

Additional AIX Security Tools on

IBM server pSeries, IBM RS/6000, and SP/Cluster

Customize the security of your pSeries systems

Explore IBM, non-IBM, and freeware security tools

Learn new approaches to security



Abbas Farazdel
Marc Genty
Bruno Kerouanton
Chune Keat Khor

ibm.com/redbooks

Redbooks



International Technical Support Organization

**Additional AIX Security Tools on
IBM @server pSeries,
IBM RS/6000, and SP/Cluster**

December 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special notices" on page 211.

First Edition (December 2000)

This edition applies to AIX 4.3.3.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. JN9B Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000. All rights reserved.

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

| | |
|---|------|
| Preface | vii |
| The team that wrote this redbook | viii |
| Comments welcome | ix |
| Chapter 1. Introduction | 1 |
| 1.1 Security framework - Overview | 1 |
| 1.2 Security framework - Planning | 3 |
| 1.3 Security framework - Architecture | 6 |
| 1.4 Security framework - Implementation | 7 |
| 1.5 Security framework - Monitoring | 8 |
| 1.6 Security framework - Incident response | 10 |
| 1.7 Next steps | 12 |
| Chapter 2. Firewalls | 13 |
| 2.1 Misconceptions about firewalls | 13 |
| 2.2 Types of firewalls | 15 |
| 2.2.1 Static packet filter firewalls | 15 |
| 2.2.2 Circuit level firewalls | 15 |
| 2.2.3 Application layer (proxy) firewalls | 16 |
| 2.2.4 Dynamic packet filter firewalls | 16 |
| 2.2.5 Comparison between the different types | 16 |
| 2.3 Firewall designs | 17 |
| 2.3.1 Basic firewall design | 17 |
| 2.3.2 Firewalls with demilitarized zone (DMZ) | 18 |
| 2.3.3 Compartmentalized firewall environment design | 21 |
| 2.4 Securing firewalls | 24 |
| 2.5 Firewalls on AIX | 24 |
| Chapter 3. Check Point FireWall-1 | 25 |
| 3.1 FireWall-1 features | 25 |
| 3.2 Complementary software for FireWall-1 | 26 |
| 3.3 Securing FireWall-1 | 27 |
| 3.3.1 Closing vulnerabilities during system startup | 28 |
| 3.3.2 Managing FireWall-1 logs | 29 |
| 3.3.3 Securing FireWall-1 default configurations | 29 |
| 3.3.4 Creating a useful rulebase | 34 |
| 3.3.5 Viewing connections | 36 |
| 3.3.6 Enabling other defense mechanisms | 37 |
| 3.4 List of ports that Check Point FireWall-1 uses | 43 |

| | |
|--|-----|
| Chapter 4. IBM Secureway Firewall | 47 |
| 4.1 IBM Secureway Firewall features | 47 |
| 4.2 Complimentary software for IBM Secureway Firewall | 48 |
| 4.3 Firewall hardening | 49 |
| 4.4 Network Security Auditor (NSA) | 49 |
| 4.4.1 Installing NSA | 49 |
| 4.4.2 Using NSA | 50 |
| 4.4.3 Interpreting NSA output | 51 |
| Chapter 5. Secure remote access | 59 |
| 5.1 Secure Shell (ssh) | 59 |
| 5.1.1 Obtaining SSH | 61 |
| 5.1.2 Difference between SSH1 and SSH2 | 62 |
| 5.1.3 Key concepts of SSH | 62 |
| 5.1.4 Installing OpenSSH on AIX | 65 |
| 5.1.5 OpenSSH using SSH1 | 68 |
| 5.1.6 OpenSSH using SSH2 | 71 |
| 5.1.7 Other interesting SSH daemon configuration options | 75 |
| 5.1.8 SSH2 interoperability between OpenSSH and SSH.Com | 76 |
| 5.1.9 SSH clients for the PC | 76 |
| 5.1.10 Implications of having SSH | 77 |
| 5.1.11 Alternatives to SSH | 77 |
| 5.2 TCP Wrapper | 77 |
| 5.2.1 Obtaining and installing TCP Wrapper | 78 |
| 5.2.2 Configuring TCP Wrapper | 79 |
| 5.2.3 Additional TCP Wrapper security features | 82 |
| Chapter 6. Port and network scanning | 83 |
| 6.1 fping | 84 |
| 6.1.1 Obtaining and installing fping | 85 |
| 6.1.2 Using fping | 86 |
| 6.1.3 Protection against ping sweeps | 88 |
| 6.2 Network Mapper (NMAP) | 89 |
| 6.2.1 Obtaining and installing nmap | 90 |
| 6.2.2 Nmap usage | 92 |
| 6.2.3 Protection against port scanners | 94 |
| 6.3 Security Administrator's Integrated Network Tool (SAINT) | 94 |
| 6.3.1 Obtaining and installing SAINT | 95 |
| 6.3.2 Using SAINT | 98 |
| 6.4 PortSentry | 98 |
| 6.4.1 Obtaining and installing PortSentry | 99 |
| 6.4.2 Defense provided by PortSentry | 103 |
| 6.5 List Open Files (lsof) | 103 |

| | | |
|---|--|------------|
| 6.5.1 | Installing Isof | 104 |
| 6.5.2 | Using Isof | 105 |
| 6.6 | Intrusion detection | 106 |
| Chapter 7. System and data integrity | | 109 |
| 7.1 | Tripwire | 110 |
| 7.1.1 | Obtaining and installing Tripwire | 111 |
| 7.1.2 | Configuring and using Tripwire | 112 |
| 7.1.3 | Configuring Tripwire | 114 |
| 7.1.4 | Comments on configuration | 118 |
| 7.1.5 | When should Tripwire be run | 118 |
| 7.1.6 | Alternatives to Tripwire | 119 |
| 7.2 | John the Ripper | 119 |
| 7.2.1 | Obtaining and installing John the Ripper | 120 |
| 7.2.2 | Configuring John the Ripper | 121 |
| 7.2.3 | Using John the Ripper | 122 |
| 7.3 | Pretty Good Privacy (PGP) | 124 |
| 7.3.1 | PGP basics | 124 |
| 7.3.2 | Obtaining and installing PGP | 126 |
| 7.3.3 | Using PGP | 127 |
| 7.3.4 | Protecting your private key | 134 |
| 7.4 | MD5 | 134 |
| 7.4.1 | Ensuring the integrity of downloads | 136 |
| Chapter 8. Securing AIX | | 139 |
| 8.1 | Overview | 140 |
| 8.2 | Step 1: Remove unnecessary services | 141 |
| 8.2.1 | Removing entries from /etc/inittab | 142 |
| 8.2.2 | Removing entries from /etc/rc.tcpip | 144 |
| 8.2.3 | Removing entries from /etc/inetd.conf | 148 |
| 8.3 | Step 2: Tighten configurations of remaining services | 153 |
| 8.3.1 | Domain Name System (DNS) | 153 |
| 8.3.2 | Network File System and Network Information Service | 168 |
| 8.3.3 | Simple Mail Transfer Protocol (SMTP) | 175 |
| 8.3.4 | Simple Network Management Protocol (SNMP) | 180 |
| 8.3.5 | Trivial File Transfer Protocol (TFTP) | 181 |
| 8.3.6 | Securing X11 | 182 |
| 8.3.7 | File Transfer Protocol (ftp) | 184 |
| 8.3.8 | Protecting TCP services using SOCKS | 186 |
| 8.4 | Step 3: Set proper network (no) options | 186 |
| 8.4.1 | SYN attack protection | 187 |
| 8.4.2 | Broadcast protection | 187 |
| 8.4.3 | IP routing options | 188 |

| | |
|---|------------|
| 8.5 Step 4: Tighten up user accounts | 189 |
| 8.5.1 Removing unnecessary default accounts | 189 |
| 8.5.2 Setting user attributes | 190 |
| 8.5.3 Securing root | 192 |
| 8.5.4 Other attributes | 193 |
| 8.6 Step 5: Set up strong password policy | 194 |
| 8.6.1 Modifying user password attributes. | 194 |
| 8.6.2 Password cracker utility | 197 |
| 8.7 Step 6: Install additional security tools | 197 |
| 8.8 Step 7: Monitor logs, audit trails, and system behavior. | 200 |
| 8.8.1 Monitor system logs | 201 |
| 8.8.2 Enable auditing. | 201 |
| 8.8.3 Monitor files and directories | 202 |
| 8.8.4 Monitor cron and at jobs | 203 |
| Appendix A. NSA Scan Options. | 205 |
| Appendix B. Script used to scan a network with fping | 207 |
| Appendix C. Script to merge the AIX passwd files. | 209 |
| Appendix D. Special notices | 211 |
| Appendix E. Related publications | 215 |
| E.1 IBM Redbooks | 215 |
| E.2 IBM Redbooks collections. | 215 |
| E.3 Other resources | 216 |
| E.4 Referenced Web sites. | 216 |
| How to get IBM Redbooks | 221 |
| IBM Redbooks fax order form | 222 |
| Index | 223 |
| IBM Redbooks review | 243 |

Preface

Information is one of your most important assets in the digital economy. Whether in an online store or a complex Internet-based commodity exchange, the viability of e-commerce is critically dependent on the secure flow of information.

From firewalls to operating system hardening, this redbook illustrates additional tools and techniques that you can use to enhance the security environment of your IBM RS/6000 and IBM @server pSeries. The approach taken is from outside to inside and top to bottom. We move from the servers on the far reaches of your network that are visible to the outside world to those on the innermost recesses of your intranet that contain your most confidential data. As we move through these servers, we work from the application layer at the top to the network layer at the bottom. Along the way, we cover third-party software that is readily available, modifications to the standard software that comes with AIX and PSSP, and assorted techniques that can all be used to provide enhanced security in your environment.

This redbook is the third in the security redbook trilogy:

- *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962
- *Exploiting RS/6000 SP Security: Keeping It Safe*, SG24-5521
- This redbook

Like the others, this book is primarily aimed at Information Technology (IT) professionals responsible for managing and securing their server environment, be it all RS/6000 SP, all RS/6000 standalone, or a mixture of both. We assume, at a minimum, that you are familiar with AIX and, in an SP environment, PSSP.

Subjects covered in this redbook include:

- Firewalls
- Secure Remote Access
- Network Mapping and Port Scanning
- System Integrity
- Securing AIX

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Abbas Farazdel is an SP System Strategist, Technical Consultant, and Senior Project Manager at the International Technical Support Organization, Poughkeepsie Center. Before joining the ITSO in 1998, Dr. Farazdel worked in the Global Business Intelligence Solutions (GBIS) group at IBM Dallas as an Implementation Manager for Data Warehousing and Data Mining Solutions and in the Scientific and Technical Systems and Solutions (STSS) group at the IBM Thomas J. Watson Research Center as a High Performance Computing Specialist. Dr. Farazdel holds a Ph.D. in Computational Quantum Chemistry and an M.Sc. in Computational Physics from the University of Massachusetts.

Marc Genty is a Systems Management Integration Professional working for IBM Global Services at the Western Geoplex Service Delivery Center in Boulder, Colorado. He has worked in UNIX environments for 10 years and is an RS/6000 Certified Advanced Technical Expert (CATE). He holds a BS degree in Manufacturing from Colorado State University. His areas of expertise include UNIX, AIX, RS/6000 SP, HACMP, and DCE/DFS. He is currently the Global Web Architecture (GWA) AIX Architecture Team Lead in Boulder. He was also a contributing author of the book, *Exploiting RS/6000 SP Security: Keeping It Safe*, SG24-5521.

Bruno Kerouanton is a System and Security Engineer working for Sysicom in France. He has four years of experience with AIX and RS/6000 SP system administration and support. Bruno is an RS/6000 Certified Advanced Technical Expert (CATE). He also holds CCSA and CCSE certifications for CheckPoint Firewall-1/VPN. His areas of expertise include network and system security including penetration tests and architecture design.

Chune Keat Khor is an IT Availability Specialist from IBM Singapore. He joined IBM in 1998 after graduating from National University of Singapore with a bachelor's degree in Electrical Engineering. He is an AIX Certified Advanced Technical Expert and has received Check Point CCSA and CCSE certifications.

Thanks to the following people for their invaluable contributions to this project:

IBM Boulder

Tom Kleespies, Jeff Quaintance

IBM Poughkeepsie

Chris Derobertis, Larry Parker

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 243 to the fax number shown on the form.
- Use the online evaluation form found at ibm.com/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

x Additional Security Tools for AIX Systems

Chapter 1. Introduction

"More than any other single factor, the potential of e-commerce hinges on people's confidence that the network can keep confidential transactions confidential, and private records private."

IBM Chairman, Lou Gerstner, December 1996 Internet World

Information is one of your most important assets in the digital economy. Whether we are talking about an online store or a complex, Internet-based commodity exchange, the viability of e-commerce is critically dependent on the secure flow of information.

IBM has always been serious about security and has unparalleled experience designing and implementing secure systems. By using this knowledge and leveraging services based on open standards, the IBM RS/6000 and the new IBM @server pSeries servers deliver a complete security package that you can count on from one end of your enterprise system to the other in a flexible, easy-to-manage manner.

The security infrastructure shipped with AIX 4.3 is covered in the redbook:

- *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962

and that of PSSP 3.2 in the redbook:

- *Exploiting RS/6000 SP Security: Keeping It Safe*, SG24-5521

The purpose of this redbook is to provide you with information about additional security tools (both IBM and non-IBM) and techniques that can be used to further enhance the security of your AIX 4.3 and/or PSSP 3.2 environments. Many of the tools described here are available for download from the Internet.

1.1 Security framework - Overview

This book begins by describing a framework for security. Your environment will only be as secure as your weakest link. For this reason, it is critically important to take a holistic view of security, from planning to architecture to implementation to ongoing support.

Once the security framework has been described, we go on to cover some of the tools and techniques you can use to implement it. The framework

provides both the context and the direction for which tools and techniques you use and how you use them.

The security approach we take is top-down. The topology of a typical eBusiness environment is structured something like the following (as viewed from the Internet side):

- Boundary routers
- External firewalls
- Public servers in the demilitarized zone (DMZ)
- Private network in the DMZ
- Internal firewalls
- Private internal servers
- Private internal network

Also, viewing each of the servers top-down, we see:

- Application software
- Operating system
- Network (hardware and software)
- Physical server and network media

Each element in the environment and each layer of each element needs to be taken into account when building your security framework.

The steps to building a security framework and filling in the structure are as follows:

1. Planning - This is where you define your overall security policies and goals. In many organizations, this step is performed at the corporate level and, likely, has already been completed.
2. Architecture - This is where you design your environment to meet the requirements defined in the Planning phase.
3. Implementation - This is where the rubber meets the road, and you build what you designed in the Architecture phase.
4. Monitoring - Once your environment is operational, you need to continuously monitor it for vulnerabilities and suspected attacks. This phase forms the feedback loop for all of the previous phases. A problem discovered here should pass through some, if not all, of the previous phases on its way toward resolution.

5. Incident response - This is the phase that you hope you will never have to go into. However, if the worst should happen, you will want to be well prepared. The absolute worst time to begin working on this phase is after an attack has already occurred. Time spent in the beginning considering how you would respond to a real attack will pay for itself many times over if you ever find yourself confronted with the real thing. Think of this as the “peace of mind” phase.

In this chapter, we cover all of the phases and give you ideas on things to consider when you work through them for your organization. There are many good books on the subject of security planning and architecture. The focus of this book is primarily on security implementation, monitoring, and, tangentially, incident response. Keep in mind as you read it that it is meant as a companion to the other two security redbooks mentioned at the start of this chapter. Throughout the book, we refer you to those other two books when appropriate so as not to duplicate detailed information contained there.

1.2 Security framework - Planning

The purpose of this phase is to define the overall security goals and policies for your organization. Some of the questions that should be answered during this phase are:

- How much security do you need?
- How much security can you afford?
- What is the nature of the enemy?

The first two questions are interdependent. In general, the more security you need, the more it will cost. This is another place where the 80/20 rule holds. There is a lot that you can do in terms of security for not a great deal of money (80 percent security - 20 percent money), but if you need very high security, you should expect to pay a premium for it (additional 20 percent security - additional 80 percent money).

