grapher is currently one of the best freeware options available. The same download is both the freeware version and a full-featured, time-limited trial version. PRTG is the only offering that supports data collection via sniffing, SNMP, and NetFlow. The graphs PRTG produces are very functional and a Web interface is provided that enables you to drill down into the data without having to be on the PRTG server or having to have any software installed. This means that if you want to collect data via sniffing, you don't have to worry about providing remote access to the PRTG server; you can access the reports via any Web browser. Follow these steps to get PRTG up and running.

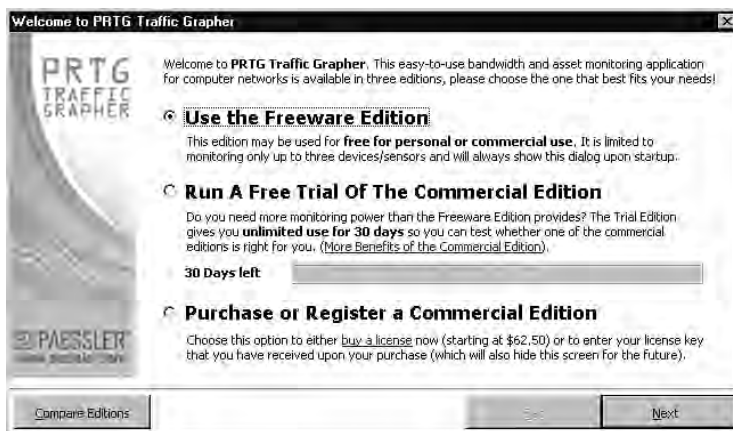
1. Download the setup file from www.paessler.com/download/prtg and run it.
2. Click **Yes** to confirm that you wish to install the freeware/trial version.
3. Click **Next**.
4. Select the **I accept the agreement** radio button to accept the license terms, and then click **Next**.
5. Select the installation directory, and then click **Next**.
6. On the **Select Components** screen you can select the defaults and then click **Next**.
7. Choose if you wish for the Web interface to be enabled or not. If you *do* want the Web interface enabled, you can leave it at the

defaults. The *PRTG Watchdog* service is a process that will monitor and restart the PRTG process if it terminates unexpectedly. You should leave this option enabled unless you have a specific reason not to. When satisfied with your selection, click **Next**.

8. Click **Finish** to complete the installation and start PRTG.

When you first run PRTG it will present you with a window where you can choose which version to install. The freeware edition (limited to three sensors), a trial edition (which will work for 30 days), or you can purchase the commercial edition. This window is shown in Figure 7.3. You also have the option of comparing the various versions. In this example we're looking at the freeware edition, so select the corresponding radio button (which should be the freeware version by default), click **Next**, and then **Finish**.

Figure 7.3 Version Activation Screen



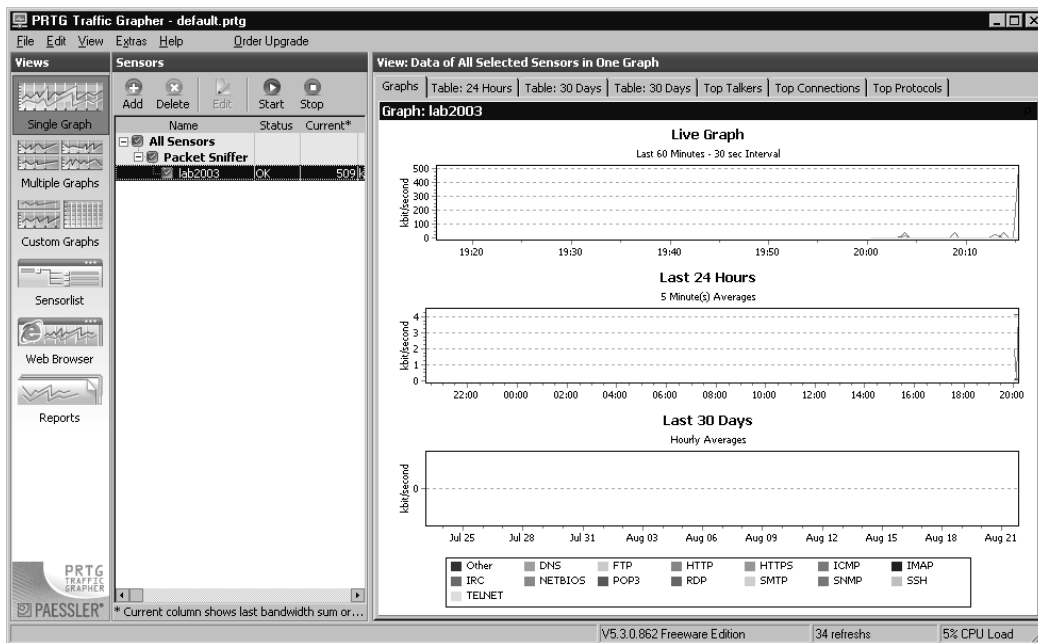
9. During the next step you will see an empty window, with a button in the center that says **Click here to add your first sensor!** Click this button.
10. Click **Next**
11. Choose which data collection method you wish to use. In this context a sensor can be any of several data collection devices, including a router supporting NetFlow, a PC sniffing traffic, or any device that will support SNMP (including Windows systems). For this example I

am using a PC that will sniff the network traffic. If your network infrastructure devices don't support SNMP data collection or NetFlow data, this may be your only option anyway. A final option is to make this installation a latency monitoring system. In this mode, the system will use pings to monitor the round trip time between this host and various other hosts on the network. After making your selection, click **Next**.

12. On the next screen, enter the name of the sensor or leave the default value.
13. Place a check next to the interface you wish to use and click **Next**.
14. On the next screen you have the option of excluding certain traffic. I would suggest leaving the default of **Monitor all traffic**. The filters can be edited later if desired. Click **Next**.
15. On the next screen you have the option of choosing what protocols to monitor (called channels). You can also define your own by clicking **Edit Port Filter Library**. For now, just click **Select All** and then **Next**. We will demonstrate creating your own "channel" shortly.
16. On the next screen you can choose a grouping for you to add sensors under (with a limit of three sensors this grouping probably isn't that critical, but if you had hundreds of devices it would be very useful). You can also select the scanning interval. Unless you have reason to do otherwise, simply leave the defaults in place and click **Finish**.

The main PRTG Traffic Grapher console is shown in Figure 7.4. There are three panes. The leftmost is called Views and enables you to select between different layouts and to display different data. The middle one, Sensors, enables you to select different sensors to see their data. The rightmost pane is **View: <description>**, and will change depending on what you select in the Views pane on the far left. While the same installation file is used for Windows XP and Windows 2000, once installed, the interfaces have slight differences. The differences are very minor and these instructions should work for either version.

Figure 7.4 PRTG Traffic Grapher Application



If you double-click on a given graph you will get an enlarged view that you can also use to edit the graph colors, units, and several other options. By default, the Web interface will be available on the IP of the machine you installed PRTG on, using port 8080. In this case you can open a browser to <http://192.168.1.104:8080>. The browser interface is shown in Figure 7.5.

You should now be collecting data, which should be visible in the graphs. You might wish to customize a few features though. If you wish to disable, or modify the Web interface (and many other settings) navigate to **Extras | Options**. In the left pane, select **Web Server** as shown in Figure 7.6. Uncheck **Enable Internal Webserver** if you wish to disable the Web interface completely. If you plan on leaving the Web server enabled, you should place a check next to **Write webserver access logfile**. You should also change the **Website Access Control** to **Limited Access**. Because the sensor data and reports could contain confidential information, the default of unlimited access to the Web interface is not secure.

Figure 7.5 PRTG Web Interface

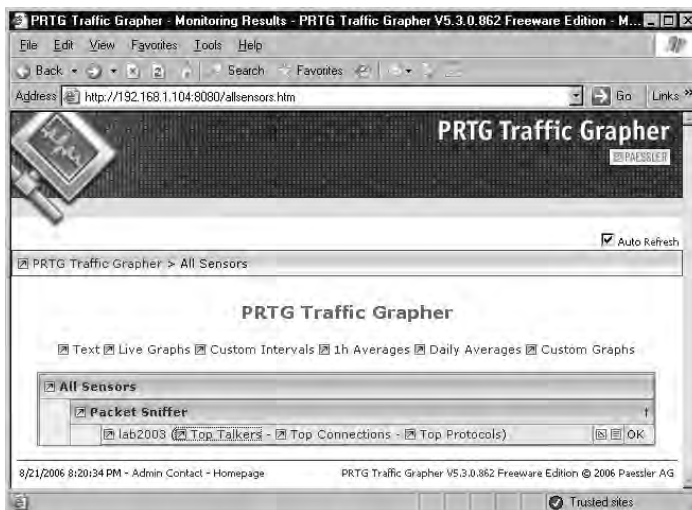
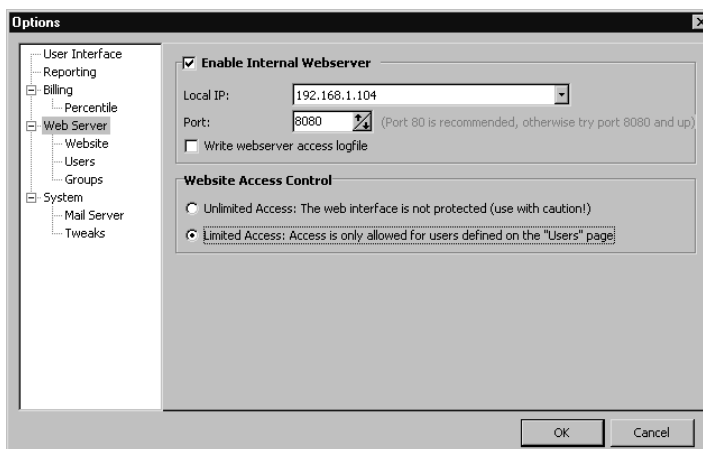


Figure 7.6 Options (Web Server) Screen



After configuring the **Web Server** options, select **Web Server | Users** in the left pane. The default configuration will be to permit the PRTG administrator only. Note that this is not the local machine's administrator account; this account is specific to PRTG. This might be all you need, but if you need to permit additional accounts click **Add** and enter the account information followed by **OK**.

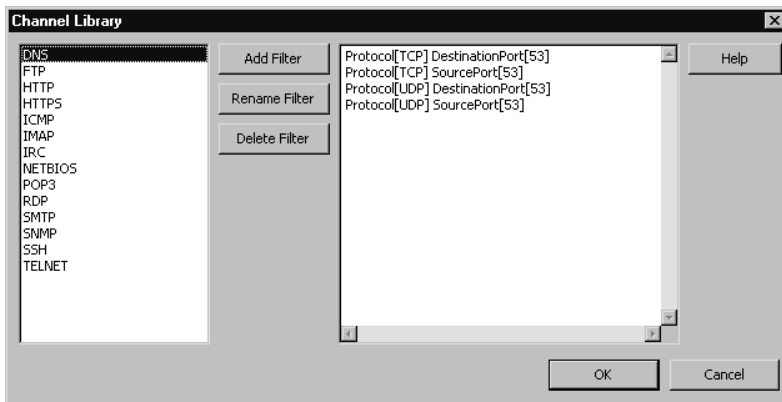
If you want to send e-mail alerts, you will need to configure the mail server options within PRTG. Do this by navigating to **Extras | Options |**

System | Mail Server. Enter the IP address or hostname for the SMTP server. Also enter an e-mail address, which will be the alert e-mail’s “from” address. If SMTP authentication is needed you can enter the username and password in this window as well. Once you are satisfied with your selection click **OK** to accept the changes.

Now let’s suppose you have some custom applications, or even just some applications you want to specifically target in the reporting. Any protocols/ports that do not have a channel defined will fall in to the “other” channel. This could be applications that were designed in-house using a non-standard protocol/port number, or a more common application that PRTG doesn’t have defined yet, such as syslog. You can add to the list of “channels” and define your own by following these steps.

1. Navigate to **Extras | Channel Library**. The Channel Library window is shown in Figure 7.7

Figure 7.7 Channel Library



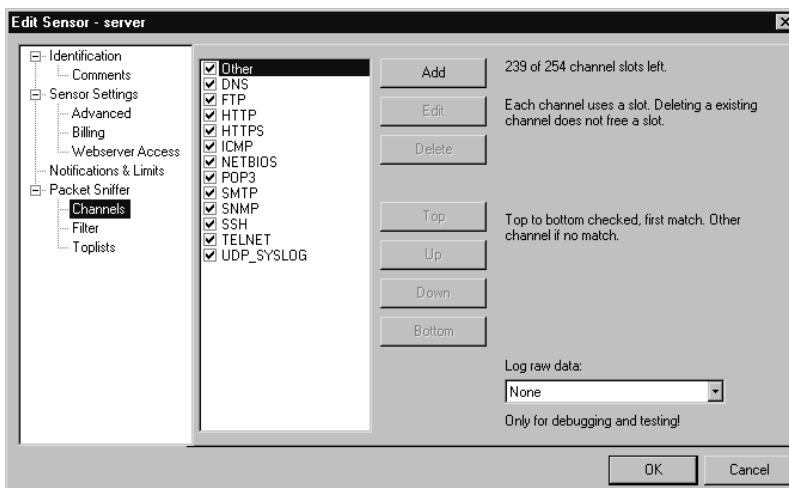
2. To add a specific graph entity for UDP-based syslog messages, for example, click **Add Filter**.
3. Enter a name for the channel, such as `UDP_SYSLOG`, for example, and click **OK**. The window will go back to the way it was, but the new channel name will appear in the list on the left. To edit the rules of the channel, select the channel to edit in the left pane, and then click in the right pane.

4. Enter **Protocol[UDP] DestinationPort[514]** and click **OK**. This adds the channel to the PRTG console but not to a specific sensor yet.

TIP

Remember that the “channels” use port numbers to identify an application. The reliability of this identification depends on the application using a consistent port. There is nothing stopping someone from running a Web server on TCP23 instead of port TCP80, in which case it will show up in the graphs as Telnet traffic. Other applications, like instant messengers and file sharing applications in particular, will use a wide range of ports in an attempt to find one that will get through corporate firewalls. Creating a filter to identify these based solely on port numbers will be unreliable at best.

5. Ensure that the proper sensor is selected in the **Sensors** pane and navigate to **Edit | Edit**. You’ll notice the next window is titled **Edit Sensor**.
6. Select **Channels** in the left pane and click **Add**, as shown in Figure 7.8.

Figure 7.8 Edit Sensor

7. Select the new channel in the left pane and click **OK**.
8. Use the **Top, Up, Down, and Bottom** buttons to place the channel in the order you desire. The channel matches work much like a firewall access control list in that PRTG will stop processing the list as soon as it finds a match. If there is no match it will categorize the traffic as “Other.”
9. Click **OK** again to close the **Edit Sensor** window.

You should now have a new channel displayed in the legend on the **Graph** tab. You are limited to only 254 channel definitions, though this limitation isn't likely to pose much of an issue in a smaller environment. You should now have PRTG collecting data via sniffing. You can view the graphs from the Web page or the PRTG console. You also have defined any additional ports you want PRTG to recognize as a specific application as its own “channel.” The final configuration options I will discuss are that of configuring the alerts. Odds are good you won't be able to sit and stare at the graphs all day and all night, so setting up some notifications might be one way to save time and energy, not to mention make you look like you have “network ESP” to your manager. To configure notifications and limits, navigate to **Edit | Edit** and select the **Notifications & Limits** tab. This tab provides the following options:

- **Error Notification** This will be triggered only if a sensor reports an error. Be aware that if there is a connectivity outage to the sensor, the sensor cannot report the error until connectivity is restored.
- **Threshold Notification** This is used to set specific upper *or* lower limits on a per-channel basis. An *optional* time span can be configured from seconds to days. If you do not specify a time span, the event will trigger as soon as the threshold is exceeded.
- **Volume Notification** This is similar to the threshold notification, except the volume is defined as an upper threshold and a minimum time span of one hour is required.
- **Limit Line** The limit line only serves to add a line to the 30-day graph. This could be useful, for example, to set the limit to 75

megabits on a 100-megabit network (75 percent) as a warning of when your infrastructure is reaching capacity.

All the notifications have several options that can occur when triggered. You can choose to send an e-mail, perform an HTTP *Get* request, execute a program/batch file, and change color of the graph background for each trigger. You can also combine multiple notification methods for a single trigger, such as changing the color of the graph background *and* sending an e-mail. Given that PRTG includes a fully functional notification system, this really makes PRTG stand out among its peers as one of the best free network reporting tools available. As an example, let's assume you have a single T1 line at work (1.5 megabits per second) and you want to send an e-mail if traffic levels for FTP exceed 1 megabit per second. Follow these steps to configure the notification.

1. Select the desired sensor in the **Sensors** pane; then right-click and select **Edit**.
2. Select **Notifications & Limits** in the left pane.
3. Click the **Add Threshold Notification** button.
4. Choose a name, such as 1Mb_FTP.
5. In the **Channel** drop-down box, select **FTP**.
6. In the **Threshold** section, select **over, 1, megabit per second**.
7. Under **Notification** in the left pane, select **Email**.
8. Place a check next to **Send Email**.
9. In the **Address** field, enter the e-mail address of the e-mail recipient.
10. Select the e-mail template you wish to use and click **OK**.
11. In the **Edit Sensor** window, click **OK** to accept the changes.

With the capability to execute an external program based on thresholds and volumes, the possibilities are near limitless. If you wanted to integrate your PRTG alerts into a syslog infrastructure, you could use the EXE notification method to execute a batch file that uses a command-line utility to generate a syslog message. I hope that I have demonstrated what a powerful and full-featured product PRTG is. There is very little functionality that is

unavailable in the free product, and a limitation of three sensors will likely pose little problem to a smaller organization. The graphing and reporting capabilities are exceptionally robust for a free product.

Configuring ntop

Ntop will run on many operating systems and while the initial setup will vary, once you have ntop installed, the configuration and usage is primarily via a Web interface, so the data will be presented in a uniform manner regardless of the underlying operating system. The ntop Web site offers multiple versions of the package for download. For Windows, they offer the source files, in which case you must compile them yourself, and they offer a pre-compiled binary distribution. The downside is that they have chosen to limit the precompiled version to capturing only the first 2000 packets, which makes it fairly useless to most people. This limitation does not exist in the Linux versions or the Windows source files. If compiling the source code is not a task your relish, you can download and install a precompiled version (meaning no 2000 packet capture limit!) of ntop from www.openxtra.co.uk, whose tagline is “network management for all.” OPENXTRA Limited has made available precompiled files for many popular packages, including Ethereal, MRTG, Net-SNMP, ntop, Windump, and Nmap. Follow these steps to install and configure the ntop package from OPENXTRA on a Windows host.

1. Download the installation file and execute the file.
2. Click **Next** on the welcome screen.
3. Accept the license agreement and click **Next**.
4. Enter a user name and organization if desired, choose whether the start menu entries will be made for all users or only the user who is running the Setup Wizard, and then click **Next**.
5. Choose which components to install (the defaults are recommended) and the installation target directory and click **Next**.
6. When the installation is complete, click **Finish**.

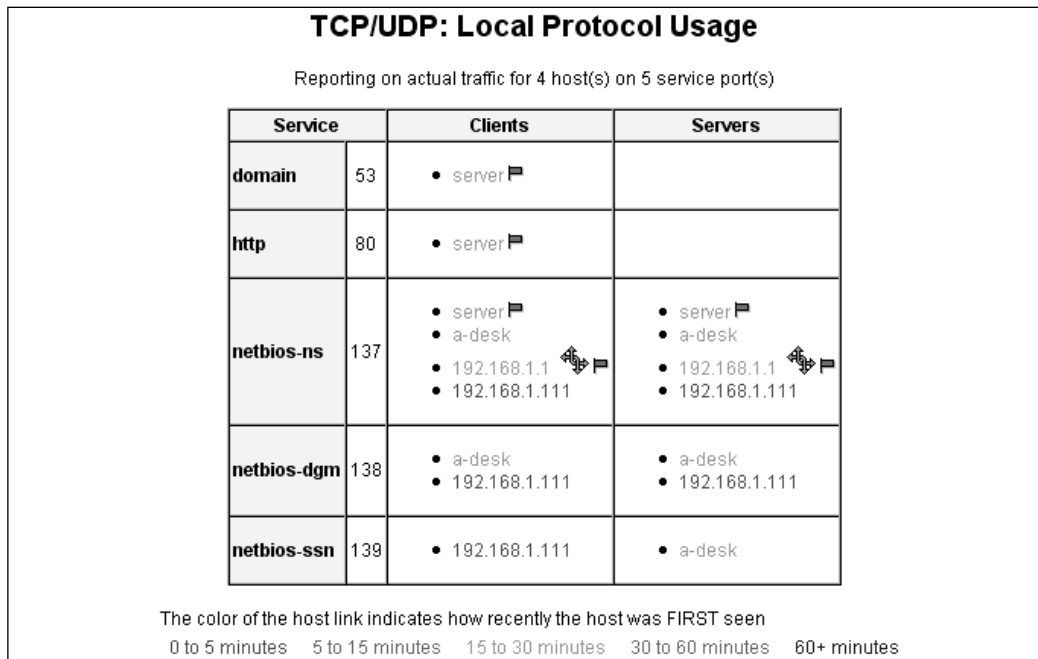
After the installation is finished, you should have a new icon in the system tray called **OPENXTRA Commander**. Double-click this icon to open the

OPENXTRA Commander. If the **NTop Service** plug-in is not started, click **Start** in the **Action** column to start it. Once it is started, click the **Launch** action for the NTop plug-in, which will open your browser. If all is well, you will already be collecting some impressive data. If you have more than one *network interface card* (NIC) in the host, you may need to select the proper NIC. Do this by selecting **Admin | Configure | Startup Options** in the menu listing at the top of the page. You will notice that some menu entries have a small padlock icon in them; these are the Web pages that require a password to access. The default credentials for the XTRA package is user = admin, password = admin. The top of the page will contain a listing of your network interfaces. You *can* collect data from more than one interface if desired. If you change the selected interface, you will need to use the OPENXTRA Commander to stop and restart the NTop Service.

The ntop FAQ can be a little hard to find; it's located at www.ntopsupport.com/faq.html. Because ntop is focused only on displaying data in a format that is easy to drill down into, there is very little to configure once you have it working properly. There is a host of information on the various screens. Any of the options under Summary can be useful. If you navigate to **IP | Local | Ports Used** you will see a screen similar to the one shown in Figure 7.9. This is just one of the many screens full of data that ntop provides; to include screen prints of all the available graphs and tables would require an inordinate number of pages.

Many of the elements on the ntop pages are actually hyperlinked to additional pages so that you can drill down and obtain more and more detailed information. Despite ntop's lack of native support for alarms and other active actions, you can perform several useful functions with a little customization. For example, the OPENXTRA Commander can be used to provide some handy links. By adding an appropriately configured .INI file to `\OPENXTRA\Common\Plugins\`, you can add entries to the Commander menu. For example, the following contents placed in an INI file would open Notepad.

Figure 7.9 ntop IP Ports Used



[APPLICATION]

NAME=Notepad

DESCRIPTION=Launch notepad

ACTIONDESCRIPTION=Start

VERSION=1.0.0

COPYRIGHT=ESS

COMMAND=notepad

COMMANDARG=

If you change the plug-in options, you will need to navigate to **View | Plug-ins | Reload Plug-ins** before your changes will take effect. There are also a few Perl scripts available in `\OPENXTRA\NTopWin32\www\perl\`. Any of these can be executed with **perl <scriptname>** from the command line. Here is a short summary of the provided Perl scripts.

- dumpFlat.pl** This returns a flat listing of all the host data in ntop for the current time slice in a Perl-like format, and loops every minute (by default) generating current data. Usage information is discussed further in this section.

- **mapper.pl** This script will use a host IP address to return a GIF flag of their location. This is what is used to generate the flag labels in various reports within ntop.
- **remoteClient.pl** This returns a Perl-like listing of all the host data in ntop for the current time slice. Usage information is discussed further in this section.

Being able to pull this raw data opens a host of possibilities. As a simple example, let's suppose you wanted to produce a listing of all the hostnames that ntop currently knew about. `Dumpflat.pl` is configured by default to loop and dump the output on one-minute intervals. `Remoteclient.pl` is configured for a Perl-like output, which isn't conducive to parsing in a DOS window. Copy the `remoteClient.pl` and rename it to `raw.pl`. Edit `raw.pl` and find the line that says

```
$URL = "http://".$ntopHost."/".$ntopPort."/dumpData.html?language=perl";
```

and change it to read

```
$URL = "http://".$ntopHost."/".$ntopPort."/dumpData.html?language=text";
```

Now executing **perl raw.pl** will produce the raw data held in memory by ntop. This output is easily parsable because each host it knows about has all of its data on one line. Now the following command would parse the output from `raw.pl` and print the fourth field (the hostname) to the screen for each line.

```
FOR /F "tokens=1-4 delims=|" %a in ('perl raw.pl') do @echo %d
```

You could also use `find.exe (perl raw.pl | find "192.168.1.99")` to pull out a line of data containing a known hostname or IP address. Once you have the raw data at your disposal you can perform a wide variety of parsing options on it.

In all likelihood, you will need to alter the list of defined port numbers. This serves two purposes in that it gives you greater insight into the applications being used on the network and reduces the amount of data that gets lumped into "other." Changes to the port list will affect the output of all screens under the **IP** menu option. The "create your own custom port definitions" follow these steps.

1. Create an application port list file in the `\OPENXTRA\NTopWin32\etc\` directory, such as **portlist.txt**.

2. Enter the applications you wish to be displayed in portlist.txt file. The following list can be placed in the portlist.txt. The format is *name that will appear in ntop=port or portname, as defined by the OS, or a port range*. The first three sections are what ntop will use if no portlist file is specified.



```

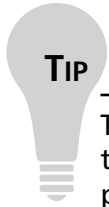
## Default ntop portlist ##
FTP=ftp|ftp-data
HTTP=http|www|https|3128
DNS=name|domain
Telnet=telnet|login
NBios-IP=netbios-ns|netbios-dgm|netbios-ssn
Mail=pop-2|pop-3|pop3|kpop|smtp|imap|imap2
DHCP-BOOTP=67-68
SNMP=snmp|snmp-trap
NNTP=nntp
NFS=mount|pcnfs|bwnfs|nfsd|nfsd-status
X11=6000-6010
SSH=22

## Default ntop Peer-to-Peer portlist ##
Gnutella=6346|6347|6348
Kazaa=1214
WinMX=6699|7730
DirectConnect=0      Dummy port as this is a pure P2P protocol
eDonkey=4661-4665

## Default ntop Instant Messenger ports ##
Messenger=1863|5000|5001|5190-5193

## Extra ports ##
Syslog=514
PCAnywhere=5631
SQL=1433

```

The Web page will display the application names in the same order as they are defined in the portlist.txt file. This means you should list the port definitions you want to see first earlier in the portlist file to keep from having to scroll the Web page to see them.

3. After you have your portlist file created, navigate to **Admin | Configure | Startup Options**.
4. Select the **IP Preferences** hyperlink.
5. In the field **TCP/UDP Protocols To Monitor**, specify the full path to the portlist.txt file.
6. Stop and restart the ntop process for the changes to take effect.

You can restrict additional ntop pages. If you want a person to have to supply the user name and password to view *any* ntop pages, you can configure this by navigating to **Admin | Configure | Protect URLs**. Click **Add URL** at the bottom and then click the **Add URL** button without filling anything into the field. This will password protect all ntop pages.

Ntop stores all of its active data in RAM, so if the system is reset, you lose all your data. There is a mechanism to store the data to disk. Be forewarned that logging all the data to disk can consume a large amount of disk space, so it will require careful monitoring. Ntop stores the data in RRD files (round-robin database). You can configure the RRD plug-in by navigating to **Plugins | Round-Robin Databases | Describe**. The active column should say Yes; if it says No, click **No** to toggle it to active. To help you decide which reporting tools to focus your energies on, Table 7.1 highlights the various features of the utilities we discussed.

Table 7.1 Reporting Tool Features

Reporting Utility Features				
	MRTG	MZL & Novatech TrafficStatistics	PRTG	ntop
SNMP	✓		✓	
Sniffing		✓	✓	✓
Netflow			✓	✓
Linux	✓			✓
Windows	✓	✓	✓	✓
Limitations	Only two lines per graph	No Report Customization	3 Sensor Limit	Data only
Notes	Supported on Netware			

Enabling SNMP on Windows Hosts

All that's a lot to digest, so let's enable the SNMP agent on a Windows system and see what it can do for us. To enable the SNMP agent on Windows XP, follow these steps.

1. Navigate to Start | Settings | Control Panel.
2. Open **Add or Remove Programs**
3. Select **Add/Remote Windows Components** on the left side of the window.
4. Highlight **Management and Monitoring Tools**, but do not check it.
5. Click **Details**.
6. Place a check next to **Simple Network Management Protocol** and click **OK**.
7. Click **Next** and then click **Finish**

Next you need to configure some specific SNMP settings on the Windows host. Do this by following these steps.

1. Click **Start | Run** and enter **services.msc** to open the services snap-in.

2. Double-click the **SNMP Service** entry in the right pane.
3. Select the **Agent** tab.
4. Enter the **Contact** information and the **Location** information. These are used as identifiers for the system when viewed from the management console.
5. Open your MMC.
6. Select **Computer Management** in the left pane.
7. Expand this to **Computer Management | Services and Applications | Services**.
8. In the **Service** section, place a check next to the types of SNMP MIBs you want to use.
9. Select the **Traps** tab.
10. In the **Community name** drop-down list, enter a community name, also called a *community string*, and then click **Add to list**. A community string is very much like a password and serves to limit who can use SNMP on a given device.
11. In the **Trap Destinations** section, click **Add** and enter the system you want to send traps to. This would be the SNMP management console. To use a previous example, this could be the IP address of the system running PRTG Traffic Grapher.
12. Select the **Security** tab.
13. Highlight the default community name of **public** and click **Remove**. If you leave public as a READ ONLY community, anyone will be able to read the SNMP data of your system. The default on most systems is “public” with read-only access and “private” with full control. As is always the case, you don’t want to use the defaults when it comes to passwords.
14. Click **Add** and enter a **Read Only Community Name**, and click **Add** again.
15. Repeat this process and add a community name with read/write access.

16. Select the radio button next to **Accept SNMP packets from these hosts** and click **Add**.
17. Enter an IP address or host name and click **Add**. In this example I entered the same IP as my PRTG management console.
18. Click **Apply** and **OK**.

Your Windows system is now ready to be managed using SNMP; all you need is a management console. In this case we can add it as an additional sensor to our PRTG system (remember the freeware version has a limit of three sensors). To add the Windows host to your PRTG console, follow these steps.

1. Open the PRTG application.
2. In the **Sensors** pane, highlight **All Sensors**.
3. Right-click and select **Add Sensor**. This will start the Add Sensor Wizard.
4. Click **Next**.
5. Select **SNMP** as your data acquisition type and click **Next**.
6. The default selection of **Standard Traffic Sensor** will enable you to see how much bandwidth is being used by the Windows system inbound and outbound. The **SNMP Helper Sensor** will enable you to view some more-detailed Windows-specific counters. For basic monitoring select **Standard Traffic Sensor** and click **Next**.
7. In the **Device Selection window**, enter a name for the sensor and the IP address. Choose your SNMP version (choose the highest one your device will support). Enter the community string to use for that sensor, and then click **Next**. It will connect to the sensor and enable you to select which interface to monitor.
8. Place a check next to the appropriate interface.
9. Choose which values to monitor (Bandwidth is the default) and click **Next**.

10. This final screen enables you to choose a group for the sensor listing and configure the scanning interval. The defaults are probably okay, so simply click **Finish**.

You should see a new set of graphs, with bandwidth in and bandwidth out, for the newly added sensor. To add a sensor that uses the SNMP helper freeware, the steps are mostly the same. The only significant difference is that you need to run the Paessler SNMP Helper Setup.exe file on the system to be monitored. This setup file can be found in the same directory that PRTG was installed in. The setup file has no real configuration options and is very easy to install.

After installing the SNMP helper, proceed with adding the sensor following the steps above, with the following changes. When you select the type of SNMP to use, simply select **SNMP Helper Sensor**, choose the **SNMP Helper Freeware** in the drop-down box, and then click **Next**. The PRTG console will connect to the agent and present you with a list of all the possible values you can monitor. After making your selections, click **Next**.

Enabling SNMP on Linux Hosts

If you want to collect data using SNMP from a Linux host, the first step is to see if the SNMP daemon (`snmpd`) is already installed. You can enter `chkconfig —list snmpd`, which will list the `snmp` daemon *if* it is installed, and indicate which run levels it is configured to start in. If the daemon is not installed, it can be installed using whatever method is appropriate for your distribution. On Fedora Core 5, for example, all it takes is `yum install net-snmp`. The daemons and some utilities are included in the `net-snmp` package. There is also the `net-snmp-utils` package, which includes utilities aimed more at using SNMP, such as MIB browsers and such, but does not include the daemon.

After you have `snmpd` installed, it needs to be configured. We will discuss two ways to configure `snmpd`. The current version of the daemon is very robust, which has the side effect of including a very complicated configuration file located at `/etc/snmp/snmpd.conf`. The simplest way to get SNMP up and running to just collect data, is to save the sample `snmpd.conf` file under a new name and create your own. The new `snmpd.conf` can contain only the single line `rocommunity labcommunitystring`. In this example, the read-only com-

munity string would be set to *labcommunitystring*. Obviously, you should select a secure password of your own. This type of configuration file is quick and easy but not overly secure.

You could also use the netfilter firewall and/or tcpwrappers to restrict SNMP access based on the IP address of the communicating system. Another option is to use the access control built into the SNMP daemon. To configure the *snmpd* access control, you will need to edit the *snmpd.conf* file. Granting access to the SNMP MIB requires four basic steps, which are all done in the configuration file. The example below includes a variety of configuration settings to hopefully provide a good idea of the possibilities. Add the following lines to the configuration file.



```
##          sec.name      source          community
## -----
com2sec    local         localhost       snmplocalstring
com2sec    internal       10.0.0.0/8     snmpcommunitystring
com2sec    admin_net      192.168.1.0/24 snmpadminstring
```

These lines serve to map a *community name* to a *security name*, thus the *com2sec* directive. The security name is just a name to use to reference the access you are granting. You can use whatever name you like. The source is the hosts that will use that access profile. You can also use a source of *default*, which will mean any IP address. The community is the community string to be used for that access. Note that there is no differentiation made between a read-only community string and a read-write community string. This is because the access that the security name will have will be defined later in the configuration file.



```
##          Access.group.name  sec.model      sec.name
## -----
group      Full_Group                 v1             local
group      RO_Group                  v2c            internal
group      Full_Group                 any            admin_net
```

The next section maps a *security name* into a *group name* using the *group* directive. In this case, we mapped both the local security name and the *admin_net* security name to the same group (*Full_Group*). The *sec.model* determines which version of SNMP (1 or 2) will be used/allowed. Using version 3 requires a different configuration altogether.

```
SYNGRESS ## MIB.view.name      incl/excl  MIB.subtree  mask
syngress.com ## -----
view all-mibs      included   .1
view limited       included   .1.3.6.1.2.1.2
```

This section defines a *view*, which basically defines a portion of the MIB tree. The *all-mibs* view includes the entire MIB tree, whereas the *limited* view includes only the .1.3.6.1.2.1.2 section, which consists of the network interfaces and related metrics. You can configure a given view's MIB subtree to be *included* or *excluded*. By excluding, you can omit certain sensitive portions of the MIB tree. You can use the MIB tree name instead of the number if desired. In the example used above, you could use *included .1.3.6.1.2.1.2* or you could instead use this command: *included .iso.org.dod.internet.mgmt.mib-2.interfaces*. The mask is an *optional* bit mask on the MIB tree that specifies which bits to match for the indicated access. The default (without the mask) is to match only the bits that are indicated in the view. You would most commonly use the mask to match a particular row or rows in a table. So in this example, the all-mibs view has access to .1 and everything under it.

```
SYNGRESS ## MIB.group.name context sec.model sec.level prefix read      write  notif
syngress.com ## -----
access Full_Group ""      any      noauth   0      all-mibs none   none
access RO_Group  ""      any      noauth   0      limited none   none
```

The final section is where you define access for a *group name* to a *view*. The context is optional and *no context*, "" is the default. For SNMP v1 or v2, the sec.level needs to be *noauth*. Under the read and write column, you define the view. The notify column is where you would specify the capability to send traps, except the notify view is not currently implemented.