You must also weigh the value of the assets you are trying to protect with the amount of money you are allocating for security. For example, a server in a DMZ that can be easily reconstructed should something happen to it typically does not warrant the same level of security that you would give to an internal database server housing confidential information.

However, there are also intangibles that you need to take into account. Even though a public Web server does not contain confidential information, the damage done to the reputation of your organization should it be compromised

can be huge. Reputations take a very long time to establish and a very short time to destroy.

Finally, you need to be realistic in your planning. Having elaborate security goals and policies is all well and good, but you must also back them with the funding for the staff to support them. System and network administrators typically have their hands full just trying to keep the systems running and maintained. Depending on your needs, you may want to consider staffing a separate security team. At the very least, consider having a security lead within the ranks of your system and/or network administration team.

"If you know the enemy and know yourself, you need not fear the result of a hundred battles"

Sun Tzu, *The Art of War*

A big part of security planning is threat assessment. Even though attacks from the outside are what we hear about most often, attacks from the inside are far more common, but, for obvious reasons, rarely publicized. Your planning should take into account both groups.

A good starting point for learning how to understand your enemy is detailed in the book *Hacking Exposed: Network Security Secrets and Solutions*. Through detailed, real-world examples, this book teaches you how to think like an attacker. In it, the authors outline the typical methodology used by would-be attackers to size up their targets. The steps of that methodology are as follows:

1. Target acquisition and information gathering
2. Initial access
3. Privilege escalation
4. Covering tracks
5. Planting back doors

A brief explanation of each step is given here. For a more thorough explanation, refer to the book (*Hacking Exposed*).

Target acquisition and information gathering

This step can be broken into three phases:

- The first phase is *footprinting*, where the attacker attempts to find out as much as possible about your environment. Your IP address range(s) and DNS information are prime targets during this phase.

- The second phase is *scanning*. Having acquired your IP address range(s) and/or DNS information, the attacker now starts probing your systems looking for active ports. Tools, such as NSA, SAINT, fping, and nmap, which are all covered in this redbook, are commonly used during this phase.
- The third phase is *enumeration*. This phase is much more intrusive than the previous one. Having narrowed the target to specific systems, the attacker now attempts to find a hole that will enable initial access into one or more of your systems. Banners, default users, and poorly-configured network resources are all prime targets during this phase. Operating system hardening and strong password policy (both covered in this redbook) are two of the most effective defenses against this phase.

Initial access

The attacker next attempts to penetrate the system. The initial access is typically not root access, but no worry, that will come next. In this step, the attacker uses things, such as social engineering, password guessing, password cracking, or buffer overflow techniques, to gain the first toehold into your systems. There are many tools and techniques described in this redbook to help you prevent this from happening. We also cover additional tools and techniques that can act as an early warning device to suspicious activities and alert you prior to the attacker moving to the next step.

Privilege escalation

Once root is compromised, all is compromised (at least on this system). This is every system administrator's worst nightmare. Many of the same techniques used in the previous step apply here, but now the attacker has the luxury of actually being on the system, which makes the job considerably easier. In addition to the tools and techniques for protecting normal user accounts, there are additional considerations and steps for root. They too are covered in this redbook.

Covering tracks

Having gone through all the work in the previous steps to gain access to the system, the last thing the attacker wants is to be discovered and have all that work go for naught. Altering log files, configuration files, and mimicking normal system user behavior are but some of the tricks the attacker uses to not be detected. Your first defense here is to know what "normal system behavior" looks like. There are many monitoring tools (system-provided and third-party) to help you with this. In addition, tools, such as Tripwire, provide a baseline snapshot of your critical system directories and files for you to

compare things against. We cover both system monitoring and Tripwire later in this redbook.

Planting back doors

The attacker knows full well that the door used to enter the system may not be there when needed again. Like any good burglar, the attacker wants to make sure that there are multiple points of entry available should the need arise to again visit the scene of the crime (and, in the case of the attacker, the need almost always arises again). To this end, the attacker plants backdoors and Trojan horses, not only to secure future access, but to also, hopefully, gain enough additional information to be able to launch successful attacks on other systems in your environment. The defense tools covered in the previous step also apply here, especially tools like Tripwire.

Of course, internal attackers typically have the advantage of being able to hide under the cover of a normal user account. Beyond that, though, their methods are quite similar to their external counterparts, even though their motives probably are not.

Once you have defined your overall security goals and policies, both defensive and offensive, you are ready to move on to laying out your security architecture. It is important to remember that time invested in the early phases (such as planning and architecture) pays off by orders of magnitude in the later phases.

1.3 Security framework - Architecture

In general, security requirements differ across environments. Security requirements for an isolated lab environment are quite different from those for a large eCommerce site. The primary focus for the first is on physical access controls, whereas the primary focus for the second is on preventing attacks.

In the architecture phase, you examine your environment and plan your high-level defenses. Some of the questions to consider are:

- What are the weakest points in your environment?
- What type of attacks are you expecting?
- From where do you anticipate the attacks to originate?
- Do you concentrate most of your effort and resources on the perimeter, or do you set up rings of defense, with the perimeter being the outermost ring?

The specific questions (and answers) depend not only on the type of environment you have, but also on whether or not it is a new or existing environment. For example, fortifying a DMZ is a much different task than creating one. Additionally, when planning your architecture, don't forget to take into account attacks from the inside as well as those from the outside. Think of this phase as a chess match:

- How do you plan to protect the king?
- What strategy do you plan to use to out-fox your opponent?
- How will you recover if your initial defenses falter?
- Should you use subterfuge to try and throw him off?
- What will you do if your opponent does not use the strategy that you are expecting?

As you lay out your architecture, it is very important to keep in mind funding and support considerations. You need to build something that can be supported in terms of cost, effort, and expertise. Architecting an unsupported security environment is, in some ways, worse than having no security environment at all. You are lulled into a false sense of security by your paper architecture, which bears no resemblance to what is actually in place. Above all else, be realistic when developing your security architecture, and, if possible, include input from the people who will ultimately have to support your design.

Two excellent books to consult during this phase, especially when considering strategy, are *The Art of War*, ISBN 0-1950-1476-6, by Sun Tzu, and *Information Warfare and Security*, ISBN 0-2014-3303-6, by Dorothy E. Denning. In addition, the tools and techniques covered in this redbook can be used to provide some of the "bricks and mortar" for your design.

1.4 Security framework - Implementation

This phase also depends on whether or not you are working with a new or existing infrastructure. Retrofitting security to an existing environment is much more complicated than building it from scratch. You have to take into account existing production services and plan your changes in such a manner as to not disrupt these services. Tools, such as IPSec, TCP Wrapper, and, especially, PortSentry, are tricky to configure properly on the first attempt. You should test them thoroughly before implementation and monitor them closely afterwards to be certain that you have not inadvertently denied access to legitimate clients.

A methodical approach to this phase will serve you well. Here are some possibilities:

- Start with servers on the perimeter of your network and work your way back to the servers on your internal networks, or vice versa.
- Implement by server type. For example, all firewalls, then all Web servers, then all mail servers, and so on.
- Start with a specific security package, such as IPSec or TCP Wrapper, and implement that on all designated servers; then, move on to the next security package.
- Work from the bottom to the top. Start with the physical layer (physical security); then, move to the network layer, then the operating system layer, and, finally, the application layer.

Which ever methodology you pick, stick with it throughout your implementation phase, and document the configuration as you go. It is a very good idea to keep a complete set of your system and security documentation, such as listings of configuration files and filter rules) stored offline. In the event of a breach, this documentation becomes invaluable for damage control and postmortem activities.

One of the primary goals of this redbook is to provide you with the information you need to successfully implement your security environment with some or all of the tools and techniques covered here. It is important to understand that this phase, in particular, is iterative. Security implementation is not a one time thing. Discovery of new vulnerabilities in software and hardware requires that you routinely revisit your implementation to apply new patches, update configuration files and lists, and, at times, replace ineffective tools.

1.5 Security framework - Monitoring

So, now that you have your security environment set up and operational, the next phase involves adding a feedback loop to the previous phases. In this phase, you implement system and security monitoring. Both automated and manual monitoring should be considered here. The goal is to develop a reliable system that provides you with early warnings of suspected attacks. There is, however, a very fine line between too little information and too much information when it comes to this type of monitoring. With too little information, you run the risk of missing suspicious activities early enough to take the necessary preventative action. With too much information, you also run the risk of missing something important, but, this time, it is because it gets

lost in all of the detail and background noise. It takes time and experience with the data to find the balance that works best for you.

Here are some of the areas to consider for monitoring:

- Application logs (httpd_logs, access_logs, firewall logs, and so on)
- Special logs provided by the additional security tools you have installed
- Audit logs
- System logs (syslog, sulog, wtmp, lastlog, failedlogin, and so on)
- System errors (errpt)
- System performance statistics (vmstat, iostat, topas, and so on)
- Network performance statistics (netstat, netpmon, and so on)
- Physical access records
- Critical system file and directory permissions and ownership

Even though it is possible to automate the entire monitoring process, it is, generally, not a good idea. A combination of automated and manual monitoring is a better solution. Here are some of the reasons why:

- It is too easy to take fully-automated monitoring for granted and, over time, become complacent.
- Fully-automated monitoring follows the set logic pattern that is built into the scripts.
- Manual monitoring requires you to think about security on a daily basis. Performing a system health check every morning (at least on a random subset of your systems) keeps security in the forefront of your mind.
- Manual monitoring helps you develop a feel for the normal performance of your systems and environment. This empirical data often proves to be a far more useful early warning indicator than any of the data coming out of the automated processes.
- With manual monitoring, you can follow an infinite number of paths and make unexpected linkages that would be impossible with automated monitoring. A great illustration of this point can be found in *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*, ISBN 0-6717-2688-9, by Clifford Stoll.

Do both, automated and manual. Also, consider using multiple tools and multiple data sources to cross-check the information provided by each of the tools and logs. Like a good newspaper reporter, verify your sources. Tools are not perfect, and individual logs do not capture everything; each has its own

strengths and weaknesses. What you are striving for is an overall picture and feel for the health of your environment.

It is also a good idea to take a snapshot of each of your systems periodically and store the information on a different system and/or offline. Be sure to keep several historical copies of the snapshots as well, so that you have more than one baseline to use as a reference when investigating system anomalies.

One final note to close of the discussion of these first four phases. If you do not feel comfortable tackling these processes alone, there are a number of professional services companies (IBM being just one) that can help you through the process. They will even be there for you should the worst happen, and you find that you have to move on to the next phase (for real). To find out more about the IBM Emergency Response Service, visit:

<http://www.ers.ibm.com/>

1.6 Security framework - Incident response

Nobody likes to think about this phase. It is every system administrator's worst nightmare, but that is exactly why it should be thought about, planned for, and regularly rehearsed.

The absolute worst time to figure out how you will respond to a security incident is after one has already occurred. The goal of this phase is to be in a position such that, if the worst happens, you know exactly what to do and how to do it. In other words, your incident response team springs into action like a well-oiled machine.

So, the first step in the process is to identify your incident response team. Even if you have contracted out your emergency response services (ERS), or even if your organization has a corporate-level ERS team, you should still designate a small team of people who have a thorough knowledge of the incident response policies and procedures and can take control of the situation when a breach is discovered. Like any professional team, they should practice, practice, practice. Consider implementing quarterly dry-runs, where a simulated breach is identified and dealt with. It is also extremely important for everyone in your organization to understand that this team is in total control in the case of a breach or suspected breach, and they should all look to this team for all command and control decisions. Additionally, your incident response team should have one single leader who has the final say on all investigative activities.

Think of responding to an incident in terms of a crime scene investigation. You need to gather forensic evidence in such a way as to not contaminate the scene. In the case of incident response, you have the added challenge of not knowing if the attacker has fled the scene of the crime. You have to assume that the attacker is still present. So, you must attempt to conduct your investigation in a manner that will not alert the attacker to your activities. This is extraordinarily difficult to do, especially on an emotional level. Your first instinct is to shut everything down so that the attacker cannot get back in. This, however, fundamentally changes the crime scene, and may destroy any chance of actually identifying and catching the attacker. This is why it is a very good idea to have a neutral third-party in charge of the incident response team. Typically, the people responsible for the day-to-day administration of the systems are too close to the situation emotionally and do not have the perspective necessary to make the calm, rational decisions needed in a situation like this. That, however, is not to say that their expertise is not needed. The investigation cannot occur without it.

Incident response should follow a systematic approach similar to the following:

1. Identify the suspected scope of the breach.
2. Start by outlining what is known.
3. Work methodically through the environment, either from outside in (external attack) or inside out (internal attack).
4. Work methodically through each system. Start with the application layer and work down, or start with the network layer and work up.
5. Have a predefined list of items to look at on each system. Additional items can be inspected as needed, but, at a minimum, inspect each item on the list in the order identified on the list. The list should include things like system logs, system configuration files, directory and file permissions, audit logs, application logs, output from system commands, source code files, system error reports, and so on.
6. Document everything as you go, and update your outline of what is known.
7. Validate assumptions with facts and data. Speculation is fine, but do not be too quick to jump to conclusions, and make sure that the conclusions can be backed up with hard evidence.
8. Do not wait too long before calling in additional experts to help. Remember that it does not take long for the tail of evidence to grow cold in a crime scene investigation.

Keep this final phase in mind as you read through the remainder of this redbook. For each of the tools and techniques, think about what would be of value to you during the investigation of a security incident.

Additional information about incident response in the form of step-by-step guides is available (for a fee) from:

<http://www.sanstore.org/>

1.7 Next steps

The remainder of this redbook covers additional tools and techniques you can use to build or enhance your security environment. The material is roughly organized from an outside-in, top-down perspective. It begins with a discussion of firewalls and ends with a discussion of operating system hardening.

Many of the tools covered in this book are available for AIX in installp format from the Bull site at:

<http://www-frec.bull.fr/>

If you have never visited this site before, we encourage you to go there and browse. In addition to the security tools, it contains a wealth of other software for AIX.

Chapter 2. Firewalls

A firewall is a system designed to prevent unauthorized access to or from a private internal network. It is typically set up at the periphery of a network, whether it is between the internal and external networks or between different segments of the internal network (for example, between departmental LANs). You may liken a firewall to the security post at the entrance of a high-security area where the guards will allow only vehicles with appropriate passes to pass through.

For firewalls to properly do their job, all network traffic must be routed through them. Having alternate network paths defeats the purpose of having a firewall. Dial-up connections within the network to independent ISPs is an example of an alternate network path. The total security of a network depends on its weakest link. Having alternate paths may weaken the security posture and create an element of the unknown because of the additional vulnerabilities opened up through the alternate path. In general, elements of uncertainty should be eliminated or at least minimized in a secure environment.

Firewalls can protect against most known forms of direct attack. Since new exploits are developed or discovered constantly, keeping up to date with the latest attack methods as well as product patch or release levels is a good idea. It is impossible to predict all possible attacks that may be attempted. However, properly-configured firewalls (together with properly-configured routers, switches, and Internet servers) go a long way in providing good protection against most forms of attack.

Firewalls can be implemented as hardware or software solutions. Hardware solutions are simpler and, typically, not as sophisticated as their software counterparts. In general, hardware solutions usually have superior performance. In this book, we will focus on the software solutions.

2.1 Misconceptions about firewalls

A big misconception is that firewalls provide all the protection required for a network. This is untrue. Firewalls can be the primary security enforcer, but must be used in conjunction with other security elements. For example, if you have a badly-configured Web server that has security problems, having a firewall will not be sufficient. Attackers will be able to capitalize on the vulnerabilities of the Web server using legitimate commands from their remote client through the firewall to the Web server.

A firewall does not provide protection in server exploits where a legitimate client accesses a legitimate server on a legitimate port. Vulnerabilities on the servers themselves may be exploited by attackers to gain unauthorized access to the servers themselves. As such, it is important to stay current with security patches (operating system patches and application patches) and maintain rigorous control of the system configurations on the servers themselves.