After making your changes, you will need to restart the *snmpd* for the changes to take effect. So what does all this accomplish? The MIB is accessible from the localhost using the *snmplocalstring* as the community string, which will provide read access to the entire MIB tree. Users from the 10.0.0.0/8 networks, which are assumed to be the bulk of the user base, can access the MIB using the *snmpcommunitystring* community string, and they will have read-only access to the network interface subtree only. Hosts on the 192.168.1.0/24 subnet, a network management subnet for example, can use

the `snmpadminstring` community string to gain read-only access to the entire MIB tree.

After all this configuring you might be wondering what the difference is between one version of SNMP and another. As the protocol has matured, it has gone through several changes, the majority of which revolve around security. The basic differences between one version and another are highlighted for your reference.

- **SNMP v1** Included only basic functionality and sent SNMP data using clear text.
- **SNMP v2** Mostly introduced security features, including two new branches of the MIB tree, 1.3.6.1.5 (security) and 1.3.6.1.6 (SNMPv2). Introduced the *GetBulk* operation, used for requesting large amounts of data in a single request. SNMP v2 also sends information using clear text.
- **SNMPv3** Introduced the *digest authentication protocol*, which is used for data *integrity*, ensuring that the message that was sent is the same one that is received using MD5. Introduced the *symmetric privacy protocol* to ensure data *confidentiality* using Data Encryption Standard (DES) encryption. Note that DES is not generally considered secure by modern standards and was replaced with triple DES (3DES), and, more recently, with *Advanced Encryption Standard (AES)*. Introduced the *User-based Security Model (USM)* and the *View-based Access Control Model (VACM)*.

Troubleshooting Network Problems

Inevitably, sooner or later, something will go wrong on your network. In most cases you can troubleshoot the problem using standard resources such as the error message and syslog logs. There will be those rare times, however, where the standard information sources just aren't giving you enough information to really know what is going on. In those cases, a network sniffer may be your only option. A sniffer will basically look at all the traffic at the network layer in its raw form and collect it. This means you can see the data the way your network card sees it, which is very different from how the final product looks

to the application. Different sniffers provide varying levels of sophistication and features.

The primary difference between a commercial sniffer and a free one will be in remote management and in some cases advanced analysis options. In many cases the sniffer will require special drivers to be installed in order to sniff traffic. The sniffer's role is mostly to simply display the data it collects. It is up to the human user to interpret the data and determine what it means in the big picture. For this reason, the analysis of sniffer output (often called traces) is typically reserved for experienced networking experts. With a little basic information, however, a sniffer can often provide a quick look into the otherwise unseen background of network communications. We will explain the installation and *basic* use of GUI sniffers and a couple command-line sniffers as well.

Using a GUI Sniffer

If you search on Google for a network sniffer, one of the first hits will be Ethereal. Ethereal is one of the oldest, most full-featured, and functional sniffers available for free. While Ethereal still has a Web site up at www.ethereal.com, the project has taken a new turn and is now known as Wireshark (www.wireshark.org). Wireshark has some features not found in many of the other free sniffers that are available, such as conversation reassembly and a capture display/filter syntax that is more advanced than most. Wireshark does require the WinPcap drivers, which are very simple to set up. WinPcap can be downloaded from www.winpcap.org. We walked through the installation of WinPcap in detail in Chapter 4, “Configuring an Intrusion Detection System,” so we will only cover it at a high level here. The WinPcap drivers enable a greater degree of access and control to the network communications at the packet level than is available by going through the Windows network drivers. For this reason, many third-party utilities that do heavy packet manipulation make use of WinPcap. Follow these steps to install Wireshark on a Windows system.

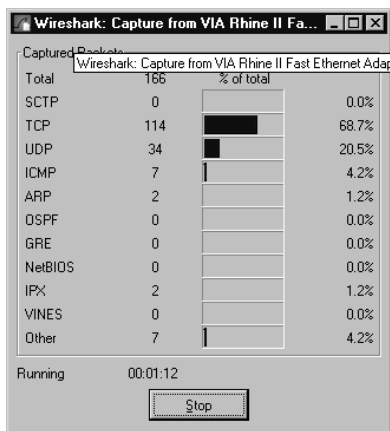
1. Go to www.wireshark.org and download the Windows installer file.
2. Run the setup executable. During the installation, you will be asked if you want to install the WinPcap drivers. Unless you already have

the latest WinPcap drivers installed, leave the default checked, which will install the WinPcap drivers.

3. Start Wireshark manually or allow the Setup Wizard to start Wireshark when it completes.
4. If you have more than one interface, navigate to **Capture | Interfaces**. This will provide you a listing of interfaces Wireshark can use.
5. Choose the one you wish to capture on and click **Capture**.

You are now sniffing the network. Remember, in a switched network you will only see traffic going to or from your machine, and broadcast traffic. You will need to enable port mirroring to see all traffic going through a switch. Wireshark will show a small window with statistics on the traffic it has collected, as shown in Figure 7.10.

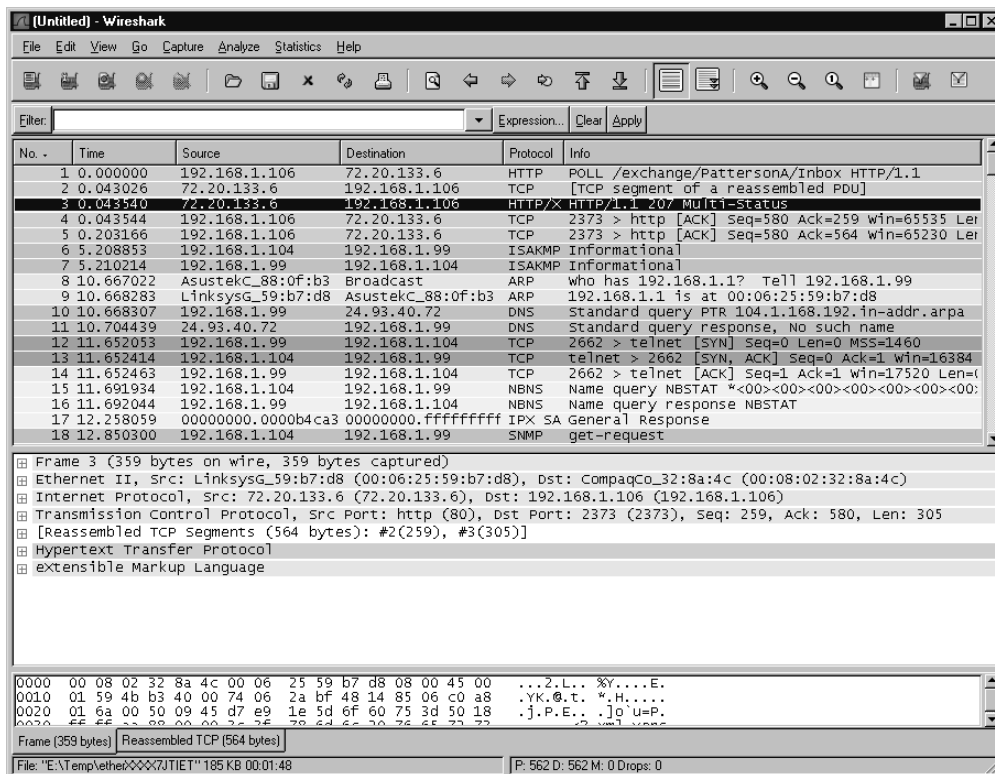
Figure 7.10 Wireshark Traffic Statistics



When you click **Stop**, the statistics window will close and the capture window will be populated with the data you have collected. The layout of Wireshark is similar to most packet capture programs. The top pane, or packet list pane, shows an entry for each packet, with some high-level information like source and destination hosts, protocol, and some other basic information. The second (middle) pane, known as the packet details pane, shows the packet contents, by header. Each header can be expanded to show more detail, which also enables you to expand the next header. The bottom pane, or packet bytes

pane, shows the raw packet data in hex and ASCII (plain text). The capture console is shown in Figure 7.11.

Figure 7.11 Wireshark Capture Console



The packets will be color coded by protocol. A useful feature of Wireshark is the conversation reassembly. You will notice packet number 12 is the initial packet on a Telnet connection. If you right-click one of the Telnet packets and select **Follow TCP Stream**, Wireshark will use the protocol information and sequence numbers to compile a list of the packet data pertaining to that conversation. If you would rather see the packets pertaining to that conversation and not just the data, right-click on a Telnet packet and select **Conversation Filter | TCP**. This will change the display filter to show only the packets involved in the Telnet conversation. These can be very powerful and save a lot of time when trying to follow a conversation and see where an error might be.

Many Linux distributions come with Ethereal installed by default. Ethereal is the previous version of Wireshark and in most cases will do what you want without needing to upgrade. If you want the most recent version, you can install Wireshark. Unlike Windows systems, which are pretty consistent when it comes to installing software, on Linux there are many different methods that can be used for installing software. Some are specific to that distribution, while others may be found on many different distributions. One of the most common methods for installing software from the command line is by using the RPM Package Manager (RPM). RPM is supported on distributions that are based on Red Hat Linux and its various flavors. This will include a large number of distributions including some live CDs. You can view a list of all installed packages by entering:

```
rpm -q -a
```

To install the Wireshark package on Linux, you need to first obtain the package file itself (from www.redhat.com/download/mirror.html, for example) or use the RPMs that were included on the installation CDs for your distribution. Different distributions may have packages specific to their configuration, or a given distribution may not support the newest version of a given piece of software. The first step is to research the latest version of Wireshark your Linux distribution can use and download the .RPM file. Then enter the following command to install Wireshark (for this example, it is Wireshark version 0.99.2-1):

```
rpm -i wireshark-0.99.2-1.src.rpm
```

If the installation is successful, you should see output similar to the following:

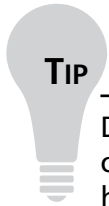
```
Preparing... ##### [100%]
 1: wireshark-0.99.2-1 ##### [100%]
```

You can even install a package directly from the Internet by specifying the full FTP or HTTP path as the path to the RPM as follows.

```
rpm -i ftp://somesite.com/5/i386/RPMS/wireshark-0.99.2-1.src.rpm
```

To uninstall the package you must use the package name, which is not the same as the name of the RPM file. To uninstall `wireshark-0.99.2-1`, enter the following command, using the `-e` switch for *erase*:

```
rpm -e wireshark-0.99.2-1
```



TIP

Due to the number of different methods to support software installation on the different versions of Linux, I couldn't possibly discuss them all here. The home page for your particular distribution will be the best place to start when it comes to getting instructions on installing new software packages. As just one example, here is a brief summary of the many tools available for software installation on just Fedora Core 5. You will need to research which methods are available on the distribution you are using.

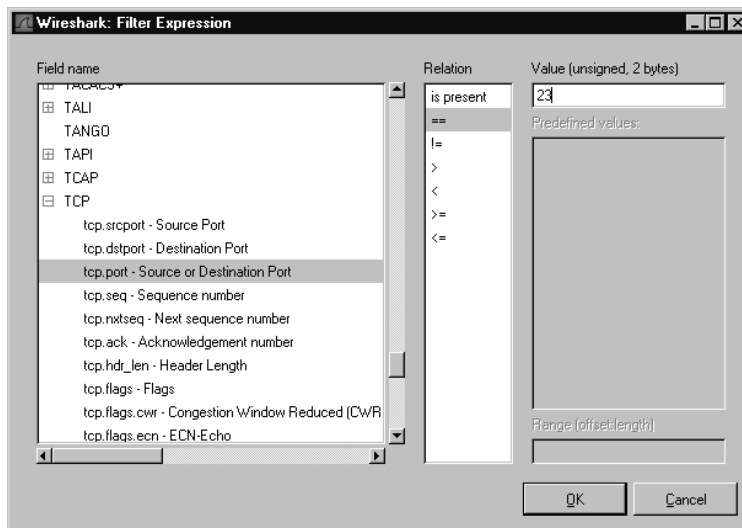
- **pup** GUI tool for updating software, accessed at Applications | System Tools | Software Updater.
- **pirut** GUI tool for managing software packages, accessed at Applications | Add/Remove Software.
- **RPM** Command-line tool for managing software packages.
- **yum** (Yellowdog Updater, Modified) command-line tool for managing software packages.
- **yum Extender** A GUI for yum (install with `yum install yumex`).

To make things more confusing, all the package management systems work slightly differently. Pirut and yum will automatically ensure you have the most current version of a package. Both of these will also automatically check and install any dependencies for the software you install. RPM does not include this functionality; you will need to check for dependencies manually when using rpm. Yum only works on RPM-based systems, and not all systems will have yum available or installed; therefore, it is suggested that you understand how to manage packages with RPM even if you choose not to use RPM for your day-to-day management.

Wireshark offers some features that you would normally have to pay a lot of money for and that the other free sniffers don't have. Let's suppose you have captured some packets and you want to see the Telnet conversation.

Locate any single Telnet packet, either through browsing in the main window, or by using a filter. To filter for all TCP packets with a source or destination port of 23 (Telnet), click in the **Filter** field and enter **tcp.port == 23** followed by **Apply**. Although the filtering syntax looks pretty intuitive, there will be times when you don't know what key words you need to use. You can construct the filter by clicking the **Expression** button. This will bring up the window shown in Figure 7.12.

Figure 7.12 Wireshark Filters



Navigate to **TCP** in the **Field name** pane, then expand the TCP subtree and select **tcp.port**. After highlighting a selection in the **Field name** field, select a logical operator in the **Relation** field, such as “==” for equals. Finally, enter the desired value (in this case 23) in the **Value** field. Then click **OK**. This will place the filter expression in the Filter field. You must then click **Apply** to apply the filter to the packet display. You should now have a list of only Telnet packets. If you select any one of the Telnet *data* packets (Wireshark will label them as such in the Info column) and right-click, you can select **Follow TCP Stream**. A separate window will open as shown in Figure 7.13, which displays all the data portions of the packets for that conversation.

Figure 7.13 Wireshark Follow TCP Stream



The complete output from the Follow TCP Stream window is included to give you an idea of what a Telnet login looks like. This also illustrates the fact that Telnet does not employ any encryption, making it an insecure choice for remote access except over trusted networks.



```

..'.%. . . . . !. ". . . . .
..%. . . . . ' . . . . .
..'. . . . . ' . SFUTLNTVER . SFUTLNTMODE . . . % . . . . . !. ". . . . .
. . . . . ] . . . . .
..'. . . . . DISPLAY . Knoppix : 0 . 0 . . . . ' . . . . % . . . .
Welcome to Microsoft Telnet Service

login:

password:

. . . . .
. . . . . xterm .
. . . . .
. . . . . xterm .

*=====
Welcome to Microsoft Telnet Server.
*=====

C:\Documents and Settings\Eric>

Volume in drive C is D1P1_OS
Volume Serial Number is 2037-12FA

```

```
Directory of C:\Documents and Settings\Eric
```

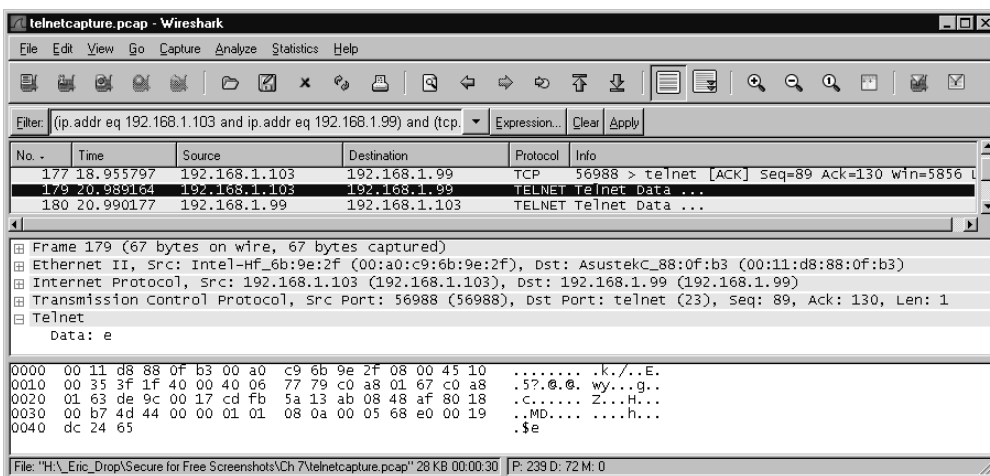
```
12/08/2003  06:52 PM    <DIR>          .
12/08/2003  06:52 PM    <DIR>          ..
05/02/2005  11:30 PM    <DIR>          My Documents
05/02/2005  11:30 PM    <DIR>          Favorites
12/08/2003  06:44 PM    <DIR>          Desktop
12/08/2003  06:44 PM    <DIR>          Start Menu
12/23/2003  12:15 AM    <DIR>          WINDOWS
02/16/2006  08:46 PM    <DIR>          .housecall
06/13/2006  08:15 PM    <DIR>          SecurityScans
06/24/2006  06:43 PM    <DIR>          .ssh
09/19/2006  03:27 PM                600 PUTTY.RND
09/04/2006  04:22 PM    <DIR>          Tenable
                1 File(s)                600 bytes
                11 Dir(s)    2,719,571,968 bytes free
```

```
C:\Documents and Settings\Eric>
```

I did remove some blank lines to conserve space, but other than that, this is the stream in its entirety. You can see the start of the conversation included *DISPLAY.Knoppix*, which was the system I was connecting from. The next line was a *Welcome to Microsoft Telnet Service*, which indicated the target host was a Microsoft system running the Telnet service. I was prompted for a login ID (I used eric) and a password. After the login was complete, I used the *dir* command to get a directory listing of the remote machine.

You will notice the user name and password are not listed in the follow TCP stream output. Because Telnet is a clear text protocol, the user name and password used is available in the data portions. Telnet will send a packet for each character you type, so with a user ID of “eric,” one packet will contain only an “e” for the data, another an “r,” another an “i,” and finally a “c.” To see this in the Packet Capture window, use the top pane to select a Telnet data packet. In the center packets, you can view the details for the various packet layers. At the bottom will be Telnet. Click the plus sign and it will expand to show the data portion of the packet. Figure 7.14 shows that packet #179 contains the “e” for its data payload, which is the beginning of me entering my login ID.

Figure 7.14 Wireshark Telnet Data



You can also search for a string within the capture by navigating to Edit | Find Packet. You then have the option of searching via a display filter, hex value, or string. If you were to select String and then enter any ASCII string in the provided field, it would search the data portion of all the captured packets. This can help you quickly locate a conversation of interest, such as searching for a URL to find an HTTP conversation you want to inspect. You'll notice that if you were to search for a string containing a Telnet user logon ID "eric," that you would never find it, because it is sent one character at a time. If you didn't know Telnet worked this way, you might be confused and incorrectly conclude that Wireshark did not capture your Telnet session. As you can see, analyzing sniffer data is not really something that you can just read about. Being able to get useful information and troubleshoot issues from a packet capture takes experience and an understanding of the how the protocols in question behave on the network.

Using a Command-Line Sniffer

Sometimes you need a sniffer with a little less overhead. A command-line sniffer has less to process and will place less of a load on the system you are running it on. Sometimes you might be in a situation where you only have console (command-line) access and can't use a GUI tool. At these times a command-line sniffer might be able to provide the sneak peek into the net-

work traffic you're looking for. I will briefly explain the use of a few command-line sniffers. One of them will require the WinPcap drivers, the other does not if you are using it on a Windows 2000 or newer operating system, and the last is a Linux command-line sniffer.

Windump

Many Linux systems have tcpdump installed by default. Because tcpdump is so common (at least on non-Windows systems), it has been ported to Windows and is called WinDump. WinDump requires WinPcap in order to work. WinDump can be downloaded from www.winpcap.org/windump/install/. There is no setup file; the .EXE you download is the entire package. Simply download it and place the file in a directory of your choosing. The manual for WinDump is pretty much the same as for tcpdump, and is located at www.winpcap.org/windump/docs/manual.htm. Although there are far too many options to explain them all in detail, here are a few that will get you started.

Windump -D lists the interfaces WinDump can see to capture on.

Windump -i 2 listens on interface number 2. You could restrict the output to show traffic only to or from the host named lab2003 by entering *windump -i 2 host lab2003*. Although the command-line syntax is relatively intuitive, there are a lot of options. The manual is very good, with some useful examples at the bottom. In our Wireshark example, we worked with a Telnet session. If we wanted to see traffic to or from host lab2003, with a source or destination port of 23 (used for Telnet), we could enter *windump -I 2 -host lab2003 and tcp port 23*. I opened a Telnet session with lab2003 and the partial output is shown in Figure 7.15. Note: I removed the time stamps so that every line would not wrap to the next line.

Figure 7.15 Windump of Telnet Session

```
windump: listening on \Device\NPF_{C428C1BF-C15A-460B-90D6-3A6F5DF68F22}
IP server.RedHat.2714 > LAB2003.23: S 639728410:639728410(0) win 16384 <mss
1460,nop,nop,sackOK>
IP LAB2003.23 > server.RedHat.2714: S 1106948603:1106948603(0) ack 639728411
win 16384 <mss 1460,nop,nop,sackOK>
IP server.RedHat.2714 > LAB2003.23: . ack 1 win 17520
IP LAB2003.23 > server.RedHat.2714: P 1:22(21) ack 1 win 17520
```

```
IP server.RedHat.2714 > LAB2003.23: P 1:4(3) ack 22 win 17499
IP LAB2003.23 > server.RedHat.2714: P 22:30(8) ack 4 win 17517
IP server.RedHat.2714 > LAB2003.23: P 4:28(24) ack 30 win 17491
IP LAB2003.23 > server.RedHat.2714: P 30:65(35) ack 28 win 17493
IP server.RedHat.2714 > LAB2003.23: P 28:31(3) ack 65 win 17456
IP LAB2003.23 > server.RedHat.2714: . ack 31 win 17490
IP server.RedHat.2714 > LAB2003.23: P 31:88(57) ack 65 win 17456
IP LAB2003.23 > server.RedHat.2714: P 65:228(163) ack 88 win 17433
IP server.RedHat.2714 > LAB2003.23: P 88:133(45) ack 228 win 17293
IP LAB2003.23 > server.RedHat.2714: . ack 133 win 17388
IP server.RedHat.2714 > LAB2003.23: P 133:319(186) ack 228 win 17293
IP LAB2003.23 > server.RedHat.2714: P 228:419(191) ack 319 win 17202
IP server.RedHat.2714 > LAB2003.23: . ack 419 win 17102
IP server.RedHat.2714 > LAB2003.23: P 319:320(1) ack 419 win 17102
```

One obvious thing you will notice is that by default WinDump shows only the high-level header information, not any of the packet data. To display this data, you will need to use the `-X` or `-XX` switch. By default, WinDump will display only a certain amount of the data portion of the packet; this is determined by the “snap length.” This is set using the `-s` option on the command line and will default to 68 bytes if it is not set manually. WinDump’s extensive filtering options make it a good tool for those times when you need very specific information. You can restrict the output using a number of parameters and end up with a very specific capture of the network traffic.

ngSniff

ngSniff is a little less robust than WinDump/tcpdump, but can be very handy to have in your network toolkit. The primary asset of ngSniff is that if you are using Windows 2000, Windows XP, or Windows Server 2003, you don’t need to install any drivers to start capturing traffic. This makes it a much more attractive candidate when you need some quick insight into what is occurring on the network but don’t want to run any setup programs or alter the drivers. If you type `ngSniff` with no options, it will show you the help screen and all the options ngSniff accepts. `ngSniff -list-interfaces` will list the available interfaces. To limit sniffing to only traffic to or from lab2003, you can enter `ngSniff -interface 0 -only-host lab2003`. Here is a sample capture of an SNMP message from 192.168.1.104 to 192.168.1.99.



```
IP HEADER 192.168.1.104 -> 192.168.1.99
```

```
-----
IP->version: 4
IP->ihl: 5
IP->tos: 0
IP->tot_len: 77
IP->id: 20699
IP->frag_off: 0
IP->ttl: 128
IP->protocol: 17
IP->checksum: 13275
```

```
UDP HEADER
```

```
-----
UDP->sport: 4337
UDP->dport: 161
UDP->ulen: 57
UDP->checksum: 48040
```

```
----- Begin of data dump -----
30 2f 02 01 00 04 09 74 65 73 74 77 72 69 74 65  0/.....testwrite
a0 1f 02 03 00 ee 18 02 01 00 02 01 00 30 12 30  .....0.0
10 06 0c 2b 06 01 04 01 cb 00 01 01 02 03 00 05  ...+.....
00
----- End of data dump -----
```

Because of its capability of being used without requiring any installation, ngSniff should have a place in your networking toolkit. I have seen it used on production systems where installing anything would have not been desirable during business hours, but we could run ngSniff from a USB pen drive and see what was happening with the application and resolve the issue very quickly. If you would like a basic GUI sniffer that does not require any installation (on Windows 2000 or newer systems), you can also try SmartSniff from www.nirsoft.net/utills/smsniff.html or IP Sniffer (part of IP Tools package) from <http://erwan.l.free.fr/>.

Tcpdump

Tcpdump is native to Linux and does not run on Windows systems. Tcpdump is installed by default on a large number of Linux/UNIX systems. Because tcpdump is so widely used, there is a wealth of support information and articles on the Internet on how to use tcpdump. The syntax and usage is nearly identical to that of WinDump, which we have already discussed, so what I will do here is demonstrate how to install tcpdump on a Linux system if you find it isn't already installed.

In order to install the tcpdump package, obtain or locate the appropriate package file for your distribution. Different distributions may have packages specific to their configuration, or a given distribution may not support the newest version of a piece of software. Then enter the following command to install tcpdump (for this example it is version 3.9.4-1):

```
rpm -i tcpdump-3.9.4-1.i586.rpm
```

If the installation is successful, you should see output similar to the following:

```
Preparing...                               ##### [100%]
 1: tcpdump-3.9.4-1                         ##### [100%]
```

To uninstall the package you must use the package name, which is not the same as the name of the RPM file. To uninstall tcpdump 3.9.4-1, enter the following command, using the *-e* switch for *erase*:

```
rpm -e tcpdump-3.9.4-1
```

We hope that by this point you have some idea of the troubleshooting power a sniffer can provide. You might wonder why you wouldn't just put Wireshark on every system in case a troubleshooting issue comes up. Consider that if a hacker manages to gain access to one of your systems; a network sniffer can be an indispensable information-gathering tool for the hacker. If you provide a pre-installed and configured sniffer for the hacker to use, you might make the hacker's job a lot easier. On top of that, installing additional software on production systems is generally something you want to minimize if at all possible. Although the WinPcap driver install has been very dependable for a while now (which wasn't always the case), installing network drivers always carries the risk of disrupting network communications if something

goes wrong. These are two very compelling reasons for using ngSniff or a comparable GUI-based sniffer that does not need to be installed on the system. In most cases these sniffers that use Windows raw sockets can be run from a pen drive or from a CD-ROM, or even run from a mapped network drive. Table 7.2 compares the features of four different sniffers.

Table 7.2 Sniffer Features

Sniffer Features				
	Wireshark	Windump	TCPDump	NGSniff
GUI Interface	✓			
Command Line	✓	✓	✓	✓
Special Drivers	✓	✓		
Windows	✓	✓		✓
Linux	✓		✓	
Notes	Most analysis options			No Install Needed

Additional Troubleshooting Tools

Because there are so many interdependencies that are involved in making a network-based application work properly, there is no single tool to isolate and identify problems. Expensive suites of software and data collection probes attempt to make the process as automated as possible, but in most cases the key tool is the human operator. Based on the symptoms of the problem, or lacking a specific symptom, starting at the bottom and working your way up, the human performing the troubleshooting has to systematically use process of elimination to try to identify the problematic component, and then take steps to remedy the issue. In most cases you will not need more than the basic troubleshooting tools available with any modern operating system, such as ping, traceroute, and various commands specific to the operating system. On some occasions, some specialized software can make the task of troubleshooting much easier. A couple of my favorites are explained here.

Netcat

Netcat has often been described as the “Swiss Army knife” of troubleshooting tools. This is because its function is so elegantly simple that it has many uses. In essence, it establishes a communications session between two systems. You can enter `nc -l -p 2222` to tell system one to listen on port 2222. On system two, the command `nc 192.168.1.10 2222` would connect to system 1 (assuming the IP address is 192.168.1.10) on port 2222. After the connection is established, anything typed at the console of either system is sent to the stdout of the other system. Although both ends of the session can be the netcat executable, they don’t have to be. In addition to providing a command line based method for transferring data, netcat enables you to test connectivity over arbitrary ports. By entering `nc <IP Address> 23`, you can connect to a system to verify that Telnet is listening and available. In fact, if you use the `-t` switch, netcat will negotiate the Telnet specifics. Although Telnet is often used in exactly the same way to test connectivity to arbitrary ports (`telnet <IP> <port>`), netcat is more versatile in that it enables you to connect via TCP (the default) or UDP.

There are additional functions netcat offers to aid the network troubleshooter. Netcat can be configured to create a hex dump of the session (via the `-o file` option). It can also be configured to execute a program when a connection is made (potentially unsafe if not used carefully), or to restart itself in listen mode after a session is terminated. Netcat can be downloaded from <http://netcat.sourceforge.net/>; the last update was in January of 2004. A Windows version (1.1) can be downloaded from www.vulnwatch.org/netcat/. Finally, a derivative project that adds twofish encryption is known as cryptcat and is available from <http://farm9.org/Cryptcat/>, with versions for both Linux and Windows. There is also a very similar utility called socat, which is being actively developed. Socat is very close to netcat in function but features additional capabilities. Socat can be downloaded from www.dest-unreach.org/socat/.

Tracetcp

Oftentimes, the ability to know the path that network traffic is traversing is key to troubleshooting connectivity issues. In most cases you can determine

this by using the traceroute utility (tracert on Windows systems). When you execute the traceroute command ICMP (Internet Control Message Protocol) is used to transmit packets to the destination with a Time to Live (TTL) value of 1, and this increases for each hop. When everything goes smoothly, each hop has to reduce the TTL by one, and when it becomes zero, the packet is dropped, and a message is sent to the receiver. The problem that often arises is that ICMP is often partially or completely filtered out by intervening routers or firewalls. In this case, you need a way to accomplish the same thing with a protocol that has a higher chance of success.

In these cases, a TCP traceroute can be a life saver. It will effectively do the same thing, by manipulating the TTL values, but it uses a TCP packet and allows a user-configurable port, which almost every firewall and router will allow if it is a well-chosen port. As an example, if you picked a popular Web site and tried a trace route, you may get several instances of “request timed out,” which indicates that the hop is not responding. In most cases this means that ICMP is being filtered by a firewall. If you instead use a TCP-based traceroute utility and specify a destination port of 80, you may get better results. A good TCP-based traceroute utility for Windows is `tracetcp` from <http://tracetcp.sourceforge.net/>. For Linux, a very robust utility is LFT, which stands for “layer four traceroute,” which can be downloaded from <http://pwhois.org/lft/>.

Netstat

You can use the `netstat` utility on Windows or Linux to see a list of network connections. The `-l` option (on Linux) will list only the listening ports for that system. On Windows, the same functionality is provided by using the `-a` option (`-a` works on Linux as well) to list all connections. From a troubleshooting perspective, there may be times when you want to verify that a service is listening on the proper port, or identify what service is listening on a given port. While older versions of `netstat` cannot show you this, modern Windows systems provide the `-b` option, which will list the process associated with a given listening port.

```
C:\>netstat -a -b
```


Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	server:4122	localhost:4123	ESTABLISHED	1988
	[firefox.exe]			

If you are running an older version of Windows, or running Linux, there are utilities to show this. For the command line on Windows NT, XP, or 2000, `fport` from www.foundstone.com/resources/proddesc/fport.htm is very handy. The abbreviated output is shown here.

```
I:\Internet\fport>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com
```

Pid	Process	Port	Proto	Path
1988	firefox	-> 4123	TCP	I:\Internet\Firefox\firefox.exe

For a Windows GUI, `tcpview` is very powerful and can be downloaded from Microsoft at www.microsoft.com/technet/sysinternals/utilities/TcpView.msp. `Tcpview` also has a command-line version called `tcpvcon`. In the Linux world, you can see which processes are using which ports with the `lsof` (list open files) utility. Although the utility can be rather complex, a quick and simple way is:

```
lsof | grep LISTEN
syslog-ng 1555 root 6u IPv4 4463 TCP *:7140 (LISTEN)
```

This will indicate that the `syslog-ng` process is listening for inbound connections on TCP port 7140.

Summary

As you can see, there are many powerful and free solutions for network reporting and troubleshooting available. Never underestimate the power of a visual aid, be it a graph, a pie chart, or whatever. A simple graph can get the point across far more quickly than a page full of numbers. Armed with the relevant data presented in an easy-to-understand format you can demonstrate the need for more Internet bandwidth. You could also demonstrate how the use of Internet streaming music is consuming all the bandwidth, and that with a policy change, a bandwidth upgrade *is not* needed. Depending on your specific policies, you could generate reports for whichever protocols are of interest to you. You could run a sniffer to try to collect and graph statistics on your own, but having free software that will do it for you is a real time saver. Far and away I would recommend PRTG and ntop as the best free solutions for tracking usage statistics.

If, despite your best efforts, problems do arise, there are powerful and free packet sniffing solutions available to help you troubleshoot problems at the network level. Once you have an understanding of the merits of the different products, you can select a sniffing tool that is right for the job. You can quickly see what is happening at the network packet level, and in some cases identify problems that would be virtually impossible to troubleshoot any other way.

Solutions Fast Track

Reporting on Bandwidth Usage and Other Metrics

- ☑ Remember that with any data collection system (that is, sensor), placement is key to ensure that the system can see the proper data to analyze.
- ☑ The type of data collection methods (sniffing, NetFlow, SNMP) you have available to you will have a huge impact on collector placement and on which tools you use.

- ☑ SNMP can be enabled on almost any modern operating system and collect and monitor assorted usage data, including product-specific metrics.

Collecting Data for Analysis

- ☑ Collecting data via sniffing will require that the collector sit inline with the traffic to be analyzed.
- ☑ Collecting data via SNMP enables you to place the collector independently from the devices to be monitored; however, you will only be able to gather data that is specifically supported by the sensor's MIB.
- ☑ Remember that SNMP creates its own network traffic, thus using up bandwidth.
- ☑ NetFlow is more focused than sniffing or SNMP, providing information focused on traffic flows and session information. The downside is that NetFlow is not supported on all devices.

Understanding SNMP

- ☑ Different devices often support different versions of SNMP, and each version has different security capabilities. These considerations could be important depending on how you plan to implement an SNMP infrastructure. SNMP version 3 is the preferred (i.e. most secure) version to use.

Troubleshooting Network Problems

- ☑ One of your first steps is to determine which sniffer is appropriate for the task. Consider factors such as: Do you need a GUI? How much filtering will you need to do? Is installing a driver acceptable or desirable?
- ☑ There is no shortcut to becoming skilled at interpreting the results of a packet capture. It requires in-depth knowledge of the underlying

network protocols, but with some basic understanding you can often find a packet sniffer useful in troubleshooting problems.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: Is there any advantage to using tcpdump over WinDump or vice versa?