Some firewalls, such as Check Point FireWall-1, provide protection from Java and ActiveX exploits by giving the option to filter out all Java and ActiveX code through the firewall. Since Java and ActiveX exploits are based on programming problems, firewalls cannot reliably decide which code is good or bad. By filtering all Java and ActiveX code, all programs written in these languages, be they well written or poorly written, will not be allowed through the firewall.

Another common error is to leave firewalls in their default configurations. Default configurations are typically insufficient and may leave certain known vulnerabilities open. For example, Java and ActiveX filters are not enabled by default in Check Point FireWall-1. Therefore, it is important to stay up to date with new vulnerabilities (through firewall vendor Web sites, mailing lists, CERT advisories at www.cert.org, and so on), and close them in a timely manner.

Firewalls do not provide protection for password breaches. If an attacker steals a password or guesses it correctly, the firewall will be unable to block such an attacker. As long as someone has the password (by whatever means), firewalls have difficulty determining which user is authentic and which is not. With this in mind, it is important to implement strong passwords on AIX (see Section 8.6, "Step 5: Set up strong password policy" on page 194, for more information on how to implement strong passwords).

Another area where there is typically a lack of focus is logging. Many system administrators do not bother with logging because either they do not realize its importance or they are under the misconception that it is unnecessary. Logging becomes critical when a security breach occurs. Hackers will typically test the site for weak spots prior to launching an attack. They will probe and scan the network looking for potential weaknesses. Oftentimes, it is possible to detect such activity by carefully watching the logs. Additionally, if a break-in attempt does occur, the logs can prove invaluable to the security experts doing the analysis and tracking. With this in mind, it is important to keep all logs (system and firewall) in a safe and trusted location and to maintain regular offline backup copies of them as well.

2.2 Types of firewalls

There are four architectural models for firewalls:

- Static packet filter firewalls
- Circuit level firewalls
- Application layer (proxy) firewalls
- Dynamic packet filter firewalls

Each is briefly described in the following sections.

2.2.1 Static packet filter firewalls

Static packet filter firewalls control traffic by examining the packet header information passing between service ports. For example, if you have an Internet Web server set up for your business, you have to configure the static packet filter firewall to allow traffic from all external users (Internet) to your Web server (port 80, typically). The level of filtering is simple and has no embedded logic to handle the more sophisticated attacks.

This type of firewall inspects packets at the network layer and is not capable of processing state information in the application-layer protocols, such as ftp/http/telnet. The security level is the lowest and is not considered secure. Static packet filter firewalls are the fastest of the four types. They are stateless, support network address translation, and can be implemented in hardware.

AIX 4.3.x provides static packet filtering through the IPsec facility (`bos.net.ipsec.rte` files), and static packet filtering is also native to many routers. Routers, in particular, are very important companions for other types of firewalls. Although the router cannot by itself be the primary security enforcer, a properly-configured router is essential to the overall security of the network. Routers with well-designed and maintained filter sets go a long way towards fending off hackers because these filters provide the first line of defense in many internet attacks (denial of service, network mapping).

2.2.2 Circuit level firewalls

Circuit level firewalls operate at the transport layer. They support TCP protocol only (no UDP). This type of firewall validates that a packet is either a connection request or a data packet belonging to a connection or a virtual circuit between two peer transport layers. The firewall forms a type of state table to monitor the handshake of each new connection. After the handshakes are complete, the firewall inspects the packets to determine if the

packets are to be allowed through or not, based on a ruleset. Once allowed, all network packets associated with this connection are allowed without further checks.

Circuit level firewalls are generally faster than application layer firewalls because less computation is done on the packets. They also support network address translation.

2.2.3 Application layer (proxy) firewalls

Application layer firewalls act as a proxy between internal and external users. No direct connections are made between the internal clients and the external server. The advantage of proxy firewalls is that the firewall has better control over what access to grant. For example, an FTP proxy can be used to restrict commands, for example, allowing `get` but disallowing `put` commands.

The disadvantage of proxy firewalls is that they are tied to specific protocols. An FTP proxy is different from a HTTP proxy and so on. Each specific protocol requires its own specific proxy, and when new protocols come out, new proxies have to be developed by the firewall vendors.

A proxy firewall sits transparently between an end-user and a service provider; both sides are unaware of the proxy in the middle. Proxy firewalls are slow in comparison to the other types of firewalls. They support network address translation and provide very good auditing and logging facilities.

2.2.4 Dynamic packet filter firewalls

Dynamic packet filter firewalls provide additional protection for the UDP transport protocol by associating the UDP packets with a virtual connection table within the firewall. This connection table keeps track of the connection at both sides of the firewall. So, for example, if an attacker attempts to simulate a reply to a request from an internal host, it will fail. The dynamic packet filter firewall will know that the internal host did not initiate the request, and, therefore, will not allow the simulated reply in. A static packet filter firewall would not be able to detect that the reply was simulated and could potentially allow it to pass to the internal host and inflict its intended damage.

2.2.5 Comparison between the different types

Static packet filter firewalls are considered outdated and the least secure. They should not be used as the sole enforcer of security because of their limitations. However, when implemented on routers at the edge of your network, they form the first line of defense in good firewall design by preventing certain traffic from reaching your network.

Circuit level, proxy, and dynamic packet filter firewalls have their strengths and weaknesses. Good firewalls are hybrids, with characteristics of each of these types of firewalls incorporated into one product.

For more information on firewall evolution and types, visit:

www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm

2.3 Firewall designs

Having covered firewall architecture, we now turn our attention to firewall design. The complexity of your network and the nature of your business will influence and shape the decisions you make here. These designs are presented merely as a starting point for your design work.

2.3.1 Basic firewall design

The most basic firewall system is one that separates two IP networks, such as the Internet and the company intranet. All traffic between the two security zones must pass through the firewall system for it to be effective. The configuration of the firewall specifies which connections are allowed and which are denied. The basic firewall design is shown in Figure 1.

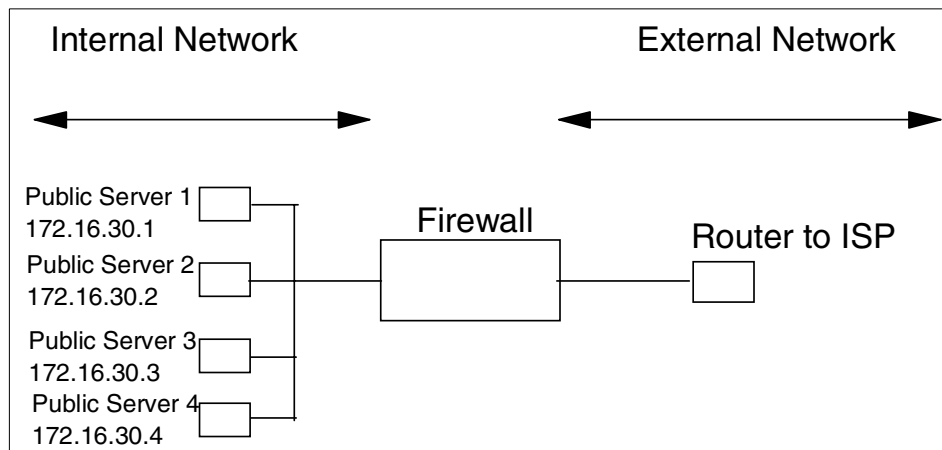


Figure 1. Basic firewall design

Different technologies can be used to control the traffic flow between the networks. Packet filtering checks individual IP packets, and proxies work on the level of connections and application byte streams. In modern firewall products, these techniques are often combined in a hybrid design that

supports both techniques as discussed in Section 2.3, “Firewall designs” on page 17.

It is important to keep in mind that a firewall is only able to check the traffic between the different attached networks. It cannot prohibit unwanted connections within one security zone. This fact can lead to major security risks. For example, if the company's public Web server is placed within the internal network, the firewall needs to be configured to allow HTTP connections to this system so that everyone can get to the Web pages. If the Web server contains security holes (due to software bugs, configuration errors, insecure dynamic content, or any one of many other possible causes), an attacker can gain full access to the Web server system. The firewall cannot prevent the attacker from leveraging this to access other systems within one security zone (in other words, the internal network). Placing important servers outside the firewall in the external network is not recommended either, since, then, they cannot be protected by the firewall against attacks.

Experience shows that it is not realistic to expect complex server software, such as Web servers, to be free of security holes. Major companies and government institutions have frequently been victims of these kinds of attacks. Every day, new security holes are found and shared in the underground by hackers. It takes time for this information to reach public Internet sites and security mailing lists and even more time for software companies to develop and test patches to fix the vulnerabilities. Knowledge is your best defense, and, to that end, it is prudent to keep up to date with the latest information by reading security-related Web sites, joining mailing lists, and learning to think as the hackers do.

2.3.2 Firewalls with demilitarized zone (DMZ)

More security can be gained by introducing a perimeter network within which public servers can be placed. This zone, known as the DMZ, is considered to be semi-secure. Secure because a firewall is protecting it, and not secure because it is accessible directly from the Internet. Servers that are to be visible from the Internet should be placed in the DMZ. All other servers, including infrastructure support servers to the DMZ, should be separated from the DMZ by another firewall. Thus, the classic DMZ setup has two firewalls and a DMZ server network in-between as shown in Figure 2.

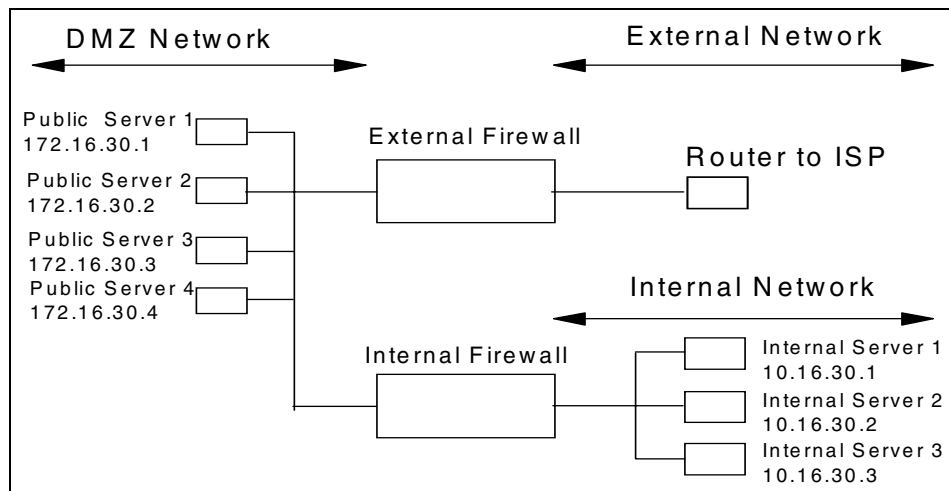


Figure 2. Firewall with DMZ

The advantage of this setup is that the publicly-accessible servers are now protected from the external network and separated from the internal network. No direct connection is allowed from the Internet to the internal servers, and two firewalls must be compromised for attackers to gain access. You may also want to consider using different firewall technologies on the two firewalls; that way, vulnerabilities in one firewall may not be used to breach the second firewall. The obvious disadvantage of this setup is that you need two firewalls, which increases the complexity and the administrative overhead, especially if different technologies are used for the two firewalls.

Further protection is still necessary for the public servers on the DMZ network. In a worst case scenario, when Public Server 1 is broken into, more security is lost than necessary. For example:

1. The intruder that broke into Public Server 1 can now freely attack Public Server 2 because there is nothing between them.
2. The intruder on Public Server 1 can easily monitor all network traffic (including company e-mail and other potentially sensitive information when collected systematically) that leaves both firewalls on the DMZ network side. This technique is known as network sniffing. Analyzing who is talking to whom is called traffic analysis (even encrypted mail typically has plain text From: and To: mail addresses information that allows some insight into possibly confidential transactions).

One common way of closing this vulnerability is to use manageable switches between the servers in the DMZ network. Another possibility is to use static packet filter routers with filters set to prevent certain traffic or certain routes. Using the routing table, you may also segregate subnets within the DMZ network. A third option is to use the IPSec facility under AIX 4.3.x to implement static packet filtering directly on the public servers.

Switches are a good choice since they do not employ a shared medium. For example, on a shared medium, account logins from one server to another are visible to all servers sharing that same medium. Services, such as telnet and ftp, transmit account IDs and passwords in the clear across the network. Suppose Servers A, B, and C are all on the shared medium, and suppose further that an attacker has compromised Server C. The account information (ID and password) passed between Server A and Server B when a user establishes a telnet or ftp session is visible (through sniffing) to the attacker on Server C. With switches, the medium is no longer shared but, rather, point-to-point, which makes the kind of sniffing outlined in this scenario much more difficult. A properly-configured switch will have a limited number of paths from the source to the destination, and access to multiple destinations is controlled entirely within the switch.

However, since active network devices, switches, and routers are designed with performance, speed, and convenience as their primary objectives. Security is somewhat of an afterthought. Experience shows that they are, therefore, not dependable as your only means of defense. In addition to missing emphasis on security in development, they usually cannot properly filter even common protocols, such as FTP, due to very limited filtering capabilities. Also, configuring filter sets is somewhat counter-intuitive, cumbersome, and error prone, and ongoing maintenance provides additional challenges.

Switches and routers have even been known to contain hardwired backdoor passwords allowing easy reconfiguration by a knowledgeable attacker. In addition, switches and routers are usually configured by sending plain text (not encrypted) passwords over the network. These passwords can be easily captured, or even guessed, and are reusable.

So, to summarize, switches and routers can be used to provide additional filtering and alarming but should never be relied on as a primary and dependable means of providing security to the business. They can be a formidable partner to firewalls if properly configured and, just as importantly, properly administered. Often, the people who set up the switches and routers and those who administer them are in different groups and have vastly different sets of skills. Typically, administrators have to maintain all sorts of

equipment, from switches to routers to servers to PCs to whatever else their company has sitting on the network, which makes it very difficult, if not impossible, to be experts in and keep current with each.

More information on classic firewall designs can be found in:

- *Building Internet Firewalls*, ISBN 1-5659-2124-0, by D. Brent Chapman, Elizabeth D. Zwicky, and Deborah Russell
- *Firewalls and Internet Security: Repelling the Wily Hacker*, ISBN 0-201-63357-4

2.3.3 Compartmentalized firewall environment design

A more secure and flexible approach suitable for complex environments is the compartmentalized firewall environment in which a single firewall system is equipped with more than two network interfaces and which can, therefore, mutually protect several different compartments, such as DMZs or security zones, from each other.

Compartments are a new name for security zones that are protected from each other by one firewall. We use it to differentiate this approach from the single, two-firewall DMZ or Secure Server Network. The design that has emerged in recent years, and may be considered state of the art, looks similar to what is shown in Figure 3 on page 22.

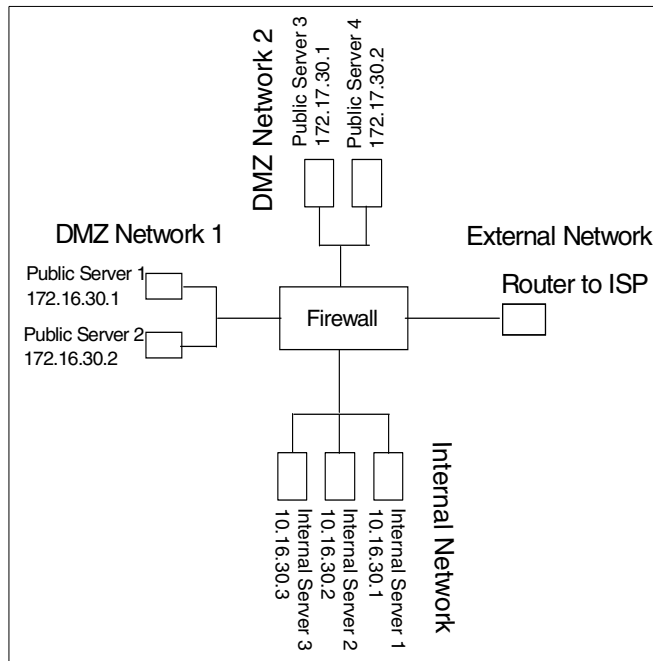


Figure 3. Compartmentalized firewall

The different compartments (External Network, DMZ Network 1, DMZ Network 2, Internal Network) each have their own physical network connected to the firewall through dedicated network cards. The firewall is now able to control all the traffic between these compartments. IP sniffing is almost useless to an attacker because they are able to see only the traffic within the one compartment they have broken into. Since the compartments are independent, a security breach in one of the attached systems (for example, Public Server 1) does not lead to a total compromise of the environment. The damage is restricted to the compartment where the compromised server is located. Careful planning of the compartments is required so that a partial security breach does not turn into a total security breach. You should strive to keep the compartments homogeneous. A Web server breach should not entitle the attacker to go after your mail servers. If possible, each type should be in separate compartments.