A: Not much, the syntax is almost identical, so functionally, it probably doesn't matter. The only factor that would tip the scales in favor of one or the other would be that if you are sniffing a very high-volume network, the Linux system using tcpdump will likely have better performance than a Windows system using WinDump, so you would run less risk of dropping packets. If speed isn't an issue, go with whichever underlying operating system you are most comfortable with.

Q: How do I make ntop do “x”?

A: For starters, do some searching online; odds are good there is documentation out there already. There are also several mailing lists available to discuss ntop and its use. For general ntop use, refer to <http://listmanager.unipi.it/mailman/listinfo/ntop>. For development issues, refer to <http://listmanager.unipi.it/mailman/listinfo/ntop-dev>. Because ntop uses Perl on the back end, with a little Perl programming you can make it do almost anything you want, and much like a batch file, Perl has the capability to call external programs. This means you could easily create a script to regularly check the ntop data and execute a certain program (like send a syslog message or e-mail) when certain thresholds are crossed.

Q: The graphs from MRTG seem very basic, is there any way to modify their appearance?

A: You can edit some of the images used to create the graphs. The most popular method is to configure MRTG to use RRDtool (which stands for Round Robin Database tool) from <http://oss.oetiker.ch/rrdtool/>. This will enable the data collected by MRTG to be stored more efficiently. MRTG supports RRDtool with little additional configuration. A document on configuring MRTG to work with RRDtool can be found at <http://oss.oetiker.ch/mrtg/doc/mrtg-rrd.en.html>. After you have your data in the RRD database, you can use one of the other third-party scripts to generate the HTML graphs, such as the *14all* script from <http://my14all.sourceforge.net/>.

Security as an Ongoing Process

Solutions in this chapter:

- Patch Management
- Change Management
- Antivirus
- Antispyware
- Intrusion Detection Systems
- Vulnerability Scanning
- Penetration Testing
- Policy Review
- Physical Security
- CERT Team

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

If you have successfully implemented a firewall, a network analysis and reporting system, and an intrusion detection system (IDS), I would like to tell you your job is done, and that you can head over to the coffee shop. Sadly, this isn't true. Securing your network and systems is only the first step of an ongoing process. Without consistent and conscious effort, an otherwise secure infrastructure will devolve into an insecure one. This chapter will discuss the various ongoing efforts that are needed to *maintain* a secure environment.

Each topic will be examined with an eye toward the security implications and security risks related to *not* following these procedures. Security is an area of networking where, more than any other, inaction will result in failure. The most secure system today will eventually be the biggest security risk on the network if it is not maintained adequately. Along those lines, almost every system that uses the network for communications, from wireless access points, firewalls, workstations, servers, and even the routers and switches themselves, needs regular maintenance in order to operate reliably and securely. This regular maintenance can be broken into several more specific categories, which we will discuss in detail.

Patch Management

Patch management is a broad category that basically means “keeping everything up to date in a controlled fashion.” A patch is a modification of software that is typically relatively small and specific in what it addresses. The changes resulting from the application of a patch are usually less significant than those of a service pack or a full incremented version release. Patch management is *not* applying all the latest patches the instant they are released to the public. Applying Service Pack 2 for Windows XP will enable the built-in Windows Firewall automatically. This isn't an issue for most users, since the firewall will allow all outbound connections by default, but if you have some service running on that system that needs to listen for an inbound connection, you will suddenly find it doesn't work after applying the patch. In this particular example, the service pack has been out for so long, most people are either already updated or, if not they are aware of the issues. Even if you didn't

know about the service pack enabling the firewall, the information is so widely disseminated that you would likely be able to find the information very quickly. When the patch was first released, however, this wasn't the case, and many organizations suffered a service disruption that could have been avoided with proper patch management.

A proper patch management system should include a regularly scheduled review of the patch levels of all systems (by the appropriate subject matter experts). The particular schedule you choose will be based on your environment. In a high-security environment, this process may occur as often as weekly, while in many environments a monthly process is adequate. A key component of the patch review will be to assign a priority to newly released patches. This should be done by an appropriate subject matter expert, or, if you have the luxury, a group of them. This is a relatively subjective art with no hard and fast rules, only guidelines. If a patch addresses a severe security flaw in hardware the company has deployed, it would be a very high-priority patch. If, on the other hand, the effected hardware was only deployed in a few places, and there were additional controls in place, the priority would be much lower. The priority that is assigned will dictate the deadlines for a patch to be applied. These deadlines will also need to be developed to meet your business needs but the following priorities could serve as a starting point.

- **Critical** These types of patches are for serious vulnerabilities that are both easy to exploit and likely to be exploited. These patches should be tested and applied as soon as possible, with a 24-hour deadline being common.
- **High** These high-risk patches will need to be applied with some urgency on an *accelerated testing schedule*. These patches could have a one-week deadline for implementation.
- **Medium** This is a typical vulnerability. Maybe it requires a very detailed set of circumstances to exploit, or the risk is a minor denial of service, but not a complete compromise of the affected device. The deadline for these types of patches would be your *standard patch cycle*, possibly monthly.

- **Low** Patches that are low priority may be for minor vulnerabilities that are nearly impossible to actually exploit, or for devices that are very well secured with additional mitigating controls. While relevant to business, these types of patches may wait until the normal *maintenance cycle* to be applied. The maintenance cycle would mean the patch would be applied indirectly with the next software upgrade. It's worth noting that these patches are still tracked as outstanding until they can be remediated.
- **Negligible** These patches do not apply to your particular hardware or configuration. These patches are generally filtered out when they are assigned their initial priority and are not tracked after that point.

The patch review meetings will likely include some type of reporting to be distributed to upper management, in a format that is easy to digest and understand so that everyone involved can be up to date and informed as to what the current status is. These reports should include a list of patches that are outstanding, schedules for applying the patches, and an indication of the success or failure of past patching efforts. This type of documentation will also be of value to auditors to show that there are processes in place and being followed for keeping the systems secure and their software up-to-date.

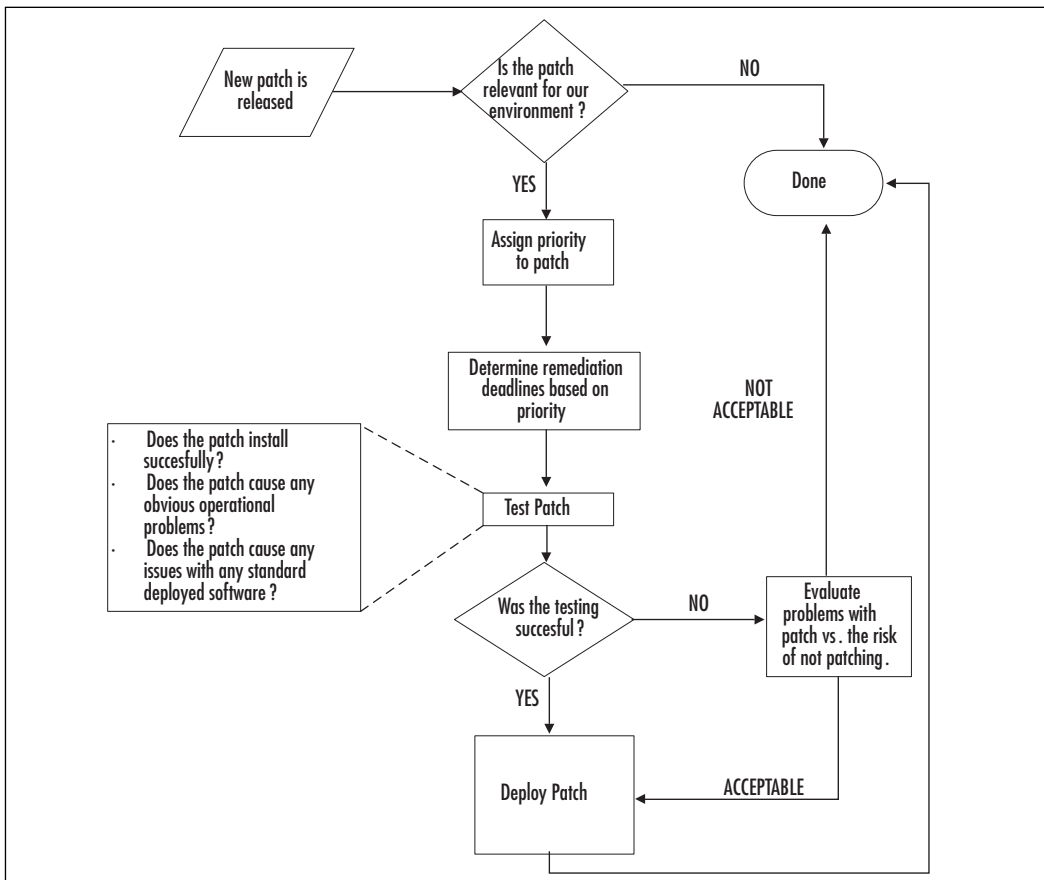
A vital component to any patch management system is testing. It would even be reasonable to say testing was the *core* of patch management. Without a procedure in place to test prospective patches and ensure compatibility with the existing hardware and software, you really only have patch scheduling, which is not adequate. The testing procedure should make use of some standardized forms, with sign-off by the person who performed the testing, and with as much detail as is reasonable, a description of the testing that was performed. It is important to capture and document the testing process in an effort to increase accountability and consistency. This will help you avoid multiple people all believing someone else tested a particular patch.

Sometimes a patch will not make it through testing successfully. This may be by design; for example, if that patch disables some feature you were using. In these cases you may need to evaluate whether or not to apply the patch anyway, even though there were issues discovered during the testing process. This will basically be a judgment call on whether the problems introduced by

the patch outweigh the risk posed by not implementing the patch. These types of decisions will need to be made, or at least approved, by the appropriate business representatives. The person who makes such a decision needs to have the appropriate authority for such a risk acceptance.

Don't forget that the testing procedures themselves will need to be regularly reviewed to ensure that the testing that is performed is still adequate. As software and features change, you will need to modify the testing procedures to accommodate the new functionality. While all of these considerations are applicable regardless of *what* you are patching, there are some considerations that are specific to the platform in question. Your individual business needs will dictate the specific procedures you employ. Figure 8.1 shows a way these procedures can be implemented and incorporated into a repeatable process.

Figure 8.1 Patch Management Process



Network Infrastructure Devices

The network infrastructure includes anything that is *part* of the network rather than being *on* the network, or *using* the network. These are the devices that move data through the network and include routers, switches, firewalls, and bridges. All these devices will require patching sooner or later. One of the biggest considerations you have to work around when developing your patch management procedures is the patch release schedule. Obviously, you can't patch a system if the patch isn't out yet. In the case of network devices, particularly firewalls and other security devices, patches for security vulnerabilities are typically released very quickly and frequently. This means your patch management system will need to include a process to schedule and prioritize the testing and application of the patches. Having a set schedule for patching, for example monthly, at least for the noncritical patches, enables the business units to schedule around the patching windows and account for any outages. By distributing a patching schedule, you help minimize the impact of your patching efforts. If you wait a week to patch a critical hole in your Internet-facing firewall, you are gambling that the hackers don't find your vulnerable firewall in the next week; on the other hand, if the patch is inadequately tested, you could create a service disruption if the patch causes any unexpected problems.

Always keep in mind that when applying patches to infrastructure devices the potential for service disruption is high. Because all your other devices rely on the underlying network infrastructure for communication, complications at the network level can have devastating consequences. If a single server patch causes the service to quit functioning, you are without the services of that one server. If the same thing happens to a core router for the network, it's possible that no network systems will be able to function. Because of the high potential for service disruption and the large scope of potential impact, patches to infrastructure devices should be tested *extra* thoroughly before being deployed. Although the risk that the patch will break something is generally low, you want to put particular emphasis on the testing of any optional or less-common features you may be using.

Operating System Patches

Operating system patches are released at different times by different vendors. In most cases, a patch will be released to address a specific security risk or to correct a bug related to particular functionality. Some vendors may have a specific schedule that they release their patches on, such as Microsoft, which releases their patches on the second Tuesday of each month. It is these release schedules and release frequencies that are the primary differences between infrastructure patches and operating system patches. Your timelines for patch review, application, and testing cycles will need to be synchronized with the vendor's release schedule. You will also need to schedule your patch updates with your change windows (see below for change management), perhaps even devoting one change window to OS patches if the volume is significant.

The processes around applying patches and the time frames for doing so also need to account for the role of the host system. The people responsible for prioritizing when patches get applied must be aware of the criticality of the services the network host provides. A server that is critical to the business might need to wait longer to receive a patch during a monthly maintenance window, while a host that is exposed to hostile attacks (such as in a DMZ) might need to receive a security patch much more quickly. These are the types of considerations that someone who is not familiar with your business operations could not take into consideration when developing a patch management strategy. To schedule these maintenance activities properly, you will need a technical understanding of the impact of the patch, be able to weigh the risk of not applying the patch, and understand the potential business impact of a service disruption.

Application Patches

Application patches are typically released by the application developer. The release frequency can vary drastically. Some applications almost never have updates, while others change frequently. Most often, the vendor will recommend you only apply patches to address functionality if you need that functionality or are experiencing an issue; in other words, “don't fix it if it isn't broken.” Because the nature of an application patch can vary so dramatically, these will need to be evaluated on a case-by-case basis in order to schedule

the patches for deployment. Unless the patch is to address a security risk, these types of updates can often wait for a regular change window to be applied. As with operating system patches, the role of the application in question may impact the scheduling and testing of the patch in question.

Change Management

Although change management and patch management are related, they are not the same thing. Patch management is a subset of change management and refers to applying patches specifically. Change management is much broader in that it encompasses any type of change to the existing environment. The change that is being managed could include the installation of new software or hardware, or a modification of settings with a system to alter the functionality. The objective of change management is to handle changes in a controlled fashion that minimizes risk. There are many things that can go wrong when implementing a change; however, a proper change management strategy can help mitigate many of those risks.

Change Causes Disruption

The biggest and most obvious risk when implementing change is that it may cause a service disruption. We are all probably familiar with this type of complication arising from a change. We applied the update and suddenly the server was unreachable. This could be a total loss of functionality, such as when the network drivers get corrupted, or a partial disruption, where only a portion of functionality is lost. In either case, the downtime can cost significant loss of revenue, or real dollars (paid to consultants or vendors) to correct, not to mention the headache involved in troubleshooting the problem.

Remember that *some* of these risks are entirely avoidable, while others are not. There really is no way to *know* if a network interface card on the server is going to quit working when you reboot the server, but having a spare one on hand can really help. In some cases, the changes themselves may work perfectly, but due to a lack of proper change management, multiple changes occurring simultaneously could interfere with each other. For example, if you are upgrading the Internet connection, another technician may not be able to perform an upgrade using software that is downloaded over the Internet.

Inadequate Documentation Can Exacerbate Problems

There are still more ways an inadequate change management system can cause problems. Imagine a co-worker installs a new switch, tests the connectivity, and then heads home. The next day, an entire group of systems become unavailable. You suspect that all the systems might be connected to the new switch but there is no documentation of the change. You don't know which systems are using the new switch, or what ports they are connected to. You now have to spend precious time trying to learn what the current environment is before you can even begin to troubleshoot the real issue. This is an example of how the change itself might not be the cause of an outage, but without proper documentation and procedures in place, changes can still negatively impact the functioning of the network. To mitigate these risks, the best practice is to develop a change management strategy.

Change Management Strategy

So now that we know that change management is a necessity, how exactly does one go about implementing a change management strategy? A good change management strategy will provide you with the tools that are needed to implement and manage change effectively and safely. Your change management strategy should include the following elements:

- Change Review
- Change Scheduling
- Change Documentation

The first step to implementing changes in a controlled fashion is to review and assess the changes that are needed. In some cases, this may be a relatively informal process conducted by a single individual. In a larger organization, this could actually be implemented by a committee and include formal meeting minutes that are made available to upper management. You will need to find a level of sophistication that suits the needs of your organization. The purpose of this phase is to identify the changes that are needed, determine how badly they are needed, and determine what level of process

needs to be used for their implementation. You probably don't want hours of reporting and meetings because someone wants to change their screensaver. The changes need be prioritized, which is somewhat subjective but includes consideration such as the following:

- The impact and risk of *not* implementing the change. Will business be impacted? Will real dollars be lost or will it only be an inconvenience for employees performing some processes?
- The risk that applying the change might negatively impact service. What are the odds of this change breaking something? How complex is the environment? How well documented and understood is the environment?
- The criticality of the host/system affected by the change. How key is the affected device to the business process?
- Nontechnical considerations. If the change breaks something, will the disruption be visible to the customer? Will there be a loss of customer confidence or business?

All these considerations will lead to the second phase, which is scheduling. I have seen many examples where the risk of making a change was simply too high, and the decision was made to simply live with the risk. Your prioritization and deadlines can probably be very similar to that of the aforementioned patch management process.

Only after you have analyzed the priority of the changes can you schedule them. Typically, a business will have more than one change window, depending on the urgency of the change. Minor changes to noncritical systems may be acceptable during any nonbusiness hours, while critical systems may only have a change window of a couple of hours each week to apply changes. A *change window* is the accepted time frame for implementing changes. This phase is best accomplished with multiple stakeholders working together. This allows for various initiatives to be scheduled without creating a conflict between them. The scheduling stage will work best if you can provide input and buy-in from representatives from multiple lines of business.

Scheduling could also be impacted by nontechnical considerations that the technical staff may not be aware of. Perhaps a new advertising campaign is

being initiated on the first of the month and traffic to the corporate Web site is expected to rise sharply. In this case, applying an operating system patch on the server the night before might not be a good idea. It is for these reasons that the person scheduling the changes should not work in a vacuum, but instead seek cooperation from the appropriate parties. In a smaller environment, these types of scheduling conflicts are less likely to occur, but you should be aware of the potential and take steps to avoid it. By having all the critical groups represented during the scheduling phase, or at the very least informed of the schedule, you can minimize the risk of a scheduling conflict.

Documenting changes is one of the most commonly overlooked or inadequately performed steps. This is especially true when in the middle of a crisis and restoring functionality is the highest concern. A well-documented change is far more likely to be successful than one that is not. This is true not only because someone may need the documentation later, but also because the process of completing the required documentation may cause one to consider some facet that may have otherwise been overlooked. Although change documentation will be different for each organization, there are some elements that all change documentation should have in common. These are outlined here:

- **Change Schedule** The change schedule will have the time and date the change will be implemented, including when the changes will begin and end. It should also include a back-out time, which is a time after which the change either must work, or the change is undone. To continue working on an unsuccessful change after the back-out time typically requires senior management approval.
- **Testing** This section should include the procedures used to test the change. In some cases this may be simple; in others it may be a very detailed section requiring its own set of documentation. This section explicitly states the manner in which you will measure success of the change. This section also serves to show that the person implementing the change performed their due diligence in testing the change, should a problem arise later.
- **Backout Plan** This section will detail what steps will be needed to reverse the changes. In some cases this section may seem trivial, while

in a more complicated change it may be very difficult to undo the changes. For example, if you upgrade a third-party application on a Web server, there may be no means to undo the upgrade other than to restore from tape, which will likely take a considerable amount of time. The steps involved in the back-out plan will be used to determine when the back-out threshold should be. For example, if your change window for Web servers is 1:00 A.M. to 6:00 A.M. and it will take three hours to back out the changes (by restoring the server from tape), your back-out threshold would be no later than 3:00 A.M.

- **Contact List** The contact list is a listing of all relevant stakeholders. This includes who will be making the changes, who will be testing the changes, and who they report to. This list should also include who should be contacted in the event of an emergency, and possibly the project manager for the effort. It is a good idea to also note the expected means of communication. For example, if the senior network manager expects an e-mail in his inbox after the change is successful, this should be noted as well. This helps ensure that these details are not forgotten.
- **Sign-off** This is one of the key sections. Not only subject matter experts but also stakeholders should provide their written approval. This could include line-of-business representatives who are not technical as well. This provides assurance that everyone who may be impacted has been represented and is in agreement with the change.

All of these procedures should have a plan for dealing with emergency changes. By definition, an emergency change is a needed change that was not planned for. This often means adequate testing and planning has not been done but the risk still may be acceptable if the alternative is less desirable. These changes can occur when a system is unreachable unexpectedly, or if you are actively being attacked by a hacker. There should be alternate procedures for implementing a change that enable one to expedite the process. Emergency change procedures should be well documented. Special consideration should be given with regards to how one seeks approval for an emergency change and who has the authority to approve emergency changes.

With well-established change procedures in place *before* implementing a change, you can minimize the risk caused by implementing emergency changes. If a service disruption does occur, you will also be more prepared for dealing with it in a timely fashion.

Antivirus

In modern times, an antivirus system of one type or another is a requirement for doing business. These can be implemented on a central gateway such as a mail server, or locally on individual workstations (or more commonly, both). In any case, the antivirus software will need frequent updates to keep the virus signature database current. Because the signature file is used to identify a virus based on a small code sample, and given the rapid development of new viruses and Trojans, an out-of-date signature file is close to not having any antivirus protection at all. In most cases, new signature files will be released very frequently and generally cause little risk of a service disruption. A schedule for signature file updates should be established and approved by management. Most commonly, the complications from a new signature file will be an improper identification of a file as malicious when it isn't, also known as a *false positive*. It is important that reports be generated to verify that the signatures are current on all systems, and that someone regularly reviews these reports to take action if that is not the case. If automated updates have been failing on a server for several months, there needs to be a process in place that will ensure someone notices that fact. While testing tends to be minimal when it comes to antivirus signatures, it is always a good idea to try a pilot deployment on a small subset of hosts before deploying any significant updates across the entire company. This is particularly true if the scanning engine itself is being updated or a new antivirus option is being implemented. You wouldn't want your antivirus product to detect your corporate accounting program as a virus and "clean" it for you.

Antispyware

Antispyware software may be integrated with your antivirus software, or it may be its own independent product. In the latter case it will need to be updated, and all the same caveats that apply to antivirus will also pertain to

independent antispymware products. Because antispymware or malware products typically use signatures, like an antivirus program, these will need to be updated regularly. It is also worth pointing out that antispymware products are prone to a lot more false positives, particularly when it comes to network-related utilities. Because an average user doesn't use these networking utilities, they are often flagged as malware. You will need to take this into account and provide some means to review the scan results, and preferably form a white list of approved users and/or products in order to reduce the false positives.

Intrusion Detection Systems

Much of the same considerations for an antivirus solution apply for an IDS or intrusion prevention system (IPS) as well, at least if it's signature based, such as Snort. Because the IDS/IPS uses a known sequence to identify hostile traffic, keeping the signatures current is vital. Typically, an IDS will alert you only of an intrusion, so there is little risk of failed or inadequate maintenance causing the IDS to result in a service disruption for other network traffic. These same guidelines are not true if you are using an IPS, which will take *active* measures to stop an intrusion. In the case of an IPS, it is critical that changes are thoroughly tested so that the IPS doesn't inadvertently identify critical business traffic as hostile and terminate the connection. Most maintenance will be related to either updating the IDS/IPS application software itself, or the attack signatures. In all cases, adequate testing of the changes will help ensure that the changes are successful and do not adversely affect normal operations.

Vulnerability Scanning

Vulnerability scanning and penetration testing are often (incorrectly) used interchangeably. Vulnerability scanning is typically a passive activity. You are simply using automated software to check hosts for known vulnerabilities or risky (that is, insecure) settings. This is not a penetration test. A penetration test is an active attempt to compromise the security measures that are in place. Vulnerability scanning is not a substitute for a penetration test, but it is useful for determining the baseline of your security risks for a given system. Because the network is dynamic and ever changing, the vulnerability scanning process is never really done. The frequency of vulnerability scans will be

determined by corporate policy, with consideration being given for the risk and sensitivity of the host in question. In the case of a very large network, it might take months for a complete scan of the network to finish, but in a small environment, a weekly scan is usually easy to implement.

Updates to the scanning engine will occasionally need to be made. More commonly will be updates to the actual attack scripts that the scanner uses. Some scanners call these scripts plug-ins, while others call them scripts. This is what tells the scanning software *how* to check for a given vulnerability. If these are not current, you may not be checking for the latest risks, and thus might believe you are secure when in fact you are not. Because vulnerability scanners carry an especially high risk of causing a service disruption on the target host, any updates or changes to the scanner or the attack scripts should be thoroughly tested before being used on a large number of hosts.



WARNING

You may have noticed an emphasis on conducting a vulnerability scan carefully. This concept is repeated for good reason. Be sure to evaluate which tests are needed carefully before scanning a host and preferably scan a test system before targeting the entire network. Some of the tests can carry a very high risk of causing a service disruption. After a given set of tests has been run several times without causing any problems, the risk will be minimal; however, if changes are made to the environment, you once again are running a high risk of causing a service disruption.

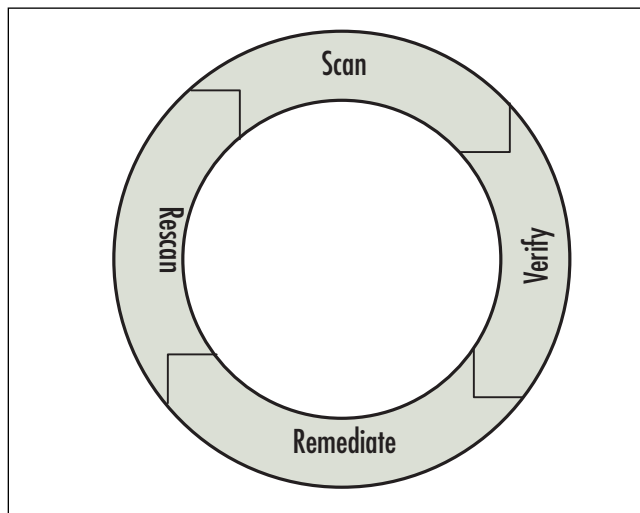
Vulnerability Management Cycle

Because vulnerability scanning must be repeated on a regular basis, you must develop appropriate policies and procedures to accommodate this. These should include a schedule for when, and under what circumstances, a given host should be scanned. There should be a process for the appropriate parties to sign off when a system is deemed “clean,” as well as a procedure to accept a given risk in cases where it is undesirable or impossible to remediate at this time. A typical vulnerability scanning cycle would consist of the following steps (see Figure 8.2):

1. **Perform the Initial Scan** This gives you a baseline of issues and is used in subsequent steps.
2. **Verify the Scan Results** Some of the issues a vulnerability scanner finds may be false positives. In other cases they may be legitimate issues, but the issues may not be relevant because of other compensating controls the scanner cannot account for.
3. **Remediate Valid Issues** This will include following all your change control procedures in an attempt to remove or mitigate the risk that the vulnerability scanner found.
4. **Rescan** This last step is often skipped but it is important. This not only verifies that your remediation steps were successful, but demonstrates to management and, if needed, to auditors, that you are actively taking steps to safeguard your network.

This cyclical process will be never ending in that once you have scanned a host, verified the scan results, performed remediation for any issues, and then rescanned the host to verify your remediation efforts, it will be time to begin the process again.

Figure 8.2 The Vulnerability Scanning Cycle



Roles and Responsibilities

As in all areas of network security, roles and responsibilities are important. This is even more so when it comes to vulnerability scanning and remediation. The roles of individuals should be clearly documented and defined, with signoff from management at the most senior level. This is for several reasons. For one, you want to ensure that when John says it's okay to leave that drive share without any authentication, John is really *qualified* and *authorized* to speak on behalf of the business unit in question. You need to know that he understands the risks, and is able to be held accountable for any repercussions that may arise. Another reason to clearly document these roles and responsibilities is to demonstrate that there are no conflicts of interest occurring. If John is the one who will have to stay up during the maintenance window all weekend to configure the servers, it may not be appropriate for the same person to say the servers don't need to be reconfigured.

Penetration Testing

Penetration testing carries many of the same concerns as vulnerability scanning but will often be performed by a third party. This is because true penetration testing requires individuals with a high level of skill and knowledge. Few businesses have the workload or budget to have such staff full time. Instead it is very common to pay a trusted third party for penetration services. The costs of such services can be considerable. Because of this, penetration testing is typically only performed on high-risk and/or high-criticality hosts and systems. Penetration testing when a new infrastructure is deployed isn't really maintenance, but subsequent penetration tests will fall into that category. The criteria whereby a system/host will need to be penetration tested again should be clearly defined. In most cases the criteria will be on a scheduled basis (such as annually), or when significant changes have occurred, whichever comes first. You should work with your legal counsel to develop a consent form that will protect you from accidental service disruptions.

Obtaining the Support of Senior Management

Due to the nature of the checks, vulnerability scanning and especially penetration testing carry a high risk of causing a service disruption. There are just too many components that must work properly for everything to go smoothly. All it takes is any single process running on the target host to stop working and the system could come to a crashing halt. A little research and you can find countless stories of the overworked network admin who is testing something in an effort to help point out a security risk, only to lose his job for “hacking” the company resources. No matter how well your intentions may be, if you use Nessus to run a vulnerability scan against the server running the company Web site and it crashes the Web site, you will surely be in for some grief. It is for this reason that you must obtain approval from the most senior management possible before undertaking any of these high-risk activities. This approval should be in writing and signed by someone with the appropriate authority to do so. This may be the only thing you have to act as proof that your activities were authorized and legitimate.

Clarify What You Are Buying

The Latin phrase *caveat emptor* means “let the buyer beware.” When it comes to penetration testing this is especially true. There are a great many security consultants out there who will perform a vulnerability scan using Nessus (for free), print out the results, and hand the customer the report along with a bill. This is not a penetration test. A true penetration test requires someone who has the skill set to be a “hacker” but chooses to make a legal living out of security testing. You can get a lot more value out of your penetration testing by taking the time to sit down and talk with your prospective provider of such services. Ask for references, do your research and outline exactly what services they will be providing, in writing, prior to signing on the dotted line. There is nothing wrong with running a vulnerability scan, and there is nothing wrong if you want an expert to run the scan for you. You just need to be sure you are buying what you think you are buying. Being an educated consumer will help you obtain a better value with security services and help your organization be more secure.

Policy Review

You may be tired of all this maintenance, but believe it or not, even your policies need regular maintenance. As technologies change so must your policies. A new type of encryption could become available, or an established encryption algorithm could be rendered ineffective due to some new attack vector. This type of maintenance will include *policies*, *standards*, and *procedures* documents. Procedures are usually pretty easy to identify, but unfortunately, you will often see policies and standards used interchangeably. The following bullet points outline each of these types of documentation:

- **Policies** These high-level documents do not contain details, but instead state a broad objective or requirement. A common example would be something like, “All data classified as highly confidential must be encrypted.” These documents are generated by senior staff and change infrequently. Changes to policy are usually prompted by business factors.
- **Standards** These documents represent the more technical requirements that are needed to meet policy. For example, a standard might state that “when data encryption is required by policy, it must be done using 3DES or AES encryption only.” These documents will change as the relevant technologies being used change.
- **Procedures** These documents contain the detailed, step-by-step instructions to be used for policy and standards compliance. They will specify exactly what utility to use, which radio buttons to enable, and what boxes to check. These documents may change often, depending on the technologies used. An update to a software product may completely change the procedures you have established. Different procedures may assume a different level of expertise on the part of the person performing the procedure, but the general rule is to make the documentation as easy to follow as possible.

Each of these documents should be reviewed on a scheduled basis to ensure that they are still adequately addressing the security needs of the organization. After being reviewed and, if needed, modified, they should be submitted

to the appropriate parties for peer review and critique by the subject matter experts. When considered final, they should require written approval by senior management. In this way you can ensure that your policies don't become outdated and lose their value. As with most changes, there should be some mechanism in place for implementing changes to policy and standards in an expedited fashion in the event of some emergency.

Physical Security

In many instances, physical security is overlooked, especially in a smaller organization where such concerns are often considered too unlikely to worry about. Physical security can include any security risks related to your environment. As an example, if someone were to suggest that you place your servers in your front lobby, you would probably consider them insane. However, I have seen many small businesses where the network equipment was in a closet with the door consistently propped open to provide ventilation. This is fine if you can ensure only authorized individuals have access to the area the closet is in, but this is rarely the case. Cleaning personnel and other support staff often have unrestricted access to sensitive systems. It may sound far-fetched, but if one of those personnel had the know-how, how hard would it be for him or her to get to your most sensitive data? What if, instead, a bucket of cleaning water were accidentally dumped on the server that was key to your business?

If you give it some careful thought, you may realize physical security is weak or nonexistent. The physical security of a site should be regularly reviewed and any weaknesses addressed. In some cases, the solution may be a simple change of process or policy; in others, some new equipment may be needed. When reviewing the physical security, consider whether or not someone can gain unauthorized physical access to networking components. Is authorized access logged or otherwise documented? Are sensitive materials being handled appropriately in all mediums? Are environmental risks being considered such as flood, fire, or temperature extremes? Are employees educated to speak up if they see something suspicious?

Some physical security risks do not involve door locks or security fences. Most high-security environments mandate that any network ports that are not

in use should be disabled. This is related to physical security because the intent is to prevent someone from plugging in an unauthorized device, such as a sniffer, and having access to the network. This concept of port access as a physical security concern can be extended to any type of data port, writable CD-ROM drives, infrared data ports, even to USB ports on the workstations. Any of these can offer an unauthorized individual access to data.

Notes from the Underground...

Know Where Your Sensitive Data Is (at All Times)

If you are trying to figure out what to secure from the view of “is this server sensitive,” you are inviting problems. The correct approach is to know what data is sensitive (meaning classify all your data), and then know where that data resides at all points in its lifecycle. As an example, a business calls you, as your customer, and places a large order. They fax you their relevant tax ID and any other needed forms. You get their payment information verbally over the phone, process the order, and e-mail them a confirmation receipt.

You know you need to store their payment information securely so you encrypt the account details on the server’s hard drive. The e-mail receipt itself may need to be encrypted depending on what information it contains, so you take that into account as well. When you’re on the phone you verify whom you are speaking to using existing account information when possible. So are all your bases covered? Is the fax machine, which contains confidential information, in a physically secured location or can unauthorized people walk by and see the faxed forms? Are the server backup tapes secured? Depending on the encryption mechanism, if you can restore the entire server from tape, you might be able to access the encrypted data.

The point here is to know where the sensitive data is at all times and to protect it accordingly.

CERT Team

When the network is impacted by a crisis, you need a group of highly knowledgeable and skilled individuals who can react to the crisis and direct the response efforts. The crisis could take any number of forms, from events such as a power outage, a virus outbreak, or a denial of service attack. This group is typically called a *Computer Emergency Response Team* (CERT) or a *Computer Incident Response Team* (CIRT). The size of this group could vary depending on the size and needs of your organization. The CERT team should meet regularly and review any outstanding security issues. This group will monitor the various security notification lists so that the organization can be aware of security issues in a timely fashion. When an issue does arise, the CERT team will meet and make an assessment of the threat and then call in the appropriate resources to deal with it. This could simply mean calling in the “IT guy,” notifying a law enforcement agency, or flying out a security expert to address the issue.

Because the CERT team is responsible for halting or minimizing the damage from an emergency, it is typically backed by the highest level of management. This backing generally includes having access to significant resources if that is what is necessary to get the job done. While a small organization might not have a group of people make up the CERT team per se, you do want to consider taking some steps to be prepared. There should be a plan in place so that people know who to call if something goes wrong. The first contact person should be someone who can make a judgment call about technical issues and understand how severe the emergency truly is. Employees should be educated so that they can make an educated determination of when a computer emergency is occurring.

The following is a list of some of the most widely used notification services. You may want to consider signing up for some of these so you can stay ahead of the information curve. While some offer alternate means of notification, the most common one is a mailing list, in which case you will receive all the e-mails anyone sends to the list. Sorting through the e-mails can consume some time, but if you really want to be ahead of the game it's worth it. You can be sure the hackers are using these services so they know about the new vulnerabilities as soon as possible. They are often counting on knowing about them before the security administrator for a company does.