A properly-configured firewall should generate an alert if the intruder tries to leverage more access from the attacked system, for example, by having the Web server try to access the mail server. One of the main functions of a firewall is to generate alarms when suspicious activity is detected (for example, the Web server connecting to the mail server) because no security

device will ever be able to protect you against all possible threats. It should alert you when you are under attack and, therefore, enable you to react. Be aware that the attackers are always one step ahead since they have the ability to plan and launch many possible attacks, and only one has to be successful whereas the defender has to defend against an infinite number of possible attacks. Obviously, the defender can only hope to detect a successful attack early and initiate reconnaissance and countermeasures.

The compartmentalized approach entails somewhat higher complexity (more network cards in the firewall and more routing issues), but the additional security is well worth the cost. The real problem in this setup (as well as all other firewall designs) is that if the firewall is broken into, all security is lost. Therefore, it is extremely important to make the firewall as secure as possible. The operating system should be hardened and always up-to-date, and maintenance procedures should be well documented and always followed. Operating system insecurities due to low integrity, quality, incomplete hardening, and human error have been major factors in many security breaches.

Plan ahead carefully before installing additional software on a firewall system. Only software that was explicitly designed, tested, and audited for use in a firewall environment should be considered for use. The firewall should serve the single-purpose of being a firewall. Application software belongs on application servers, not firewalls. Also, it is a good idea to physically disconnect the firewall from the external network prior to starting any maintenance activities. A recent successful attack was launched while a firewall was down for maintenance activities.

Always keep the worst-case scenario in mind. A single bug in the software usually enables the attacker to execute arbitrary commands and ultimately gain full control of the system. A security breach of this nature is very bad when it occurs on a server that is isolated in a compartment, but it is disastrous when it occurs on the main firewall system.

The firewall system shown in Figure 3 on page 22 should perform network traffic control and nothing else. Either IP filtering or secure proxies or any combination of both can be used for that purpose. Both have their own advantages. IP filtering makes it very difficult to break into the firewall system because only IP packets are processed, and that task is carried out by kernel modules designed exclusively for that task. Proxies that are designed exclusively for firewall use can protect against certain rare network-level attacks because new IP packets are generated by the operating system instead of forwarding the original (and potentially harmful) IP packets. In addition, the number of TCP or UDP server programs on the firewall should

be kept to an absolute minimum because these kinds of programs are usually the weak points taken advantage of by potential intruders.

2.4 Securing firewalls

Think of a firewall as an application. To secure the firewall, you need to first secure the operating system (discussed in Chapter 8, “Securing AIX” on page 139); then, secure the firewall software. A firewall is a security enforcer and, as such, it must be very secure. It holds the keys to your kingdom, and, once it is compromised, the battle is, generally, lost.

The default option settings in firewalls are usually not sufficient to provide a secure environment, and additional configuration is typically required. Many of the common attack methods are not covered by the default settings, and new attack methods are being invented almost daily; so, firewall configuration should be viewed as an ongoing activity rather than just a one-time setup. Filter rules should be kept simple and short. Complicated rules may leave unanticipated holes in the rulebase. Constant auditing and log monitoring is crucial for early detection of a potential (or real) security breach.

Some firewalls have default ports on which they listen. For example, Check Point FireWall-1 listens on ports 256, 257, and 258 by default. To identify a Check Point FireWall-1, an attacker can use a port scanner to scan those ports to see if they are active. Therefore, to prevent easy detection, you may want to add filter rules to your routers to prevent any external network traffic to those ports.

2.5 Firewalls on AIX

Two common firewalls that run on IBM RS/6000 and the new IBM @server pSeries hardware are *Check Point FireWall-1* and *IBM Secureway Firewall*. Both products are hybrid firewalls that incorporate characteristics of both dynamic packet filter firewalls and application layer (proxy) firewalls. Check Point FireWall-1 is covered in Chapter 3, “Check Point FireWall-1” on page 25, and IBM Secureway Firewall is covered in Chapter 4, “IBM Secureway Firewall” on page 47.

A good source for comparing firewall products from different vendors is:

www.icsa.net/html/communities/firewalls/certification/vendors/index.shtml

Chapter 3. Check Point FireWall-1

Check Point FireWall-1 is the industry leader for firewalls. Utilizing Check Point's patented Stateful Inspection Technology and Open Platform for Secure Enterprise Connectivity (OPSEC), FireWall-1 integrates and centrally manages all aspects of network security.

With its position as market leader, a lot of other companies have integrated their products with FireWall-1. There are a number of products that complement FireWall-1, both from Check Point and their business partners. OPSEC is the binding platform that provides for this integration.

The Check Point homepage can be found at:

www.checkpoint.com

A good Web site for Check Point FireWall-1 troubleshooting information, automation scripts, and answers to general questions is:

www.phoneboy.com/fw1

3.1 FireWall-1 features

The strength of FireWall-1 is its ease of use in terms of the user interface for configuration, management, and logging. This makes the product intuitive to set up and use. The documentation is good, and there is additional support available on the Internet. Add-on software is also available to turn FireWall-1 into a total security solution.

FireWall-1 provides a central management console facility that enables you to administer a number of firewalls remotely. A central management console increases manageability by having a central location from which to implement security policies across multiple network entry points. Traffic to and from the management console is encrypted.

FireWall-1 supports multiple authentication schemes, including SecureID and RADIUS through internal or external authentication servers. Other user authentication schemes include operating system password, FireWall-1 password, S/Key and digital certificates. It also has built-in support for LDAP directory services to centralize user management.

Encryption can be used for VPN with support for Internet standards (IKE, IPSec, DES, 3DES, RSA, Diffie-Hellman, and so on). SecureRemote can be

used to provide IPSec compliant encryption and key management to extend VPN to remote users.

There are three security servers in FireWall-1 that provide for content security:

- **HTTP Security Server** protects Web servers against malicious Java and ActiveX applications as well as undesirable URLs.
- **FTP Security Server** protects FTP servers by controlling access to `get` and `put` commands.
- **SMTP Security Server** protects Mail servers by removing certain sendmail header fields.

Check Point FireWall-1 is available in several different configurations:

- Basic firewall (includes management module and firewall module)
- Firewall with VPN support (includes VPN for multiple site setup)
- Firewall with VPN and DES encryption - may be subject to US export regulations based on key strength
- Firewall engine (no management console; requires a separate management console)

Check Point FireWall-1 requires licenses to be installed for it to work properly. Licensing is based on the number of internal hosts to be protected (25, 50, 250, or unlimited). Ensure that the correct license package is obtained.

A license is also required to use the Motif GUI to connect to the FireWall-1 management console. This license is not required with the Windows GUI. The Motif license needs to be installed on the management console, and it should be tied to the management console's hostname or IP address as appropriate. In FireWall-1 4.0 and earlier, this license is free and can be requested off the Web. In FireWall-1 4.1, you will need to pay extra for it, and it should be ordered with the FireWall 4.1 product.

For performance, built-in server load balancing (five different algorithms) may be used to transparently load balance servers behind the firewall. FireWall-1 also has network address translation (NAT) using many-to-one and one-to-one configurations.

3.2 Complementary software for FireWall-1

Other products can be used with FireWall-1 to provide a total enterprise security solution. Bandwidth management (Floodgate), compression, and

hardware-based VPN-based acceleration are used to provide better performance, which is increasingly important in today's Web environment.

For high availability, products, such as Stonebeat, can be used for automating failover. With firewalls being a crucial component of the network, Stonebeat increases the availability of the server by having a standby machine automatically take over for the primary firewall after a system or network interface failure. This product is built on top of FireWall-1 and, as such, is tailored to this firewall. Other generic high availability software, such as HACMP, may also be used with FireWall-1 or IBM Secureway Firewall. However, the strength of Stonebeat is that it was created specifically with FireWall-1 in mind. It is easily installed and configured and provides full functionality for the firewall.

3.3 Securing FireWall-1

Check Point FireWall-1 installation does not include operating system hardening; so, you will need to do this yourself. This is an important step because the firewall software sits on top of the operating system, and your security is only as strong as its weakest link. For more information on operating system hardening, refer to Chapter 8, "Securing AIX" on page 139.

In this book, we do not cover how to install and initially configure Check Point FireWall-1. That subject is well covered on the Check Point Web site (www.checkpoint.com/support/technical/documents/index.html). This site is password protected, but the password is provided to both customers and resellers. Additional information can be found in the redbook, *Check Point FireWall-1 on AIX, A Cookbook for Stand-Alone and High Availability, SG24-5492*.

However, we do cover additional procedures for tightening Check Point FireWall-1 security. Specifically, we cover:

- Closing vulnerabilities during system start up
- Managing FireWall-1 logs
- Securing FireWall-1 default configurations
- Creating a useful rulebase
- Viewing connections
- Enabling other defense mechanisms

3.3.1 Closing vulnerabilities during system startup

There is a period of time during AIX start up where the network protected by the firewall is vulnerable. It occurs at the point in AIX start up when the network is available and the firewall policies have not yet been loaded. During this short time, attackers can attack the network by routing their packets through the firewall since no security policy is loaded. To protect your network, disable IP forwarding (`no -o ipforwarding=0`) on the firewall during this period to prevent any connections from going through. Once the firewall policies have loaded, IP forwarding can be enabled again.

IP forwarding (`ipforwarding`) is an AIX network option (`no`) that specifies whether or not packets will be forwarded from one network interface to another. In other words, should the AIX server act as a gateway. During the normal operation of a firewall, IP forwarding must be allowed since it is a gateway. So, as already mentioned, you need to turn IP forwarding off during the vulnerable period and on during normal operation. At system start up, the default value of IP forwarding is 0 (disabled). After the firewall policies have loaded, turn it on by changing it to 1 (enabled).

For example, you can have a script, such as the following, executed out of `/etc/inittab`. Be sure to comment out the start up of FireWall-1 in `/etc/rc.net`, and make certain that `ipforwarding` has not been changed from its default of 0 (disabled) in `/etc/rc.net` or any of the other start up files, such as `/etc/rc.tcpip` or `/etc/rc.local`.

```
#!/bin/ksh

LOG=/tmp/log/console
FWDIR=/usr/lpp/FireWall-1/
export FWDIR

/usr/lpp/FireWall-1/bin/fwstart > $LOG 2>&1

/usr/sbin/no -o ipforwarding=1
echo "Enabling IP forwarding" >> $LOG 2>&1
/usr/sbin/no -a | grep ipforwarding >> $LOG 2>&1

exit 0
```

Also, as a side note, the portmap service is started by `/etc/rc.tcpip` during system start up. You may have already disabled portmapper as part of your operating system hardening. However, if you plan to use the Motif GUI, you will need to reenab portmapper because Motif relies on the portmap service for port allocations.

3.3.2 Managing FireWall-1 logs

When setting up FireWall-1, it is a good idea to create a separate filesystem for the logging directory, such as `$FWDIR/log` or `/usr/lpp/FireWall-1/log`. This separate filesystem prevents the `/usr` filesystem from filling up because of firewall logging activity, and it eases the management of the log directory. The size of this filesystem depends on the amount of traffic that goes through the firewall, the level of verbosity you have configured for the firewall logging activity, and the amount of historical logging data you want to keep.

Log file maintenance is best controlled through crontab. FireWall-1 provides a log switching option (`fw logswitch`), which should be run periodically to ensure that the logs do not grow too large. For example, to switch logs daily at 2:00 a.m., the following entry can be added to root's crontab:

```
0 2 * * * /usr/lpp/FireWall-1/bin/fw logswitch > /dev/null 2>&1
```

3.3.3 Securing FireWall-1 default configurations

The FireWall-1 default options make it quick and easy to install and use the product. However, you may want to further tighten security by removing the defaults and creating rules especially for the things you really need. To turn on functionality, you can either enable the option in the Properties Setup menu or create a rule for it. In general, if you enable an option from the menu, it does not enable logging for that option. If, instead, you create a rule for it, it will give you the option to turn on logging. Therefore, better control is achieved through the latter, rather than the former.

For example, by default, you can ping the firewall from anywhere. In most cases, you do not want to allow that; so, you disable the default ICMP option. You then create a new rule (with logging turned on) for this in which you specify the trusted hosts that are allowed to ping the firewall. All other hosts are forbidden. This flexibility gives you a high granularity of control over the security level on the firewall, and the logging facility gives you an audit trail for each of the rules.

Upon installation, there are a number of services that are allowed by default, independent of rulebase. They are found in the Properties Setup menu (from the main menu bar, click **Policy**, and then select **Properties**).

We recommend that you enable the following two options only:

Accepting UDP Replies

The Accepting UDP Replies option enables replies for two-way UDP communications. This is necessary if you want the firewall to track UDP

communications. Having Check Point track the state of UDP communications is a good idea. It adds state to an otherwise stateless connection. The other option is to create separate rules for UDP replies, but this does not provide the advantage of state.

Accepting Outgoing Packets

The Accepting Outgoing Packets option is used to enable outgoing packets initiated from the firewall or routed from the firewall to leave the firewall. The assumption here is that all packets that come from the firewall are trusted.

Figure 4 shows the Properties Setup menu with the Accepting UDP Replies and Accepting Outgoing Packets options turned on.

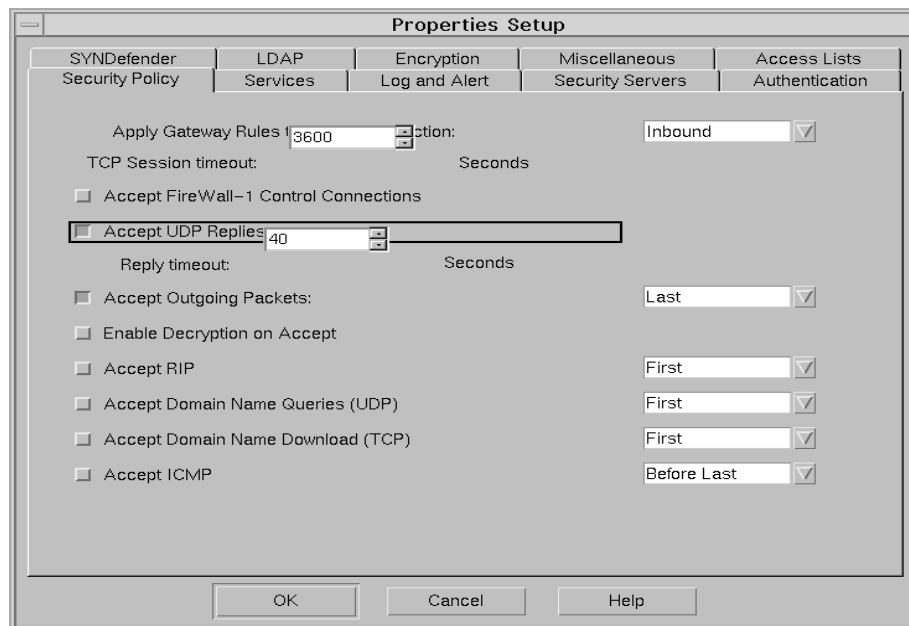


Figure 4. Security Policy tab in Properties Setup

We recommend that you disable the remaining options for better security:

Accept FireWall-1 Control Connections

The Accept FireWall-1 Control Connections option needs to be enabled if you do remote firewall management. If the management console and the firewall daemons are on the same machine, you may disable this option. However, if they are on separate machines, this option needs to be enabled (or

equivalent rules need to be created in the rulebase). Disabling this feature disables access to ports 256, 257, and 258.