- **Bugtraq** This is probably the largest and most well-known security mailing list. The volume can be high, but it is the cutting edge of security news. This list is not product specific, so much of it may not be relevant to your environment. Because this is a *moderated* list, it keeps the spam to a minimum.
www.securityfocus.com/archive/1/description
- **Microsoft Technical Security Notification Services** This service offers various levels of information using multiple channels (e-mail, RSS, and even instant messenger alerts). The scope of the notifications is of course only Microsoft products.
www.microsoft.com/technet/security/bulletin/notify.mspx
- **Full Disclosure** This list is very similar to bugtraq in that it is not product specific, but this is an *unmoderated* list, which means news will hit slightly faster. The price one pays for this extra speed is that a lot of garbage gets on the list as well and not all the reported vulnerabilities are well tested or reliable.
<https://lists.grok.org.uk/mailman/listinfo/full-disclosure>
- **Security Basics** This is a newbie-friendly list (many are not), so asking basic questions is accepted here. This is a good list for those just getting into network security. The list is *moderated* and fairly high volume. www.securityfocus.com/archive/105/description
- **Firewall Wizards** This is a very tightly *moderated* list focused on firewalls of all types. The quality of content is fairly high, so if administering a firewall is something you're interested in I highly recommend this list. <https://honor.trusecure.com/mailman/listinfo/firewall-wizards>

There are many more lists out there. Some are generic and contain information related to any type of security issue, while others are very product specific. With a little searching you can find several that can serve not only as an early warning system for security-related news, but also as a sounding board to help you make sense of tough security questions that may arise.

Summary

A secure network will stay secure only with proper maintenance. This maintenance needs to occur at many levels, such as patching software, upgrading hardware, and keeping IDS and antivirus signatures up to date. All of these upgrades and patches will need to conform to a change management process, which is in place to ensure that changes happen in a controlled fashion, designed to minimize risk. Some forethought should be applied to develop an appropriate response plan should a computer emergency occur. Remember, your network is much like a chain in that it is only as strong as the weakest link. If any one of these areas is neglected, you may be creating a security risk. As the network administrator, you have to secure all these avenues, whereas a hacker only needs to find one unpatched or out-of-date system to compromise your security.

Solutions Fast Track

Patch Management

- ☑ Virtually all network devices will need to be patched at some time or another. When patches become available, evaluate if they are needed at this time or if you can wait for a bundled (such as a service pack or new kernel) release that has had time to undergo more rigorous testing.
- ☑ Apply extra caution when patching core components whose malfunction could mean a loss of all network functionality.

Change Management

- ☑ All change carries some risk; change management is aimed at understanding and minimizing that risk.
- ☑ Documenting changes is key to understanding your environment, which in turn lends itself to being able to troubleshoot the environment in the event of a service disruption.

Antivirus

- ☑ If you don't keep your antivirus signatures up-to-date, you might as well not have any antivirus protection at all. Maintenance is absolutely vital for signature-based products.

Antispyware

- ☑ Antispyware software may be integrated with your antivirus software, or it may be its own independent product.
- ☑ Because antispyware or malware products typically use signatures, like an antivirus program, these will need to be updated regularly.

Intrusion Detection Systems

- ☑ Keeping signatures up-to-date on a signature-based IDS is critical.
- ☑ Testing any IDS is key to preventing false positives, and in the case of an IPS, adequate testing will help you avoid creating a computer emergency using the IPS.

Vulnerability Scanning

- ☑ Vulnerability scanning must be a regular, cyclical process. The task is never done unless you never make *any* change to your systems.

Penetration Testing

- ☑ Make sure there are no ambiguities, so that you know exactly what services you are getting for the agreed-upon fee.

Policy Review

- ☑ Your IT policies (and standards/procedures) will need regularly scheduled reviews, and, if needed, modifications to maintain their value and applicability.

Physical Security

- ☑ Physical security can include any security risks related to your environment.
- ☑ Most high-security environments mandate that any network ports that are not in use should be disabled. This prevents unauthorized users from inadvertently using the data port to attack the system, load malicious software, or transfer sensitive/confidential data.

CERT Team

- ☑ You should have a procedure in place to respond to computer-related emergencies, including contact procedures so that the initial discovery can be escalated properly.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: How much information should penetration testers have about the target before they begin?

A: This varies. Often, the testers will begin with no information and have to gather it on their own. This would simulate what an external attacker could come up with. After an initial round of penetration testing using only the data that was gathered by the testers, follow-up testing is sometimes performed after providing the testers with full information regarding the architecture and processes in place. This second stage is to simulate the types of compromises an inside attacker who was knowledgeable about the target environment may attempt. In this way, the first penetration attempts would have a different set of obstacles. The filters on your firewalls are likely to be very different if you are coming in from the Internet

than if you are internal to the organization. Although penetration testing isn't always done this way, it does represent a very thorough way to test.

Q: Should we tell other staff about a penetration test, which may “warn” employees to be extra vigilant?

A: Again, this is often done both ways, with the initial intrusions serving the dual purpose of testing your security *and* testing your IDSes and security response. After the initial responses, everyone will know the penetration test is underway and any pretense can be dropped, with security staff only monitoring the activities and likely providing reporting of what activity was detected.

Q: Do all these roles need to be different people or can they all be done by the same person?

A: The best guideline is to try to avoid any potential conflict of interest. If the person who is approving the server patches is the person who is applying the server patches, he or she might have a reason to “approve” applying them at lunch, rather than waiting until the weekend so that he or she doesn't need to come into work on the weekend. It probably won't be possible to avoid all potential conflicts of interest in a smaller organization. In instances where it cannot be avoided, try to seek additional management signoff so that at least an additional set of eyes have approved the maintenance activity, and make sure all maintenance is well documented.

Q: Where is the end of the vulnerability scanning cycle?

A: There is no “end.” After the vulnerability scanning cycle has been implemented, steps 2 through 4 will be repeated forever.

Index

A

access

- administrative, permitting, 72
- controls, 137
- file-level controls, 147–152
- mandatory and discretionary controls, 173
- remote, 72, 86–130
- user, group rights, 142–147

Access Control Lists. *See* ACLs

access points (wireless), security of, 388

access request forms, 362–364

AccessChk utility, 151–152

AccessEnum utility, 150

account lockout policies, 159–160

account logon events, auditing, 269

accounts, renaming Administrator, 142–144

ACID (Analysis Console for Intrusion Databases), 246, 256

ACLs (Access Control Lists)

- Cisco example, 40
- and Linux firewall rules, 36, 40–41
- and screened subnets, 28

active directory events, auditing, 269

administration

- NC client warnings, 124

permitting administrative access, netfilter, 72

Administrator account, renaming, 142–144

AES (Advanced Encryption Standard), 424

alert groups, configuring (BASE), 253

alerts

configuring Snort to send e-mail, 261

PRTG Traffic Grapher, 411

algorithms, encryption hash, 336

analysis, collecting data for, 392–394

Analysis Console for Intrusion Databases (ACID), 246, 256

analyzing

syslog logs, 312–327

Windows event logs, 277–279

Angry IP Scanner, 351–352

anomaly-based IDSs, 217

antennas, wireless, 359

antispymware software, 188–201, 211, 212, 459–460

antivirus protection, providing, 188–201, 211, 212, 458

application gateways, 26

application patches, 453–454

architecture, firewall, 26–31

assignment of user rights

Linux, 165–168

Windows, 160–163

auditing

- documentation and, 363
- policies, 160
- and system hardening, 138–139

authentication, 137–138, 156

authorization, and information access, 138

Automatic Updates (Windows), 177–179

B

back doors, and system compromise, 34

Back Oriface Trojan (BOPing), 355

bandwidth, reporting on usage, 392–394, 442

Barnyard (Snort add-on), 256

BASE (Basic Analysis and Security Engine), running with Snort, 246–254

baselines, checking for Microsoft products, 379–380

Basic Analysis and Security Engine (BASE), 246–254

Bastille hardening script, 172–173

Bird, Tina, 331–332

Blat.exe SMTP program, 320

Bmail SMTP program, 320

Bugtraq security mailing list, 469

business continuity (BC) and disaster recovery (DR) plans, 365

C

CA certificates, OpenVPN's support, 100

CD-ROM, installing Linux from, 33–36

CERT (Computer Emergency Response Team), 468–469

Certificates of Authenticity (CA), 100

cfgmaker, 398

chain of custody, ensuring, 328

chains in Linux firewalls, 36–39, 44

change management, 454–459, 470–471

checkpoint firewalls, 32

CIRT (Computer Incident Response Team), 468–469

Cisco PX firewalls, 32, 40

Clam AntiVirus program, 189–195

classifying data, 135–136

climate controls, costs of, 5–6

commands

- See also specific command*
- iptables, 46, 52–56
- Linux user and group administration, 167–168

commercial security solutions vs. free solutions, 8–16

communication, security testing, 384

Computer Emergency Response Team (CERT), 468–469

- Computer Incident Response Team (CIRT), 468–469
 - confidentiality of information, 136
 - configuring
 - alert groups (BASE), 253
 - IDSs (intrusion detection systems), 217–221, 259
 - IPSec on Linux host, 305–311
 - Linux firewalls, 42–56
 - MRTG (Multi Router Traffic Grapher), 397–400
 - MZL and Novatech TrafficStatistic, 400–403
 - netfilter, 32
 - ntop utility, 412–418
 - OpenVPN, 98–108
 - PRTG Traffic Grapher, 400–403
 - SmoothWall Express firewall solution, 80–85
 - Snort on Linux system, 240–254
 - Snort on Windows system, 221–239
 - Stunnel encryption, 300–302
 - TCP Wrappers, 187–188
 - Windows Firewall, 85–86
 - consulting costs and free security solutions, 4, 12
 - contact list for change management, 458
 - converting Windows event logs into syslog-compatible format, 283
 - costs
 - of free security solutions, 13, 19
 - and savings of free security solutions, 2–8
 - critical patches, 449
 - custody, chain of, 328
 - customization costs, 7–9
- D**
- data
 - classifying, 135–136
 - collecting for analysis, 392–394
 - encrypting sensitive, 201–209
 - encrypting syslog traffic, 285–294
 - security of sensitive, 467
 - sniffing, 392
 - Data Encryption Standard (DES), 424
 - “defense-in-depth” described, 24
 - deleting rules, chains, in Linux firewalls, 44
 - demilitarized zones. *See* DMZs
 - denial-of-service (DoS) attacks, 292
 - DES (Data Encryption Standard), 424
 - Designing and Building Enterprise DMZs* (Syngress), 30
 - desktops, proving remote, 108–125
 - destination NAT (DNAT), 49
 - devices, network infrastructure
 - hardening, 175–176, 210
 - security, 452

DHCP (Dynamic Host Configuration Protocol),
firewalls and, 6

diagrams, network security,
362–364

disaster recovery (DR) plans, 365

digest authentication protocol, 424

disabling
 Fedora Core 5 firewall, 43
 newly created policy, 273

DMZs (demilitarized zones)
 costs of, 9–10
 one-legged, 28–29
 true DMZs, 30–31
 and VPN tunnels, 87–88

DNAT (destination NAT), 49

DNS, dynamic, 85

documentation
 access request forms, 364–365
 and change management, 455,
 457
 IT security policies, standards,
 procedures, 365–366
 network security, 361–362
 syslog manual, 297

domain controllers, auditing
 policies for clients logging
 into, 272–274

domain policies, 154

DoS (denial-of-service) attacks,
292

downloading security programs,
tools safely, 356

DTLS (Datagram TLS), 286

dynamic DNS, SmoothWall
support, 85

Dynamic Host Configuration
Protocol. *See* DHCP

E

e-mail alerts, configuring Snort to
send, 261

Easy Firewall Generator, 66

edge firewalls, 28

EFS (encrypted file system),
201–208, 213

enabling
 Fedora Core 5 firewall, 43
 SELinux (security-enhanced
 Linux), 174
 Windows Terminal Services,
 109–110

encrypted file system (EFS),
201–208, 213

encrypting
 sensitive data, 201–208, 211
 syslog traffic, 285–311
 UDP-based logs, 335

encryption
 best hash algorithm, 336
 EFS (encrypted file system),
 201–208
 NetStumbler's detection of, 360
 SNMP versions, 424

EST (Enterprise Scan Tool), 380

Ethereal, 428

Event Log XP, 278

- event logs
 - analyzing Windows, 277–279
 - described, 264
 - ensuring chain of custody, 3285–66
 - ensuring log integrity, 329–331
 - generating, analyzing syslog, 279–327, 333
 - generating Windows, 264–279, 333
 - securing, 327–329, 334
- Event Viewer (Windows), using with event logs, 264–265
- EventCombMT (Microsoft), 275–277
- eventcreate.exe, 275, 285
- EventLog Analyzer 4, 314–316
- Eventlog to Syslog Utility (evtsys), 283
- eventlog.pl, 275, 277
- eventquery.vbs, 275
- events
 - generating syslog, for testing, 335
 - searching for information on, 267
- evtsys (Eventlog to Syslog Utility), 283
- exporting event logs, 278
- F**
- Fedora Core 5
 - configuring Snort on Linux system, 240
 - firewall configuration, 43
 - and FreeNX, 120
 - installing, 429
- file-level access controls
 - Linux, 168–171
 - Windows, 147–152
- File Transfer Protocol (FTP) and firewall types, 26
- Firestarter Linux configuration tool, 59–65
- firewall appliances, 32
- Firewall Builder tool, 66–75
- firewall wizards, 469
- firewalls
 - See also specific firewall*
 - architectures of, 26–31
 - blocking network pings, 339
 - costs of, 6–7, 9–10
 - Linux. *See* Linux firewalls
 - personal, hardening, 180–188, 210
 - scanning ports through, 342
 - types of, 24–26
- floppy disks, installing Linux from, 35–36
- forms, access request, 364–365
- fport utility, 441
- free security solutions
 - costs of, 2–6, 19–20
 - savings of, 6–8, 19–20
 - “selling,” 16–18, 20
 - vs. commercial solutions, 8–16
- FreeNX servers, setting up, 120–121

FTP (File Transfer Protocol) and
firewall types, 26

G

generating

syslog event logs, 279–327, 333

syslog events (Linux), 297–298

test events, 311–312

Windows event logs, 264–279,
333

GPO (group policy object),
153–156, 159

Group Policy, auditing policies for,
273

group policy object (GPO),
153–156, 159

group rights

Linux, 165–168

Windows, 160–163

groups, defining access, 142–147

GUI sniffers, troubleshooting
network problems using,
424–433

H

hardening

infrastructure devices, 175–176,
210

Linux systems, 164–175

personal firewalls, 180–188, 210

systems generally, 133–139, 209

Windows systems, 139–163,
209–210

hardware

costs of free security solutions,
3–4

IDS requirements, 218

vs. software firewalls, 32

hash algorithms, choosing, 336

HIDS (host-based IDS), 217–218

high-risk patches, 449

home network routers,
configuring with Linux
firewall, 47–51

host-based IDS (HIDS), 217–218

hosts, 214

HouseCall online virus scanner,
196

HTTP requests and firewall types,
25–26

HVAC (heating, ventilation and air
conditioning), costs of free
security solutions, 5–6

I

identifying and inventorying your
systems, 338–341

IDS Policy Manager (IDSPM),
configuring, 232–239

IDSs (intrusion detection systems)

configuring, 217–221

demonstrating effectiveness,
257–258

maintenance, 460

management capability, 11

training costs, 3–4

types of, 216–217, 259

- implementing firewalls, 31–86
 - information security, testing for, 383
 - infrastructure devices, hardening, 175–176, 210
 - installing
 - Clam AntiVirus on Linux, 189–193
 - Clam AntiVirus on Windows, 189–193
 - iPig VPN solution, 93–98
 - Linux firewalls, 31
 - Linux options, 33–36
 - SmoothWall Express firewall solution, 77–80
 - Snort, 222–225
 - Windows Terminal Services, 111–112
 - Institute for Security and Open Methodologies (ISECOM), 384–385
 - interfaces, one-legged DMZs, 28–29
 - Internet technology security, 383–384
 - intrusion detection systems. *See* IDSs
 - intrusion prevention systems (IPSs), 217, 261
 - inventorying
 - your systems generally, 338–341, 386
 - your systems using Nmap, 341–347
 - your systems using SuperScanner, 347–351
 - ipchains, 32
 - iPig VPN solution, 93–98
 - IPS (intrusion prevention systems), 460
 - IPSec
 - configuring (Linux), 305–311
 - encrypting syslog traffic using, 288–294
 - encryption capabilities, 286
 - for syslog encryption, 310–311 and VPN tunnels, 88
 - IPSs (intrusion prevention systems), 217, 261
 - iptables
 - chain and rule manipulation commands, 46
 - command summary, 52–56
 - configuring logging, 51–52
 - described, 32
 - ISECOM (Institute for Security and Open Methodologies), 384–385
 - IT (information technology), 365
- ## K
- KDE Guarddog GUI, 76
 - Kerio Personal Firewall, 180
 - Kiwi Logfile Viewer, 313–317
 - Kiwi Secure Tunnel, 287
 - Kiwi Syslog Daemon, 313, 316–320

Kiwi Syslog Message Generator,
311–312

KiwiSyslog, 295–297

L

LanManager authentication, 156

layer 7 firewalls, 26

Libwww HTML utilities, 321

Linksys firewalls, 6–7, 28, 47

Linux

Clam AntiVirus, installing,
189–193, 189–193

configuring Snort on, 240–254,
259

firewalls. *See* Linux firewalls

hosts, enabling SNMP on,
421–424

installing Wireshark on, 428

Nessus scanner, running,
371–375

patching systems, 179–180

reporting tool features (table),
418

SELinux (security-enhanced
Linux), using, 173–175

syslog, generating, encrypting,
receiving events, 297–312

syslog log analysis, 321–327

systems, hardening, 164–175,
209–210

and tcpdump, 434

user and group administration,
165–168

Linux firewalls

configuring, 42–52

Easy Firewall Generator, Firewall
Builder tools, 66–75

Firestarter configuration tool,
59–65

operation of, 36–42

security level configuration,
56–57

versions, choosing, 32–36

listening ports, changing, 109

locating

and inventorying your systems,
338–341

wireless systems, 357–358

log analysis tools, 325–327

log files

analysis plan, implementation,
331–333

analysis, resources for, 333

auditing, 138–139

retention period of, 335

logevent.exe, 275

logging

See also event logs

configuring for Linux firewalls,
51–52

ensuring chain of custody, 328

ensuring log integrity, 329–331

options, Snort, 253–254

LogMeIn Hamachi, 98

logon events, auditing, 269

logwatch reporting tool, 325–327

Lokkit menu for configuring
netfilter firewall, 58–59

Lucent Orinoco chipset, 358

M

MAC addresses, viewing, 345

management

change, 454–459, 470–471

and free security solutions, 11

patch, 448–451

patch management, 453–454

senior, support for penetration
testing, 463–464

Management Information Base
(MIB), and SNMP, 395

MASQUERADE command, 50,
58

metrics

for comparing products, 13–14

reporting, 390–392

MIB (Management Information
Base), and SNMP, 395

Microsoft

See also specific product

group policy object (GPO),
153–156

Microsoft Application Verifier, 146

Microsoft Baseline Security
Analyzer (MBSA), 379–382

Microsoft Malicious Software
Removal Tool, 200–201

Microsoft Management Console.
See MMC

Microsoft Office Visio 2003
Connector, 382

Microsoft Standard User Analyzer,
146

Microsoft Technical Security
Notification Services, 469

Microsoft Windows Defender,
197–199

MMC (Microsoft Management
Console)

collecting Windows event logs,
275–277

configuring Security
Configuration and Analysis
snap-in, 267–274

hardening Windows systems,
139–141

MRTG (Multi Router Traffic
Grapher), 391, 397–400,
444–445

Multi Router Traffic Grapher. *See*
MRTG

MySQL and Snort, 246–247

MZL, configuring, 391, 400–403

N

NASL (Nessus Attack Scripting
Language), 377, 388

NAT (Network Address
Translation)

configuring for Linux firewall,
48–51

firewalls and, 6–7

Nbtscan, 355

- ncurses and Lokkit menu for configuring netfilter firewall, 58
 - Nessus scanner
 - described, 367–368, 464
 - running on Linux, 371–375
 - running on Windows, 368–371
 - net-snmp-utils package, 421
 - NetBIOS information, gathering, 355
 - netcat troubleshooting tool, 439
 - Netfilter firewall, 187
 - netfilter Linux firewalls
 - configuration examples, 42–57
 - configuring, 32
 - and Easy Firewall Generator, 66
 - Firestarter front end, 59–65
 - and Firewall Builder tool, 66–75
 - operation of, 36–42
 - NetFlow protocol, 394, 403
 - Netgear router/firewalls, 47
 - netstat utility, 440–441
 - Network Address Translation. *See* NAT
 - network analysis tools, 15–16
 - network-based IDS (NIDS), 217–221
 - network interfaces, renaming, 100
 - network resources, basic
 - hardening, 133–139
 - network scanners
 - Angry IP Scanner, 351–352
 - Nmap, 341–347, 341–347
 - Scanline scanner, 352–355
 - special-purpose enumerators, 355–357
 - SuperScanner, 347–351
 - networks
 - documentation, diagrams, 361–362
 - infrastructure device security, 452
 - inventorying your systems, 338–341
 - perimeter protection, 24
 - troubleshooting problems, 424–438
 - Virtual Private Networks. *See* VPNs
 - nGenius Performance Manager* (Netscout), 15–16
 - NGsniff, 281
 - ngSniff sniffer, 435–436
 - NIDS (network-based IDS), 217–221
 - Nmap network scanner, 341–347, 377
 - NmapFE, 341–342
 - Novatech TrafficStatistic,
 - configuring, 391, 400–403
 - NTLast event log analyzer, 279
 - ntop utility, 392, 412–418, 444
 - NTsyslog, 283–284
 - NX Client, 132
- ## O
- Object Identifier (OID), and SNMP, 395–396, 400

- objects, auditing access, 270
- OID (Object Identifier), and
SNMP, 395–396, 400
- Oinkmaster (Snort add-on),
254–257
- one-legged DMZs, 28–29
- open source software, 8
- OpenSSH
 - generating new certificate and
key using, 303–305
 - port forwarding with, 299
- OpenSSL, generating message
digests using, 331
- OpenVPN, configuring, 98–108
- OPENXTRA Commander,
412–414
- operating system patches, 453
- operating systems, hardening,
134–139
- organizational units (OUs)
 - auditing policies, 272
 - and group policy objects
(GPOs), 153–156
- OSSTMM (Open Source Security
Testing Methodology
Manual), 382–385
- overwriting event logs, 274
- P**
- packets, and packet-filtering
firewalls, 25
- password policies, 139
- patch management
 - generally, 448–451, 470
 - network infrastructure devices,
452
 - operating system, application
patches, 453–454
 - patching Linux systems, 176–177
 - patching Windows systems,
177–179
 - for protected hosts, 214
- penetration testing, 463–464,
472–473
- performance of free security
solutions, 14
- permissions, Windows, 149–151
- personal firewalls, hardening,
180–188
- physical security
 - ongoing, 466–467
 - policies for, 136
 - testing for, 384
- ping scanning
 - detecting risky ports, 387
 - inventorying hosts on network,
339
- PIX firewalls (Cisco), 32
- PKI (public key infrastructure),
and OpenVPN, 100, 105–106
- planning
 - business continuity (BC) and
disaster recovery (DR) plans,
365
 - change management backout,
457–458
- plug-ins, writing custom security,
388

- PNAT (port NAT), 50
 - point-to-point tunneling protocol (PPTP), 89–90
 - policies
 - account lockout, 159–160
 - audit, 160
 - auditing changes to, 270
 - domain, 154
 - firewall. *See specific firewall*
 - and hardening systems, 135–137
 - IT security, 365
 - maintenance and review, 465–466
 - password, 139
 - policy groups, Firestarter Linux configuration tool, 63–65
 - port forwarding
 - configuring SmoothWall, 83–84
 - OpenSSH, 303–305
 - SSH functionality, 132
 - port NAT (PNAT), 50
 - ports
 - changing listener, 109
 - inventorying your systems, 339
 - risky to target hosts, 387
 - scanning through firewalls, 342
 - PPTP (point-to-point tunneling protocol), 89–90
 - privacy and wireless systems, 361
 - privileges, auditing user rights, 270
 - procedures
 - IT security, 365–366
 - and policy review, 465
 - process security, testing for, 373
 - products
 - See also specific product*
 - metrics for comparing, 13–14
 - programming languages
 - NASL (Nessus Attack Scripting Language), 377, 388
 - Perl, 397
 - proposals for selling free security solutions, 17–18
 - protocols. *See specific protocols*
 - PRTG Traffic Grapher, 391–392
 - public key infrastructure. *See* PKI
 - PuTTY, and SSH, 128–130
- ## R
- Racoon daemon (Linux), 307–308
 - RealVPC, 113
 - Red-Hat-based Linux, security level configuration, 56–57
 - regular expression (regex) filtering syntax, and swatch, 324
 - remote access
 - permitting administrative access, netfilter, 72
 - providing secure, 86–130
 - remote desktops, providing, 108–125
 - remote monitoring (RMON) protocol, 394
 - remote shell, providing, 125–130
 - removing malicious software, 200–201

renaming
 Administrator account, 142–144
 network interfaces, 100

reporting
 on bandwidth usage, 390–392, 442
 and free security solutions, 11–12
 Novatech TrafficStatistic, 400–403

retention
 configuring event log, 274
 of log files, 335

review of security policies, 465–466

rights, user. *See* user rights

RMON (remote monitoring) protocol, 394

roles and responsibilities, vulnerability scanning, 463

routers
 configuring home, with Linux firewall, 47–51
 hardening, 175–176

RRDtool (Round Robin Database tool), 445

rules
 creating Snort to trigger for specific traffic, 262
 deleting in Linux firewalls, 44
 in Linux iptables, 40–42

rulesets, saving for netfilter, 52

runas.exe tool, 145

S

savings and costs of free security solutions, 2–8, 19–20

Scanline scanner, 352–355

scanners
 Nessus, 367–375
 NetStumbler, 358–361
 network. *See* network scanners
 online virus, 196, 212
 X-Scan, 375–379

scanning
 for vulnerabilities, 366–367, 387, 473
 vulnerability, 366–372, 460–463

scheduling, and change management, 455–456

screened subnets, 27–28

scripting languages, 397

scripts
 Bastille hardening, 172–173
 ntop utility, 414–415
 scanning, 461

Seattle Wireless Web site, 359

secedit.exe utility, 158

Secure Shell (SSH), using, 126–130

securing
 event logs, 327–331, 334
 IDSs (intrusion detection systems), 217–221
 network perimeter, 24

security

- assessments, performing, 382–385
- of free security solutions, 14
- and network documentation, 361–362
- network infrastructure devices, 452
- as ongoing process, 448
- patch management, 448–451
- physical, 466–467
- policy review, 465–466
- remote access. *See* remote access
- for sensitive data, 467
- SNMP (Simple Network Management Protocol), 396–397
- solutions, free. *See* free security solutions
- Windows documentation for hardening systems, 152–153
- Security Configuration and Analysis snap-in, audit policy configuration, 267–274
- Security Templates snap-in, 156–157
- SELinux (security-enhanced Linux), using, 173–175
- selling free security solutions, 16–18, 20
- Senao wireless cards, 358
- sendEmail SNMTP program, 321
- sensitive data, security, 201–208, 211, 467
- servers
 - inventorying your systems, 338–341
 - iPig VPN solution, 93–98
 - securing logging, 292
- service level agreements (SLAs), 7
- Service Pack 2 for Windows XP, and patch management, 448
- set group ID (SGID) bit, 169–171
- set user ID (SUID), 169–171
- shell, remote, 125–130
- Simple Mail Transfer Protocol (SMTP), 26
- Simple Network Management Protocol. *See* SNMP
- SLAs (service level agreements), 7
- SmoothWall Express firewall solution, 76–85
- SMTP (Simple Mail Transfer Protocol) and firewall types, 26
- SNARE utility, 283–284, 286
- SNAT (source NAT), 49, 50
- sniffers, 425–433
 - features (table), 438
 - ngSniff, 435–436
 - tcpdump, 437–438
- troubleshooting network problems using command-line, 433–438
- troubleshooting network problems using GUI, 424–433
- Wireshark, 425–433

- sniffing data generally, 392–394
- SNMP (Simple Network Management Protocol)
 - described, 394–397, 443
 - devices, 393
 - enabling on Windows hosts, 418–421
 - Multi Router Traffic Grapher, configuring, 397–400
 - MZL and Novatech TrafficStatistic, configuring, 400–403
 - ntop utility, configuring, 412–418
 - PRTG Traffic Grapher, configuring, 403–412
 - security, 396–397
 - traps, configuring, 297
 - versions of, 424
- Snort IDS (intrusion detection system)
 - and Basic Analysis and Security Engine (BASE), 246–254
 - configuring on Linux system, 240–254, 259
 - configuring on Windows systems, 221–239, 259
 - configuring to send e-mail alerts, 261
 - creating rule to trigger for specific traffic, 262
 - Oinkmaster and other add-ons, 254–257
 - and SmoothWall firewall, 84–85
 - training costs, 3
 - using GUI front end, 231–232
- Snort Intrusion Detection and Prevention Toolkit* (Syngress), 217–221
- Snortsnarf log analyzer, 257
- SNScan, 355
- software
 - antivirus, antispyware systems, 188–201, 212, 459–460
 - commercial vs. free security solutions, 8–16
 - costs of, 7–8
 - “free,” 21
 - free security solutions. *See* free security solutions
 - IDS/IPS, 460
 - removing malicious, 200–201
 - vs. hardware firewalls, 32
- source NAT (SNAT), 49, 50
- spyware, and antispyware software, 188–201, 212, 458–459
- Squid (Snort add-on), 256–257
- SSH (Secure Shell)
 - enabling using SmoothWall, 80–81
 - and Linux security level configuration, 56–57
 - and OpenSSH, 304
 - port forwarding, 305
 - for syslog encryption, 310
 - using, 126–130
- SSL (Secure Sockets Layer) encryption, 286–287

- and OpenVPN, 98
- for syslog encryption, 310
- and VPNs, 88
- standards
 - IT security, 365–366
 - policy review, 465
- stateful inspection firewalls, 25
- strategies for change management, 455–456
- Stunnel, sending logs over SSL-encrypted tunnel, 299
- subnets, screened, 27–28
- su.exe utility, 145
- SUID (set user ID) bit, 169–171
- SuperScanner network scanner, 347–351
- swatch (simple watcher),
 - configuring, 321–325
- switches, hardening, 175–176
- symmetric privacy protocol, 424
- syslog
 - encrypting traffic, 285–294
 - generating, analyzing event logs, 279–327
 - and Linux firewall logging, 51–52
- syslog-ng, 298–300, 299, 311–312
- system events, auditing, 270
- systems
 - antivirus, antispysware systems, 188–201
 - hardening generally, 133–139, 209

- hardening Linux, 164–175
- hardening Windows, 139–163, 209–210
- inventorying your, 338–341
- patching, 176–180
- scanning with Angry IP Scanner, 351–352
- scanning with Nmap, 341–347
- scanning with Scanline, 352–355
- scanning with SuperScanner, 347–351
- special-purpose enumerators, 355–357
- wireless. *See* wireless systems

T

- tables in Linux firewalls, 36
- targets in Linux firewalls, 38
- TCP (Transmission Control Protocol) and server configuration, 25
- TCP ports, scanning, 343–344
- TCP traceroute command, 440
- TCP Wrappers, configuring, 187–188
- tcpdump, 434, 444
- tcpview utility, 441
- templates
 - for security assessments, 384
 - Security Templates snap-in, 156–157
- testing
 - and change management, 457
 - downloaded software, 356

- free security solutions, 14
 - patches, 450–451
 - penetration, 463–464, 472–473
 - syslog events, 335
 - time policies, creating for netfilter, 75
 - TLS (Transport Layer Security), 286
 - tools
 - See also specific tool*
 - downloading safely, 356
 - network scanners. *See* network scanners
 - troubleshooting, 438–441
 - topology, network maps, 362–364
 - traceroute, 344–345
 - traceroute command, 440
 - tracetcp tool, 439–440
 - traffic
 - enabling syslog, 285–294
 - permitting to, from Linux firewall, 44–46
 - TrafficStatistic
 - configuring, 400–403
 - described, 391
 - training costs of free security solutions, 3
 - Transmission Control Protocol. *See* TCP
 - Transport Layer Security (TLS), 286
 - troubleshooting
 - BASE (Basic Analysis and Security Engine), 205
 - investigating event logs, 267
 - network problems, 424–438, 443–444
 - tsclient, 112
 - tunnels
 - Kiwi Secure Tunnel, 287
 - VPN, configuring, 87–93
 - Turtle Firewall Project, 76
 - two-factor authentication, 138
- ## U
- UltraVNC, 113–119
 - updates
 - See also* patch management
 - SmoothWall firewall, 81–83
 - and vulnerability scanning, 461
 - USB drives, installing Linux from, 35
 - User-based Security Model (USM), 424
 - user rights
 - assignment of Linux, 165–168
 - assignment of Windows, 160–163
 - auditing, 270
 - users, defining access, 142–147
 - USM (User-based Security Model), 424
- ## V
- versions, choosing Linux, 32–36
 - Virtual Network Computing (VNC), 109, 113–119

virtual tunnel networks. *See* VPNs
 virus, antivirus systems, 212, 458
 viruses, antivirus systems, 188–201
 Vmailer SMTP program, 320
 VNC (Virtual Network Computing), 109, 113–119
 VPN concentrators, 87–93
 VPNs (Virtual Private Networks)
 costs of, 7
 providing secure remote access, 86–89
 vulnerability scanning, 366–367, 387, 460–463, 473

W

war driving, 357–358
 Web sites, open source software, 12–13
 Windows
 analyzing event logs, 277–279
 Clam AntiVirus, installing, 189–193
 collecting event logs, 275–277
 and EFS, 201–208
 hosts, enabling SNMP on, 418–421
 IPSec policy, 310–311
 Microsoft Windows Defender, using, 197–199
 Nessus scanner, running, 368–371
 patching systems, 177–179
 reporting tool features (table), 418

Snort configuration on, 221–239, 259
 syslog log analysis, 282–293, 312–321
 systems, hardening, 139–163, 209–210
 using as VPN concentrator, 89–93
 X-Scan scanner, running, 375–379
 Windows 2000, configuring
 Windows host as VPN endpoint, 90–93
 Windows Firewall
 configuring, 85–86, 180–186, 213
 configuring Linux firewall and, 46–47
 and patch management, 448
 Windows packet capture driver.
 See WinPcap
 Windows Server 2003 Resource Kit Tools, 275
 Windows Server Update Services (WSUS), 178–179
 Windows Terminal Services,
 providing remote desktops using, 109–113
 Windows XP
 adjusting SNMP security settings, 398
 configuring Windows Firewall, 85–86
 enabling remote desktop functionality, 109–110

- Firewall Builder configuring
 - Linux netfilter example, 66–75
 - Service Pack 2, and patch management, 448
 - WinDump sniffer, 147, 434–435, 444
 - WinPcap, 224–225, 341, 378, 425, 434
 - wireless systems
 - access points, security of, 388
 - detecting with NetStumbler, 358–361
 - locating, 357–358
 - security testing, 384
 - Wireshark sniffer, using, 425–433
 - WSUS (Windows Server Update Services), 178–179
- ## X
- X-Scan scanner, 375–379
 - X Window System, providing remote desktops using, 119–125
 - X.Org foundation, 119–120
- ## Y
- yum (Yellowdog Updater, Modified), 179, 429
- ## Z
- ZoneAlarm, 180