Enable Decryption on Accept

The Enable Decryption on Accept option will decrypt incoming encrypted packets even if there is no encryption rule. This option should be disabled unless you are using encryption.

Accept RIP

RIP (Routing Information Protocol) is the protocol used by the `routed` daemon for dynamic routing. The other commonly used dynamic routing protocol is OSPF (Open Shortest Path First). The Accept RIP option should be enabled only if you are using RIP (not OSPF). Dynamic routing has its insecurities and should be disabled unless you are certain it is properly configured.

Accept Domain Name Queries (UDP)

The Accept Domain Name Queries (UDP) option is used to enable or disable DNS queries. Rather than enabling this option, you should create a separate rule for it. A rule gives you better control and lets you turn on logging.

Accept Domain Name Download (TCP)

The Accept Domain Name Download (TCP) option pertains to DNS nameserver zone transfers. Rather than enable this globally, you should create individual rules for the DNS servers that need to do zone transfers.

Accept ICMP

The Accept ICMP option is used to enable or disable ICMP protocol (things, such as `ping` and `traceroute`). If you want to take advantage of stateful inspection on ICMP, you have to turn this option on. (Refer to the www.phoneboy.com/fw1 FAQ for more information on stateful inspection).

There are various ICMP types, and you should allow only the types you need, such as echo request (8), echo reply (0), destination unreachable (3), and time exceeded (11). Some other types, such as timestamp request (13), address mask request (17), and redirect (5), may be dangerous and not necessary when initiated from the Internet. For more information about the ICMP types, refer to the redbook, *TCP/IP Tutorial and Technical Overview*, GG24-3376.

Additional options that need to be configured are under the Services tab of the Properties Setup menu. We recommend that you disable all four options:

Enable FTP PORT Data Connections

The FTP protocol uses two connections, one for control (port 21) and one for data (port 20). When you enable the FTP services in the rulebase, it only enables control port (21) access. To enable commands, such as `ls`, `get`, `put`, and so on, to run on the data port (20), you need to enable the Enable FTP PORT Data Connections option.

Enable FTP PASV Connections

The Enable FTP PASV Connections option is used for passive FTP. See [****section X****](#) on passive FTP to decide if you need this. A large number of FTP servers on the Internet require passive connection; so, you need to enable this option if you want to allow FTP.

Enable RSH/REXEC Reverse stderr Connections

The Enable RSH/REXEC Reverse stderr Connections option is used to allow `rsh` and `rexec` protocols to open a reverse connection for errors. It should be disabled if you are not allowing `rsh` and `rexec` protocol (see Chapter 8, “Securing AIX” on page 139, for information about OS hardening)

Enable RPC Control

The Enable RPC Control option is used for RPC dynamic port allocation and should be disabled unless required. RPC should not be allowed when facing the Internet as it has many security problems (see Chapter 8, “Securing AIX” on page 139, for information about OS hardening).

Figure 5 on page 33 shows the Services tab of the Properties Setup menu with the Enable FTP PORT Data Connections, Enable FTP PASV Connections, Enable RSH/REXEC Reverse stderr Connections, and Enable RPC Control options turned off.

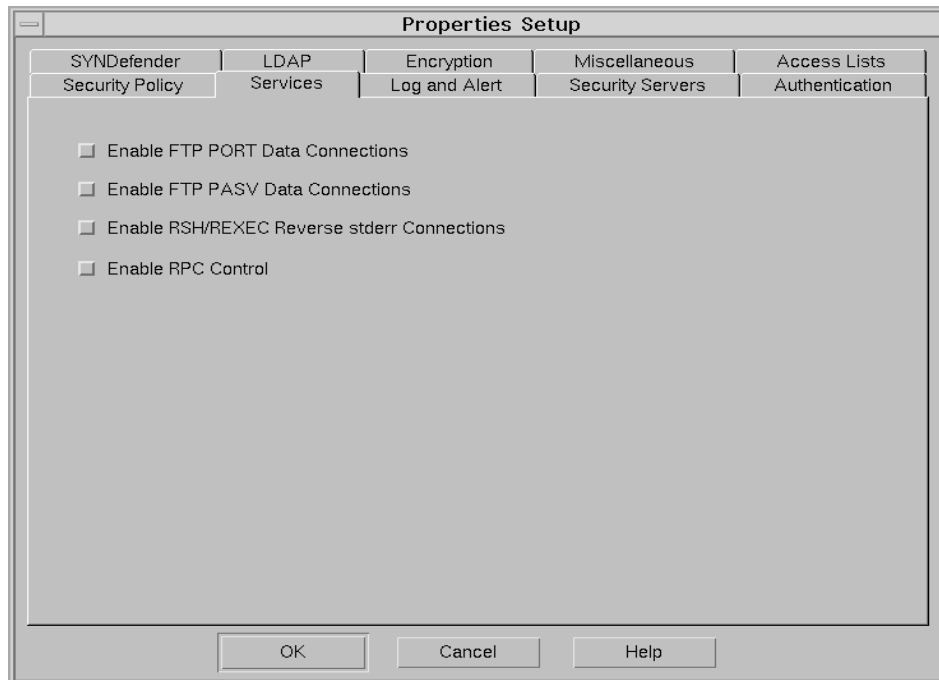


Figure 5. Services tab in Properties Setup

Lastly, there is an option under the Log and Alert tab of the Properties Setup menu that needs to be configured.

Set the IP Options Drop Track option to either *Alert* or *Log*. By default, FireWall-1 drops IP packets with IP options. These packets are frequently used by attackers; so, we recommend that you select *Alert* as an early warning mechanism for potential attacks.

Figure 6 on page 34 shows the Log and Alert tab of the Properties Setup menu with the IP Options Drop Track set to *Alert*.

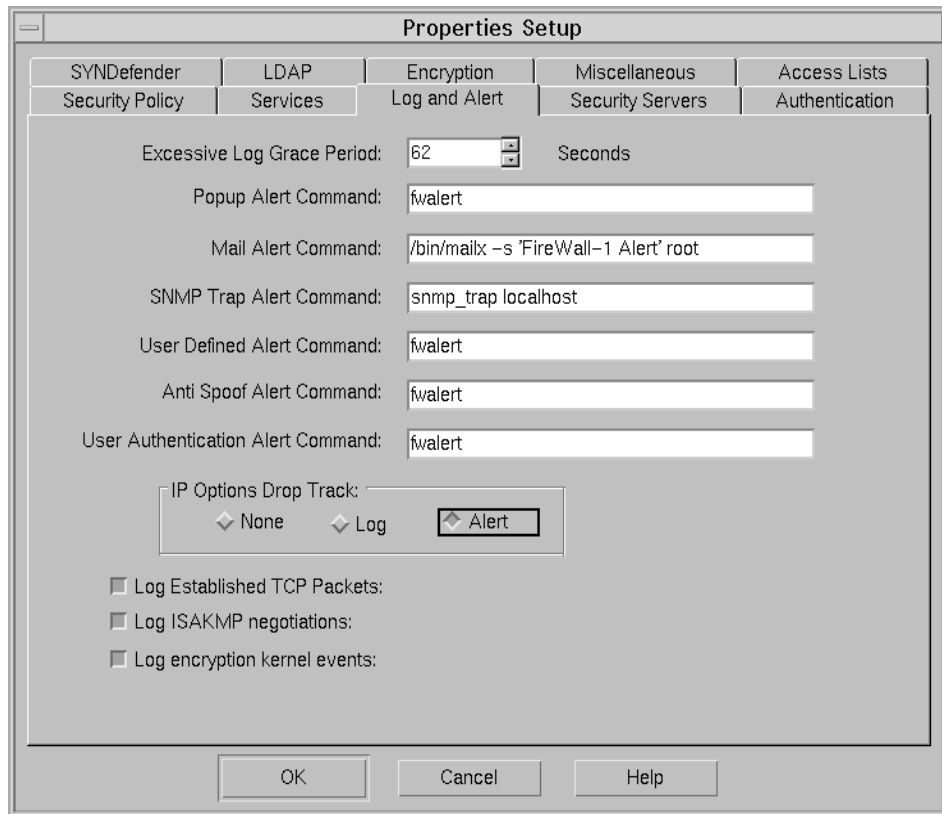


Figure 6. Log and Alert tab in Properties Setup

3.3.4 Creating a useful rulebase

Our example rulebase (shown in Figure 7 on page 36) is for a rather simple setup and is only meant to illustrate some principles. There are nine rules, processed sequentially from top to bottom, in our rulebase. We explain them in order of importance:

- Rule 9: The first rule you should set up is actually the last rule in the ruleset and is considered the *cleanup rule*. It is used to drop all packets which have not been accepted by any of the other rules. Check Point FireWall-1 has an *implicit drop rule* that, essentially, does the same thing as the cleanup rule. The difference between the cleanup rule and the implicit drop rule is that with the cleanup rule, you can log all traffic that is dropped. This logging serves as an important early warning device in that it can alert you to a potential attacker snooping around your network.

- Rule 5: This is considered the *stealth rule* because it drops all connections that have the firewall as the destination. This rule prevents attackers from connecting to the firewall or even pinging the firewall. Notice that you should log this rule as well as drop the packets, but do not select reject because it will send an RST back to the originator.
- Rule 4: Set up this rule to prevent excessive logging. NetBIOS (NBT) consists of nbname, nbssession, and nbdatagram service traffic, which may be dropped (and not logged) since it is voluminous and does not pose a security risk.
- Rule 1: Add this rule to allow an admin client to remotely connect to the firewall for administrative purposes. That client can also ping the firewall and vice versa for network debugging purposes. You may remove echo-reply service and just allow echo-request service if you only want the client to ping the firewall and not the other way around. Alternatively, you can enable the Accept ICMP option as discussed (this is not recommended).
- Rules 6,7, and 8: These rules are for Web and mail servers. You will need to modify them for your specific environment. For compartmentalized firewalls that have connections to the internal network as well as the DMZ and the Internet, you should include rules to prevent traffic from the DMZ or the Internet from reaching the internal network. All such attempts should be logged because they may be an indication of attempted spoofing. Remember also that the DMZ is considered semi-secure and should not be trusted completely. Only applications that need to connect to internal servers should have such connectivity allowed in the rulebase.
- Rule 2: Set up this rule if the firewall is also a DNS server that serves the Internet. Some exploits have been known to use the UDP port 53 (DNS port). Therefore, enable it only if it is absolutely required. If the DNS server serves only internal users, you should change the Source to DMZ or Internal, or, better yet, move the DNS server off of the firewall. Notice that this rule must come before the stealth rule.
- Rule 3: Only set this rule up if you have a secondary DNS server that needs to do zone transfers.

Figure 7 on page 36 shows a useful rulebase.

| No. | Source | Destination | Service | Action | Track | Install On | Time |
|-----|-----------|-------------|---|--------|-------|------------|------|
| 1 | admin-cli | firewall | echo-request echo-reply FireWall1 | accept | Long | Gateways | Any |
| 2 | Any | firewall | domain-udp | accept | | Gateways | Any |
| 3 | dns-slave | firewall | domain-tcp | accept | | Gateways | Any |
| 4 | Any | firewall | NBT | drop | | Gateways | Any |
| 5 | Any | firewall | Any | drop | Long | Gateways | Any |
| 6 | Any | webserver | http https | accept | Long | Gateways | Any |
| 7 | DMZ | Any | http https | accept | Long | Gateways | Any |
| 8 | Any | mailserver | smtp | accept | Long | Gateways | Any |
| 9 | Any | Any | Any | drop | Long | Gateways | Any |
| - | FW1 Host | Any | Any | accept | | Gateways | Any |

Figure 7. Useful rulebase

Options chosen in the Properties Setup menu are already part of the rulebase. To view the complete list of rules, including those from the Properties Setup menu, click **View**, and then select **Pseudo-rules**. As shown in Figure 7, pseudo-rules have no numbers and appear in yellow.

3.3.5 Viewing connections

To view the current connection table, you can use the FireWall-1 command:

```
#fw tab -t connections -u
```

There is a PERL script available from www.enteract.com/~lspitz/fwtable.txt that lists the current connection table in a much more usable format:


```
# perl ./fwtable.pl
---- FW-1 CONNECTIONS TABLE ----

Src_IP      Src_Prt  Dst_IP      Dst_Prt  IP_prot  Kbuf  Type   Flags      Timeout
9.12.0.50   32782    9.12.0.50   53        6        0     16385  ffff0400  3176/3600
9.12.0.50   32780    9.12.0.18   23        6        0     16385  ffff0600  2440/3600
9.12.0.50   846      9.12.0.50   755       6        0     16385  ffff0600  3433/3600
9.12.0.50   784      9.12.0.50   755       6        0     16385  ffff0600  3484/3600
9.12.2.168  1971     9.12.0.50   23        6        0     16385  ffffff00  3600/3600
```

3.3.6 Enabling other defense mechanisms

There are a number of ways to make Check Point FireWall-1 more secure. We recommend that you enable SYNDefender and IP spoofing protection, but do not use Fast Mode TCP. We also recommend that you consider implementing Intrusion Detection. Each of these is described in the following sections.

3.3.6.1 SYNDefender

The SYN attack falls under the category of *denial of service attack*. Before going into the details of the SYN attack, we need to cover how a typical TCP connection is established as follows:

1. A client attempting to connect to a server sends a SYN packet (TCP packet with the SYN bit set) to the server.
2. The server replies with a SYN/ACK packet.
3. The client receives the SYN/ACK packet from the server and sends an ACK packet back.

This is known as a 3-way handshake, and, upon successful completion, the connection is considered established. (Note that with stateful inspection, the firewall will monitor this handshake process as an active onlooker.)

Here is how the SYN attack works:

1. The attacker uses a forged and non-reachable IP address and sends several SYN packets to the server.
2. The server responds with the appropriate SYN/ACK packets, but, since the sender's IP is a non-reachable address, the 3-way handshake cannot complete.
3. The connections remain pending and will not be cleared for the duration of the TCP timeout (unless the `clear_partial_conns` network option is set at

the OS level), which effectively prevents legitimate clients from connecting to the server on that service port.

The key to the SYN attack is that the source IP must not be reachable. If it is, the server sending the SYN/ACK will get an RST packet in return. This is because the legitimate client (whose IP address has been stolen) will detect that the SYN/ACK packet is not in response to a SYN packet that it had sent, and will reply with an RST packet to reset the server connection (which defeats the attack). For an in-depth description of the attack, go to www.fc.net/phrack/files/p48/p48-13.html.

Check Point FireWall-1 has two built-in defenses against SYN attacks. They are the SYNDefender Gateway and the SYNDefender Passive Gateway. You can set them on the SYNDefender tab of the Properties Setup menu (see Figure 8 on page 39). For the following explanations, assume the server is behind the firewall, and the client is in front of it.

SYNDefender Gateway Mode works as follows:

1. The client sends the SYN packet to the server.
2. The server sends the SYN/ACK packet back to the client.
3. FireWall-1 sends the ACK packet to the server on behalf of the client.
4. The client sends the ACK packet to the server, but it is absorbed by FireWall-1.
5. The connection between client and server is established.
6. If the client does not send the ACK packet (event 4) within the timeout period configured in FireWall-1 (Figure 8), the firewall sends an RST packet to the server to tear down the connection. This is the recommended method to protect against SYN attacks.

SYNDefender Passive SYN Gateway works as follows:

1. The client sends the SYN packet to the server.
2. The server sends the SYN/ACK packet back to the client, but it is intercepted by FireWall-1.
3. FireWall-1 resends the SYN/ACK packet to client.
4. The client sends the ACK packet back to server.
5. The connection between the client and server is established.
6. If the client does not send the ACK packet (event 4) to the server within the timeout period configured in FireWall-1 (see Figure 8 on page 39), the firewall sends an RST packet to the server to close its pending connection.

During the timeout period, the pending connection table on the server still has this connection in it. In other words, from the server's point of view, the 3-way handshake has never completed.

Figure 8 shows the SYNDefender tab in the Properties Setup screen.

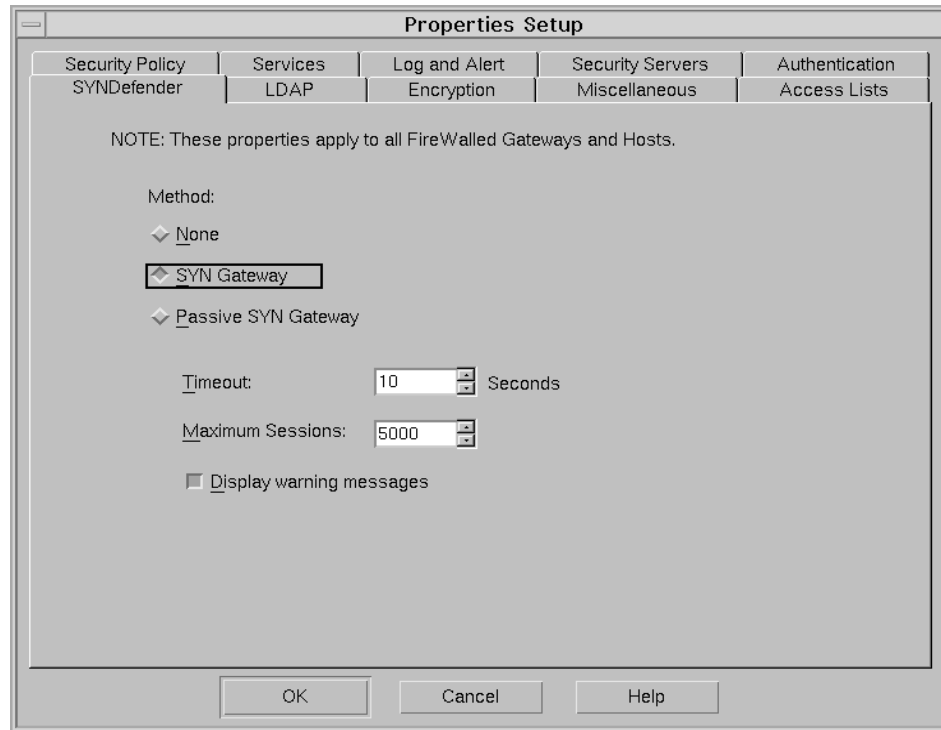


Figure 8. SYNDefender tab in the Properties Setup screen

3.3.6.2 IP spoofing protection

Anti-spoofing protects against forged packets that claim to have originated from an internal IP address (a spoofed IP packet). Anti-spoofing ensures that the packets are coming or going on the correct network interfaces. For example, packets with a source IP address from the internal network should be coming into the firewall from the network card on the internal interface. If packets carrying the source IP address from the internal network come into the firewall from the Internet, it is likely due to a spoofed packet from someone trying to attack your network.

You need to know which networks are reachable from which interface. You then need to associate each group of addresses with the appropriate network

interface. Perform the following steps to set up IP spoofing protection in FireWall-1 (refer to Figure 9 on page 40):

1. From Network Objects, select **firewall**.
2. Select the **Interfaces** tab.
3. Select an interface (in our case, the tr0 interface is the external interface).
4. From Spoof tracking, select **Log** or **Alert**.
5. From Valid Addresses, select **This net**, **Others**, or **Specific**, depending on how your network is configured, as shown in Figure 9. (More information follows.)

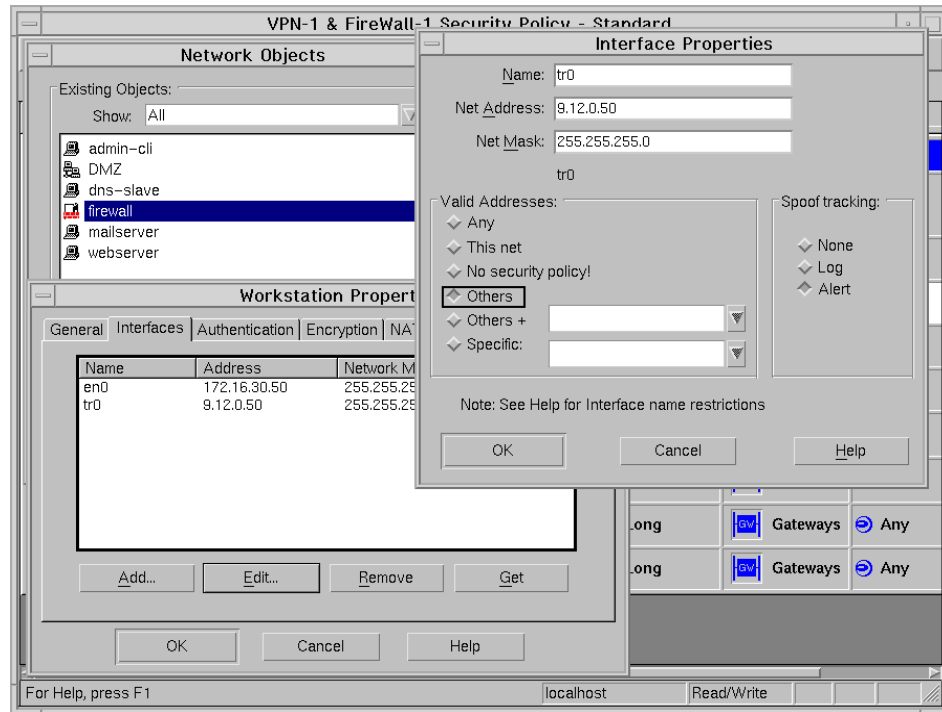


Figure 9. Setting up IP spoofing

The Valid Addresses selection specifies which addresses are allowed to be on the chosen interface. In other words, IP addresses that are not considered to be spoofed. The option you choose depends on whether or not you are using network address translation (NAT) at the firewall.

If you are not using NAT, use the following for your Valid Addresses selections as shown in Table 1.

Table 1. Valid Addresses without NAT

| Interface | Valid addresses |
|----------------------------|-----------------|
| External (facing Internet) | Others |
| Internal | This net |

If you are using NAT, use the following for your Valid Addresses selections as shown in Table 2.

Table 2. Valid Addresses with NAT

| Interface | Valid addresses |
|----------------------------|-----------------|
| External (facing Internet) | Others |
| Internal | Specific |

In the case of NAT and an Internal Interface, you have to specify a group of addresses for Specific. For the servers on the internal network, use the IP address for the internal network, and the NAT IP addresses for the servers on that network. For example, Figure 10 on page 42 illustrates a with NAT enabled for Web Server 1 (Public Server 1), Web Server 2 (Public Server 2), and Mail Server (Public Server 3). The real IP address (configured on the network card on the server) and the NAT IP address (publicly known IP address) are listed in the figure. When using NAT, the group for Specific consists of the internal network address (172.16.30.0) and the public IP addresses 203.12.12.1, 203.12.12.2, and 203.12.12.3.

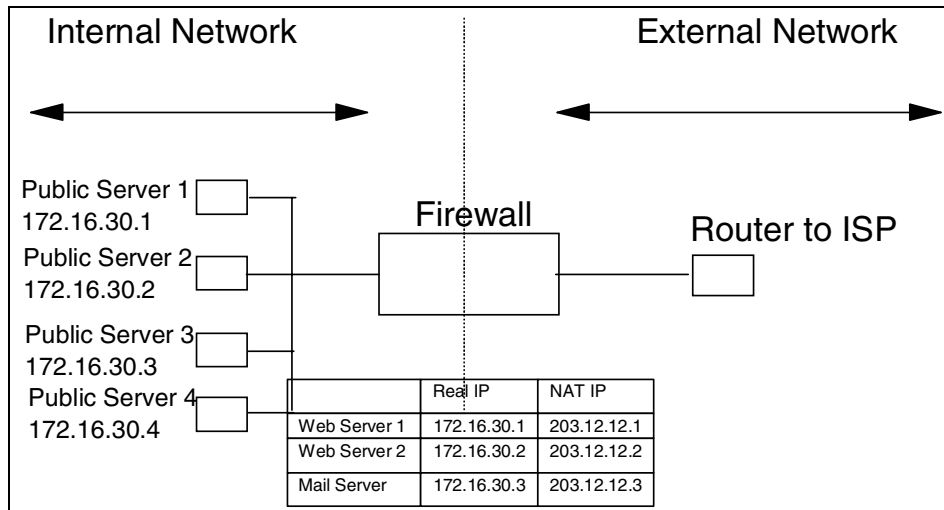


Figure 10. Illustrating IP spoofing with NAT

3.3.6.3 Fast Mode TCP

For faster throughput, the Fast Mode TCP option can be used. We do not recommend that you do this. Fast Mode works by allowing all non-SYN/NO-ACK packets through the firewall. The firewall assumes that all such packets are part of an established TCP connection and does no further checks on them. The assumption is that if these packets are not part of an established TCP connection, the end servers will drop the packets anyway because they are out of sequence. SYN packets are still checked with the rulebase, but successful connections are not logged into the connections table (removing the stateful inspection). This reduces the security awareness of the firewall and should not be used.

To check the Fast Mode setting, from the main menu bar, select **Manage**, then, under Services, select **TCP** (as shown in Figure 11 on page 43). Notice that Fast Mode only applies to the TCP protocol.

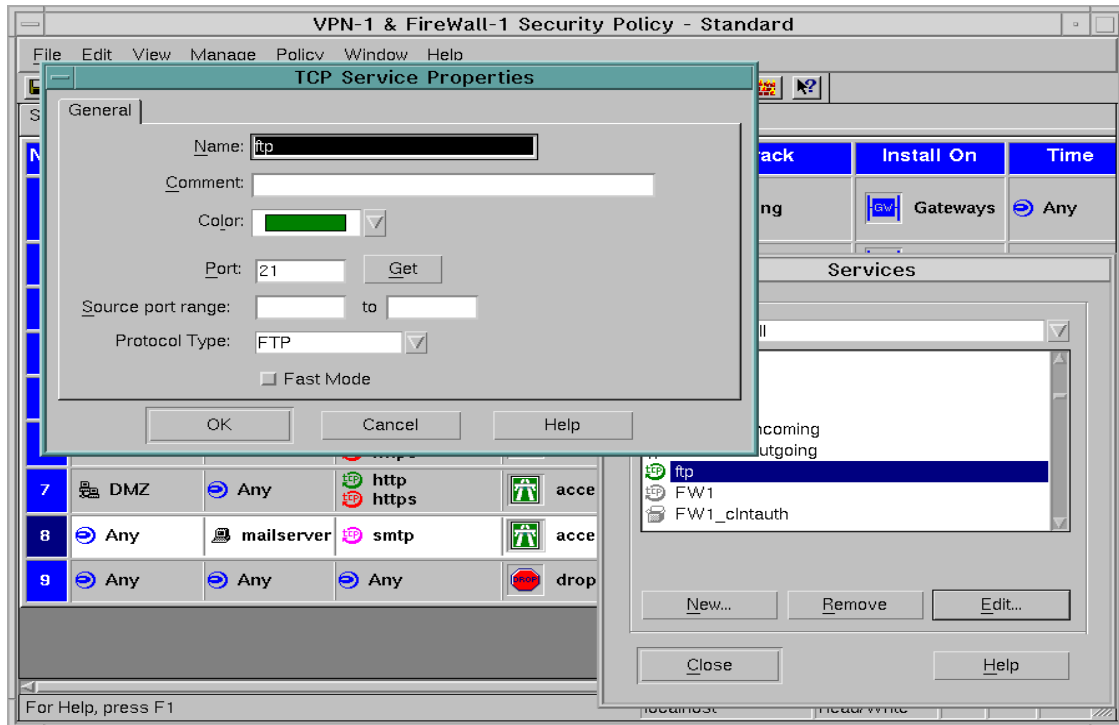


Figure 11. Fast Mode TCP

3.3.6.4 Intrusion detection on Check Point FireWall-1

There is a very good article on how to implement intrusion detection on Check Point FireWall-1 at www.enteract.com/~lspitz/intrusion.htm. We will not be discussing it further here.

3.4 List of ports that Check Point FireWall-1 uses

Check Point FireWall-1 has a list of ports that it uses by default. If there are ports that you are not using, you can disable them (just as you would normal TCP/IP ports in the `/etc/inetd.conf` file).

To disable authentication related services, modify the appropriate entries in the `$FWDIR/conf/fwauthd.conf` file or in the `$FWDIR/conf/fwopsec.conf` file. The latter file applies only to the SAM and LEA services (described later). Ports 256, 257, and 258 are controlled from the Properties Setup menu with the Accept FireWall-1 Control Connections option discussed in Section 3.3.3, "Securing FireWall-1 default configurations" on page 29.

To view all active ports on your system, use:

```
# netstat -af inet
```

If you are not sure which application is bound to a particular port, try using the `lsof` tool instead (see Section 6.5, "List Open Files (`lsof`)" on page 103, for more information). Among other things, this tool can tell you which applications are bound to which ports. From that information, you can then determine which ports to leave active and which ports to disable.

The following is the list of ports that the firewall uses. (Note: This information is taken verbatim directly from the www.phoneboy.com/fw1/ Web site.)

- **TCP Port 256** is used for three important things:
 - a. Exchange of CA and DH keys in FWZ and SKIP encryption between two FireWall-1 Management Consoles.
 - b. SecuRemote build 4005 and earlier uses this port to fetch the network topology and encryption keys from a FireWall-1 Management Console.
 - c. When installing a policy, the management console uses this port to push the policy to the remote firewall.
- **TCP Port 257** is used by a remote firewall module to send logs to a management console.
- **TCP Port 258** is used by the `fwpolicy` remote GUI.
- **TCP Port 259** is used for Client Authentication.
- **UDP Port 259** is used in FWZ encryption to manage the encrypted session (SecuRemote and FireWall-1 to FireWall-1 VPNs).
- **UDP Port 260** and UDP Port 161 are used for the SNMP daemon that Check Point FireWall-1 provides.
- **TCP Port 264** is used for Secure Client (SecuRemote) build 4100 and later to fetch network topology and encryption keys from a FireWall-1 Management Console.
- **TCP Port 265**, according to the 4.1SP1 `objects.C`, is labeled "Check Point VPN-1 Public Key Transfer Protocol." It is probably used by FireWall-1 to exchange public keys with other hosts.
- **UDP Port 500** is used for ISAKMP key exchange between firewalls or between a firewall and a host running Secure Client.
- **TCP Port 900** is used by FireWall-1's HTTP Client Authentication mechanism.

- **TCP Ports above 1024** are generally any Security Servers that are active. The actual ports used by these servers will vary.
- **TCP Port 18181** is used for CVP (Content Vectoring Protocol, for anti-virus scanning).
- **TCP Port 18182** is used for UFP (URL Filtering Protocol, for WebSense and the like).
- **TCP Port 18183** is used for SAM (Suspicious Activity Monitoring, for intrusion detection).
- **TCP Port 18184** is used for Log Export API (LEA).

Chapter 4. IBM Secureway Firewall

IBM Secureway Firewall has been protecting IBM for more than 10 years. It is a hybrid of three firewall architectures: Filtering, proxy, and circuit-level and includes a built-in Web proxy server and a SOCKS server. In 1999, Infoworld named it as their product of the year:

www.infoworld.com/supplements/99poy_win/99poy_s.html

Online information about IBM Secureway Firewall can be found at:

- <http://www-3.ibm.com/security/>
- <http://www-4.ibm.com/software/security/firewall/>

4.1 IBM Secureway Firewall features

IBM Secureway Firewall is comprised of a suite of tools that can be used individually or in combination depending on your needs and environment. A brief overview is given here.

IBM Secureway Firewall ships with a tool called Network Security Auditor (NSA). NSA can be used to proactively scan any hosts on the network (including the firewall) for potential security vulnerabilities. It is an excellent tool for identifying potential problems before they become breaches. The NSA tool alone provides reason enough for purchasing the IBM Secureway Firewall. Installation and configuration of NSA is covered in Section 4.4, "Network Security Auditor (NSA)" on page 49.

IBM Secureway Firewall comes with an embedded Web proxy server (via the enhanced version of Web Traffic Express). The proxy server includes added support for industry-standard reporting utilities, support for persistent Web sessions of HTTP 1.1, and support for URL blocking. This means that the full HTTP proxy functionality is within the firewall itself with no need for additional software.

IBM Secureway Firewall includes Virtual Private Network (VPN) support based on the IPSec standards. VPNs enable you to create secure tunnels between remote locations across the Internet. This reduces the need for expensive leased lines.

IBM Secureway Firewall comes with a two-user license Security Dynamics ACE Server (SecurID Protected) that is used for secure authentication.

The firewall has built-in support for email and pager notifications keyed off of specific, configurable events. This feature enables administrators to be notified of suspicious activities and to take action in the early stages of an attack.

Administration of multiple firewalls is simplified through a centralized administration facility. Real-time performance statistics and log monitoring are also provided.

The installation of the firewall hardens the operating system *automatically*. Non-important users and groups are removed, and non-essential services are disabled to reduce potential vulnerabilities. The firewall offers advanced email protection to help block unwanted email and prevent mail spoofing.

SOCKS Version 5 (V5) support is included for both TCP and UDP applications. Network address translation (NAT) support using many-to-one and one-to-one configurations is included as well.

4.2 Complimentary software for IBM Secureway Firewall

As with Check Point FireWall-1, IBM Secureway Firewall also has a list of products to complement it. IBM Secureway Network Dispatcher can be used for scalability, load-balancing, and failover of the firewall(s). Since users connect through the firewall, network dispatcher can be used to enhance performance by transparently splitting the load across multiple firewalls. More information can be found at:

www-4.ibm.com/software/network/dispatcher/

Another complimentary product is MIMESweeper, an SMTP, HTTP, and FTP content security and management solution. MIMESweeper examines application content and provides protection against malicious code and viruses. It is transparent to users and provides logging for further analysis. More information can be found at:

www.mimesweeper.integralis.com

Another useful product is Telemate.Net, which is used to translate large, abstract firewall log files into meaningful business reports. You can easily customize the reports and distribute them through email or translate them into HTML for publication on the intranet. Telemate.Net runs on Windows 95 or Windows NT and collects IBM Secureway Firewall log files. The firewall periodically FTPs the logs to the Telemate.Net server sitting behind the firewall. More information can be found at:

4.3 Firewall hardening

The IBM Secureway Firewall installation procedure includes a section on hardening. The hardening is an automated procedure that happens during firewall installation. This is in contrast to Check Point FireWall-1, which requires OS hardening as a manual step. To make sure that this automatic hardening is sufficient for your needs, refer to Section 2.3.2 of the redbook, *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855, for the complete list of steps that IBM Secureway Firewall takes to harden itself. Also, Chapter 8, "Securing AIX" on page 139 of this book, discusses in detail the manual procedures you can use to harden the operating system.

4.4 Network Security Auditor (NSA)

NSA is an extremely good tool for auditing your servers (AIX and others) for potential security vulnerabilities in the services that are running. The output from the NSA scan of your environment gives you insight into an attacker's perspective of your network. Review the results of the scan and plug the security holes proactively before an attacker strikes.

The NSA tool is essentially a port scanner, but it incorporates additional tests to look for vulnerabilities within the ports it scans. A single `nsa` command provides the entire output, thus, making it an easy tool to use.

For more information on NSA, see:

<http://www-4.ibm.com/software/security/firewall/about/>

4.4.1 Installing NSA

The required NSA fileset is `nsaauditor.base`. The NSA version may vary, depending on the version of IBM Secureway Firewall that you purchase. NSA Version 2.1.2 comes with IBM Secureway Firewall Version 4.1.

To install NSA:

```
TYPE          smitty installp
SELECT        Install and Update from ALL Available Software
TYPE          /dev/cd0 [ESC+4 or F4 to select the CD ROM drive]
TYPE          F4 or ESC+4 to list
```

SELECT find nsauditor.base (**F7** or **ESC+7** to select the fileset)

PRESS **Enter** three times to install

Ensure that the fileset is properly installed as follows:

```
# lppchk -v
# lspp -l nsauditor.base
Fileset                Level  State      Description
-----
Path: /usr/lib/objrepos
nsauditor.base         2.1.2.0  COMMITTED  Network Security Auditor
```

4.4.2 Using NSA

Various scan methods are defined in the file `/etc/nsa/scannerdefs` file. The types of scan are *default*, *baseline*, *medium*, *standard*, *fulltcp*, *complete*, and *firewall*. The extent of the port scans and tests varies depending on which type you choose. Details of each type are in the `scannerdefs` file and in Appendix A. The *firewall* type provides the most extensive scan, and is the type used in this book. The redbook, *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855, is also a good source of information about the NSA tool.

Important

Some companies have strict restrictions against doing such scanning. Ensure that you have the necessary authorization prior to running these scans. Also, you may want to play it safe by ensuring that backups are done prior to scanning, as crashes may occur as a result of the scans.

The NSA documentation is located in the `/usr/lpp/nsauditor/doc/` directory. You will be unable to scan until you obtain valid licenses. Once obtained, these licenses must be located in the password-protected `/etc/nsa/license.nsa` file.

To scan a server, use:

```
# nsa scan -d /tmp/<output_file> --scantype=firewall <host_to_scan>
```

To view the output of the scan, use:

```
# nsa report -d /tmp/<output_file>
```

The report can also be output in HTML format for viewing with a Web browser.

4.4.3 Interpreting NSA output

An NSA scan produces considerable output. The report can be divided into these categories:

- Configuration settings
- Potential vulnerabilities
- Dangerous services
- Information leaks

For each of these categories, a description of the potential security loopholes is given. This is very helpful for administrators trying to fix these potential loopholes and minimize the security risks.

With a list of services running, you can decide if you need those services. Default installations will have some services enabled by default (for example SNMP). It is better to turn on only those services that you are certain that you need, and turn the rest off. For those services that you need, ensure that they are at their most secure configuration and latest security patch level. See Chapter 8, “Securing AIX” on page 139, for more information.

The problem is that, sometimes, administrators are unaware of such services running on the server, for example, when someone installs a new application that adds new, active ports. Whatever the case, it is a good idea to audit your servers regularly to ensure that the security levels remain the same and to alert you if any changes are made.

The following screen contains a sample of an NSA scan that was run on the control workstation of an SP complex.

Network Services Audit Report

Network Services Audit Report

Report Date: Friday, September 01, 2000 17:24

o Name: sp5cw0

Operating System: UNIX IBM AIX 4.3

Audit Date: `Friday, September 01, 2000 16:59'

Auditor: `root@arthur'

Security Audit Summary

- o Configuration Settings - 7
- o Potential Vulnerabilities - 2
- o Dangerous Servers and Services - 6
- o Information Leaks - 9

Security Audit Breakdown

Configuration Settings - 7

- o Access Control Configuration - 5
 - o Network File Sharing Configuration - 3

Potential Vulnerabilities - 2

- o NFS server vulnerabilities - 2

Dangerous Servers and Services - 6

- o Dangerous Network Servers - 6
 - o Dangerous Sun RPC servers - 2
 - o Other - 4

Information Leaks - 9

- o Information About User Accounts - 3
- o Information About System Resources - 6

Security Audit Findings

- o Configuration Settings
 - o Access Control Configuration
 - o Network File Sharing Configuration
 - o `~/spdata/sys1/install/pssplpp' is shared to everyone via NFS.
 - o `~/usr/sys/inst.images' is shared to everyone via NFS.
 - o `~/spdata/sys1/install/aix433/lppsource' is shared to everyone via NFS.
- o SNMP at UDP port 161 read community string is `public'.
 - o [177/UDP] XDMCP server accepts all management requests.
- o [SMTP at port 25] EXPN command is enabled.
- o [SMTP at port 25] VRFY command is enabled.

- o Dangerous Servers and Services
 - o Dangerous Network Servers
 - o TFTP is active on UDP port 69.
 - o TFTP server on port 69 serves file `.`.
 - o rshell is active on TCP port 514.
 - o rlogin is active on TCP port 513.
 - o Dangerous Sun RPC servers
 - o pcnfs (Sun RPC program 150001) is active on UDP port 32784.
 - o rstatd (Sun RPC program 100001) is active on UDP port 32780.
- o Potential Vulnerabilities
 - o NFS server vulnerabilities
 - o [32795/UDP] NFS mountd allows mount requests to come from arbitrary port.
 - o [32774/TCP] NFS mountd allows mount requests to come from arbitrary port.
- o Information Leaks
 - o Information About User Accounts
 - o rusers (Sun RPC program 100002) is active on UDP port 32781.
 - o [SMTP at port 25] EXPN command is enabled.
 - o [SMTP at port 25] VRFY command is enabled.
- o Information About System Resources
 - o portmapper (Sun RPC program 100000) is active on UDP port 111.
 - o SNMP is active on UDP port 161.
 - o nfsmountd (Sun RPC program 100005) is active on UDP port 32795.
 - o nfsmountd (Sun RPC program 100005) is active on TCP port 32774.
 - o portmapper (Sun RPC program 100000) is active on TCP port 111.
 - o rstatd (Sun RPC program 100001) is active on UDP port 32780.
- o Active Network Servers
 - o SSH is active on TCP port 22.
 - o X is active on TCP port 6000.
 - o SMTP is active on TCP port 25.
 - o telnet is active on TCP port 23.
 - o FTP is active on TCP port 21.
 - o rshell is active on TCP port 514.
 - o rexec is active on TCP port 512.
 - o rlogin is active on TCP port 513.
 - o SNMP is active on UDP port 161.
 - o TFTP is active on UDP port 69.
 - o XDMCP is active on UDP port 177.
 - o status (Sun RPC program 100024) is active on TCP port 746.
 - o aix4_rstatd (Sun RPC program 200001) is active on TCP port 755.
 - o calendar (Sun RPC program 100068) is active on TCP port 784.
 - o NFS (Sun RPC program 100003) is active on TCP port 2049.
 - o unknown (Sun RPC program 200006) is active on TCP port 2049.
 - o tttdserver (Sun RPC program 100083) is active on TCP port 32769.
 - o nlockmgr (Sun RPC program 100021) is active on TCP port 32775.
 - o nfsmountd (Sun RPC program 100005) is active on TCP port 32774.

- o portmapper (Sun RPC program 100000) is active on TCP port 111.
 - o unknown (Sun RPC program 300667) is active on TCP port 32770.
 - o portmapper (Sun RPC program 100000) is active on UDP port 111.
 - o calendar (Sun RPC program 100068) is active on UDP port 32785.
 - o status (Sun RPC program 100024) is active on UDP port 746.
 - o aix4_rstatd (Sun RPC program 200001) is active on UDP port 755.
 - o nlockmgr (Sun RPC program 100021) is active on UDP port 32818.
 - o NFS (Sun RPC program 100003) is active on UDP port 2049.
 - o nfsmountd (Sun RPC program 100005) is active on UDP port 32795.
 - o autofs (Sun RPC program 100099) is active on UDP port 32839.
 - o spray (Sun RPC program 100012) is active on UDP port 32783.
 - o pcnfs (Sun RPC program 150001) is active on UDP port 32784.
 - o rwall (Sun RPC program 100008) is active on UDP port 32782.
 - o rusers (Sun RPC program 100002) is active on UDP port 32781.
 - o rstatd (Sun RPC program 100001) is active on UDP port 32780.
- o Available Network Services
 - o Unauthenticated File Service
 - o TFTP server on port 69 serves file `.`.
 - o Shared File Systems
 - o `~/spdata/sys1/install/pssplpp` is shared to everyone via NFS.
 - o `~/usr/sys/inst.images` is shared to everyone via NFS.
 - o `~/spdata/sys1/install/aix433/lppsource` is shared to everyone via NFS.
 - o User Login Services
 - o SSH is active on TCP port 22.
 - o XDMCP is active on UDP port 177.
 - o telnet is active on TCP port 23.
 - o FTP is active on TCP port 21.
 - o rshell is active on TCP port 514.
 - o rexec is active on TCP port 512.
 - o rlogin is active on TCP port 513.
- o Operating system is `UNIX IBM AIX 4.3'.
 - o Server Version Strings
 - o [22/TCP] SSH server version is `Protocol 1.99; Server OpenSSH 2.1.1'.
 - o [25/TCP] SMTP server version is `Sendmail AIX4.3/8.9.3'.
 - o [21/TCP] FTP server version is `4.1 Sun Jul 9 18:28:14 CDT 2000'.
 - o SNMP at UDP port 161 read community string is `public'.
 - o Network Transport Information
 - o IP Transport Information
 - o Host responded to ICMP Echo Request.
 - o Port Scan Information
 - o TCP Port Scan Data
 - o The following TCP ports were scanned:
 - 21-23, 25, 109-111, 139, 143, 512-514, 746, 755, 784, 2049, 6000, 32769-32770, 32774-32775, 37661, 49690
 - o The following TCP ports were visible:
 - 21-23, 25, 109-111, 139, 143, 512-514, 746, 755, 784, 2049,

- 6000, 32769-32770, 32774-32775, 37661, 49690
- o The following TCP ports were active:
 - 21-23, 25, 111, 512-514, 746, 755, 784, 2049, 6000, 32769-32770, 32774-32775, 49690
- o The servers on these TCP ports could not be identified: 49690
- o The servers on these TCP ports terminated immediately: None
- o UDP Port Scan Data
 - o The following UDP ports were scanned: 69, 111, 137, 161, 177, 746, 755, 2049, 32780-32785, 32795, 32818, 32839
 - o The following UDP ports were visible: 69, 111, 137, 161, 177, 746, 755, 2049, 32780-32785, 32795, 32818, 32839
 - o The following UDP ports were active: 69, 111, 161, 177, 746, 755, 2049, 32780-32785, 32795, 32818, 32839
 - o The following UDP ports did not respond: None
 - o The servers on these UDP ports could not be identified: None
- o Sun RPC Registrations
 - o Sun RPC program 100000 (portmapper) is registered on TCP port 111.
 - o Sun RPC program 100083 (tttdbserver) is registered on TCP port 32769.
 - o Sun RPC program 300667 (unknown) is registered on TCP port 32770.
 - o Sun RPC program 100003 (NFS) is registered on TCP port 2049.
 - o Sun RPC program 200006 (unknown) is registered on TCP port 2049.
 - o Sun RPC program 100005 (nfsmount) is registered on TCP port 32774.
 - o Sun RPC program 100024 (status) is registered on TCP port 746.
 - o Sun RPC program 200001 (aix4_rstatd) is registered on TCP port 755.
 - o Sun RPC program 100021 (nlockmgr) is registered on TCP port 32775.
 - o Sun RPC program 1342177279 (unknown) is registered on TCP port 37661.
 - o Sun RPC program 1342177280 (unknown) is registered on TCP port 49690.
 - o Sun RPC program 100068 (calendar) is registered on TCP port 784.
 - o Sun RPC program 100000 (portmapper) is registered on UDP port 111.
 - o Sun RPC program 100001 (rstatd) is registered on UDP port 32780.
 - o Sun RPC program 100002 (rusers) is registered on UDP port 32781.
 - o Sun RPC program 100008 (rwall) is registered on UDP port 32782.

```

o Sun RPC program 100012 (spray) is registered on UDP port
  32783.
  o Sun RPC program 150001 (pcnfs) is registered on UDP port
    32784.
  o Sun RPC program 100068 (calendar) is registered on UDP port
    32785.
  o Sun RPC program 100003 (NFS) is registered on UDP port 2049.
  o Sun RPC program 200006 (unknown) is registered on UDP port
    2049.
  o Sun RPC program 100005 (nfsmount) is registered on UDP port
    32795.
  o Sun RPC program 100024 (status) is registered on UDP port
    746.
  o Sun RPC program 200001 (aix4_rstatd) is registered on UDP
    port 755.
  o Sun RPC program 100021 (nlockmgr) is registered on UDP port
    32818.
  o Sun RPC program 100099 (autofs) is registered on UDP port
    32839.

o SNMP Variables Retrieved
  o SNMP/161 variable sysDescr value is `RISC System/6000
    Architecture\nMachine Type: 0x0100 Processor id:
    000509306700\nBase Operating System Runtime AIX version:
    04.03.0003.0000\nTCP/IP Client Support version:
    04.03.0003.0000'.
  o SNMP/161 variable sysObjectID value is
    `1.3.6.1.4.1.2.3.1.2.1.1.3'.
  o SNMP/161 variable sysName value is `sp5en0'.

o Active Users
  o rusers/32781 shows the following users logged on:

root <-
root <- tot75
root <- :0.0
root <- tot75
root <- :0.0
root <- :0.0
root <-
root <-
root <-
root <-

o Server Banners
  o [22/TCP] SSH server banner -

SSH-1.99-OpenSSH_2.1.1

o [25/TCP] SMTP server banner -

220 sp5en0.msc.itso.ibm.com ESMTP Sendmail AIX4.3/8.9.3/8.9.3;
Fri, 1 Sep 2000 16:59:24 -0400

```

o [23/TCP] telnet server banner -

telnet (sp5en0)

AIX Version 4

(C) Copyrights by IBM and by others 1982, 1996.

login:

o [21/TCP] FTP server banner -

220 sp5en0 FTP server (Version 4.1 Sun Jul 9 18:28:14 CDT 2000) ready.

Chapter 5. Secure remote access

In this chapter, we discuss how to implement security measures for remote access. Remote access is typical in UNIX environments. Over the years, security has become more of an issue, thus, introducing the need to implement secure remote access. The goal is to provide assurance that the communication channels between two hosts remain private, particularly in situations where access to the remote host is through the Internet.

Secure shell (ssh) is a protocol that can be used to provide a secure channel between two hosts. It uses encryption and authentication while ensuring data integrity. Encryption, authentication, and data integrity are all required to ensure security and are discussed at length in this chapter.

Certain services under the control of the inetd daemon (such as ftp or telnet) should also be protected. TCP Wrapper is a tool that provides this protection. All requests for the protected services pass through TCP Wrapper first and are checked against access control lists. Only the requests from trusted hosts (as specified in the access control lists) are then allowed onto the protected services. TCP Wrapper can also be configured to log all connection attempts to the protected services.

5.1 Secure Shell (ssh)

SSH is a client-server application that allows secure login or secure execution of commands on a remote computer. The ssh daemon typically listens to port 22.

To ensure security, you need to:

- Ensure that users are who they say they are
- Protect the communication channel such that anyone sniffing the network will not be able to read the communication
- Ensure that the information sent is not altered in transit

SSH is a protocol that can achieve these requirements by providing strong authentication, encryption, and data integrity.

SSH is intended to replace the traditional BSD “r” commands (rlogin, rsh, rcp, and so on). These commands have various security vulnerabilities. For example, when authentication is required, passwords are not encrypted. Instead, they are sent as clear-text across the wire. However, in many cases, passwords are bypassed entirely. That is because the more common

configuration is to have these commands authenticate based on hostname or IP address (contained in the `.rhosts` or `/etc/hosts.equiv` files). Any user with access to a host listed in one of these files on the remote host may automatically be granted access to that host. Worse still is the scenario where an attacker steals a legitimate IP address from a host listed in one of these files and configures an imposter host with the stolen IP address. The imposter host then has the same access to the target host as the host whose IP address was stolen. In the case of the root user, the consequences are disastrous.

SSH provides the same functionality as the BSD “r” commands, but with added security. SSH is flexible in its setup. It supports either public-private key pairs or `.rhosts` files for authentication and has various encryption and data integrity schemes. These will be covered soon.

SSH protects against IP spoofing (where a remote host commandeers the IP address of an internal trusted host). SSH also protects against DNS spoofing where the attacker forges DNS entries. Protection is accomplished by way of the private key, public key pair. The public key of a client must be physically located on the server for the client to be able to open an SSH connection to the server. That connection is authenticated through a combination of the client's private key on the client and the client's public key on the server. A fake host claiming to have the IP address or DNS name of the client will not be able to establish the SSH connection without also having access to the client's private key. (It goes without saying that private keys need to be rigorously protected and never sent across unencrypted network connections.)

SSH also protects against password sniffing, listening X authentication data attacks, and manipulation of data by intermediary hosts. Encryption channels ensure this protection.

SSH supports two modes: Interactive, and non-interactive. In the interactive mode, SSH is used to establish a session on a remote server. In non-interactive mode, SSH is used to execute a command on a remote server and return the results to the local client.

The following Web site is an excellent source of information for SSH:

www.onsight.com/faq/ssh/ssh-faq.html

Important

If you plan to use SSH as a key component in your security environment, couple it with a tool, such as Tripwire. One very common insidious attack involves patching the SSH client executable such that it logs all account IDs and passwords used within the SSH sessions. Without a tool, such as Tripwire, in place, it is very difficult to discover that this attack has occurred.

5.1.1 Obtaining SSH

There are two versions of SSH covered in this book:

1. The original version from *SSH Communications Security*, which we will refer to as SSH.Com.
2. The version developed by the *OpenBSD* project (OpenSSH)

The first, SSH.Com, requires that you have access to a C compiler, whereas the second, OpenSSH, is available in installp format. OpenSSH and SSH.Com differ slightly in their implementations, especially in their configuration files. Compatibility between the two versions is covered in Section 5.1.8, "SSH2 interoperability between OpenSSH and SSH.Com" on page 76.

SSH.Com is available from numerous sources. The main site is <ftp://ftp.ssh.com/pub/ssh/>. A list of mirror sites can be found in <http://www.ssh.com/ssh/download.html>.

The homepage for OpenSSH is <http://www.openssh.com>. The installp version is available from <http://www-frec.bull.fr>.

There are two different protocol versions of SSH: SSH1 and SSH2. SSH.Com and OpenSSH support both SSH1 and SSH2. We cover both versions in this book. For more information on the differences between the two protocols, see Section 5.1.2, "Difference between SSH1 and SSH2" on page 62.

Commercial versions can be obtained from SSH Communication Security, F-Secure, and Van Dyke Software:

- SSH from SSH Communication is available from <http://www.ssh.com/ssh>. The commercial version includes clients for Windows, Linux, Solaris, HP-UX, and AIX. The UNIX versions include a limited server that allows up to two simultaneous connections. A full server version is also available for UNIX.

- F-Secure SSH is available at <http://www.f-secure.com/products/ssh/>. Both client and server versions of SSH are available, and SSH1 and SSH2 protocols are supported. It runs on UNIX, Windows, and Macintosh operating systems.
- SecureCRT from Van Dyke Software is available at <http://www.vandyke.com/products/securecrt/>. This SSH client runs on Windows platforms and supports the SSH1 and SSH2 protocols. Also available is SecureFX, a file transfer product that does FTP tunneling over SSH2.

5.1.2 Difference between SSH1 and SSH2

SSH1 and SSH2 are entirely different protocols. For all practical purposes, SSH2 is a complete rewrite of SSH1. The key differences with SSH2 are:

- It has better security, performance, and portability than SSH1.
- It supports DSA and other public key algorithms for encryption and authentication. (SSH1 supports RSA.)
- It has built-in support for both SOCKS and secure file transfer protocol (sftp).

The bulk of the new development effort is going into SSH2. However, at present, SSH1 has support for more platforms and more authentication schemes (Kerberos, AFS, .rhosts). It has been around much longer and, thus, has a much larger installed base. Because of this, SSH1 will continue to be important for some time.

5.1.3 Key concepts of SSH

SSH provides *authentication*, *encryption*, and *data integrity*. An understanding of these concepts is key to an understanding of how SSH fits into a total security solution.

5.1.3.1 Authentication

Authentication is the process of identifying an individual and ensuring that the person is who he or she claims to be. Authentication is important because access rights are usually granted based on user ID, thus, the need to ensure that the person logging in with a specific user ID is indeed that person.

In AIX, the most common means of authentication is through passwords. Protocols, such as telnet, ftp, and rsh, transmit these passwords in clear text across the network. This means that they are easily picked up by anyone sniffing the network. Also, in AIX, passwords have a practical length limit of eight characters (characters after the eighth are ignored).

SSH improves security by encrypting any traffic used during authentication. Passwords are not transferred in the clear. A more significant benefit is the use of public-key, private-key pairs to authenticate instead of the traditional UNIX user ID and password scheme. This form of authentication is significantly harder to break than the more conventional methods. As has already been mentioned, SSH does support other authentication methods.

To understand public-key, private-key authentication, you first need to be familiar with the types of cryptography. There are two types of cryptography algorithm: Symmetric and asymmetric.

In symmetric key cryptography, the same key is used to encrypt and decrypt the data. This is the older of the two. The advantage of this method is that it is very fast with both encryption and decryption of the data. However, the drawback is key management. How do we distribute the keys safely from sender to recipient? This is the big limitation of symmetric key cryptography that resulted in the development of asymmetric key cryptography.

In asymmetric key cryptography, keys exist in pairs: Public and private. These two keys form a pair and are generated at the same time. Although mathematically related, deriving the private key from the public key is an extremely difficult task that takes more resources than are commonly available. The longer the key length, the harder this is to do. Longer key lengths translate into better security.

To use SSH, you must have a public-key, private-key pair. A single program generates both keys. The public key must be copied to the SSH servers to which you want access. The private key remains on the local SSH client and must be kept safe. When the keys are generated, you are given the opportunity to protect the private key with a passphrase. A good passphrase is one that is 10-30 characters long, is almost impossible to guess and has a hidden meaning only for the person generating it. Notice that a passphrase can be much more secure than a standard UNIX password by virtue of its increased length.

Here is an example of how this works:

- User A wants to connect to SSH Server B via SSH.
- User A first copies their public key to SSH Server B.
- User A then requests a log in on SSH Server B via SSH.
- SSH Server B checks its set of public keys looking for the one belonging to User A (which it finds).

- SSH Server B challenges User A for the matching private key. Since the private key is protected by a passphrase, only User A can use it.
- If the challenge is successful, access is granted.

SSH uses strong cryptography, which means it is very impracticable to decode encrypted data without the right key or passphrase. SSH incorporates two different and incompatible cryptographic algorithms for authentication: RSA, and DSA. SSH1 uses RSA keys; SSH2 uses DSA keys.

Depending on the version being used and how it is configured, SSH can authenticate by the following methods:

- Password (/etc/passwd or /etc/shadow in UNIX)
- Public-key (RSA in SSH1, DSA in SSH2)
- Kerberos (SSH1 only)
- Host-based (.rhosts or /etc/hosts.equiv in SSH1, public key in SSH2)

The methods vary according to implementation. Refer to the documentation for your implementation to see if all are available. The public-key method is the one that is most commonly used, and it is the only one that we cover in this book.

5.1.3.2 Encryption

Encryption is the translation of data into a secret code that should not be decipherable to anyone other than the intended recipient. With encryption, even if someone is sniffing the network, the data will not be decipherable. Thus, a secure channel is established, and can be used even over the Internet to provide a virtual private network, reducing the need for expensive dedicated lease lines.

The ciphers or cryptographic algorithm used for encryption depend on the version of SSH that is running. SSH1 uses DES, 3DES, IDEA, or Blowfish while SSH2 uses 3DES, Blowfish, Twofish, Arcfour, or Cast128. These methods are used to provide the secure and encrypted channel to transfer data. Each of them has its own strengths and weaknesses, but they all serve the same functionality. Notice that conventional symmetric ciphers are used to encrypt communication channels.

This, however, does not mean that the encryption cannot be broken. Strong encryption just means that it is impractical to do so because it takes too long or it takes more computational power than is commonly available.

5.1.3.3 Data integrity

Data integrity helps give assurance that the received data matches the send data and has not been tampered with. SSH does not protect against data tampering as such; instead it just merely informs you of when such activity has taken place, either intentionally or by accident. In the case of SSH, no news is good news. If the data is clean, SSH is silent.

To understand how SSH makes this determination, an understanding of hashing functions is necessary. Hashing functions are mathematical in nature. A hash function takes an input of any length and calculates a fixed length output (where the length depends on the specific hashing algorithm). Any change to the input will result in a totally different output (though the length will still be the same). The sending machine calculates the hash on the input data and sends both the data and the hash to the receiving machine. The receiving machine recalculates the hash on the data and compares it to the received hash. If the data has not been tampered with, both hashes will be identical. If they are not, SSH warns you about it.

5.1.4 Installing OpenSSH on AIX

We now show you how to install the *freeware.openssh.rte* version of SSH. This version of SSH supports both SSH1 and SSH2. It is available for download in installp format from <http://www-frec.bull.fr>. It has the following prerequisite filesets: *freeware.egd.rte*, *freeware.zlib.rte*, and *freeware.openssl.rte*. The fileset versions we used during the production of this book are shown in Table 3.

Table 3. Freeware OpenSSH filesets

| Fileset | Version |
|----------------------|---------|
| freeware.openssh.rte | 2.1.1.4 |
| freeware.egd.rte | 0.8.0.0 |
| freeware.openssl.rte | 0.9.5.1 |
| freeware.zlib.rte | 1.1.3.2 |

Once you have downloaded the software distribution, ensure that it has not been tampered with. See Section 7.4.1, "Ensuring the integrity of downloads" on page 136, for more information on how to do this.

```
#ls -l
total 7152
-rwx----- 1 root system 104740 Aug 18 17:40 egd-0_8_0_0.exe
-rwx----- 1 root system 532553 Aug 18 17:40 openssh-2.1.1.4.exe
-rwx----- 1 root system 2878322 Aug 18 17:40 openssl-0.9.5.1.exe
-rwx----- 1 root system 137279 Aug 18 17:40 zlib-1_1_3_2.exe
```

You will need to install the freeware.egd.rte fileset first because it creates the local socket, /dev/entropy, which must be present for the remaining installation of SSH to work properly. Inflate egd-0_8_0_0.exe to obtain the installp-format, egd-0.8.0.0.bff file:

```
# chmod u+x ./egd-0_8_0_0.exe
# ./egd-0_8_0_0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: egd-0.8.0.0.bff
#
```

Install the fileset either through SMIT or at the command line. The following `installp` command will apply, commit, and automatically install prerequisite filesets:

```
#installp -acgX -d . freeware.egd.rte
```

```
Installation Summary
```

| Name | Level | Part | Event | Result |
|------------------|---------|------|---------|---------|
| freeware.egd.rte | 0.8.0.0 | USR | COMMITT | SUCCESS |

You now need to create and load the local socket file. This file must be in place prior to running the remainder of the OpenSSH installation so that the installation process can create all of the necessary DSA keys automatically. Without this file, that creation process will fail with a “Couldn't connect to EGD socket "/dev/entropy": Connection refused” message, and will need to be done manually afterwards.

Run the following command to create the /dev/entropy local socket file:

```
#!/usr/bin/perl -w /usr/opt/perl5/bin/egd.pl /dev/entropy
```

Note

This command was also added to /etc/rc.tcpip by the installation of the freeware.egd.rte fileset.

Inflate the other three filesets, and verify that the files are properly inflated. You need to have the openssh-2.1.1.4.bff, openssl-0.9.5.1.bff, and zlib-1.1.3.2.bff filesets. The files with extensions .asc are the PGP signatures as will be discussed in Section 7.4.1, "Ensuring the integrity of downloads" on page 136.

```
#im .toc
#ls -l
total 29192
-rwxr--r-- 1 root system 153600 Jul 06 03:39 egd-0.8.0.0.bff
-rwx----- 1 root system 104740 Aug 18 17:40 egd-0_8_0_0.exe
-rw-r--r-- 1 root system 1433600 Aug 03 07:51 openssh-2.1.1.4.bff
-rw-r--r-- 1 root system 2599 Aug 03 07:58 openssh-2.1.1.4.bff.asc
-rwx----- 1 root system 532553 Aug 18 17:40 openssh-2.1.1.4.exe
-rw-r--r-- 1 root system 9420800 Jul 28 10:39 openssl-0.9.5.1.bff
-rw-r--r-- 1 root system 2599 Aug 01 04:26 openssl-0.9.5.1.bff.asc
-rwx----- 1 root system 2878322 Aug 18 17:40 openssl-0.9.5.1.exe
-rw-r--r-- 1 root system 256000 Mar 20 12:13 zlib-1.1.3.2.bff
-rw-r--r-- 1 root system 2571 Mar 20 12:14 zlib-1.1.3.2.bff.asc
-rwx----- 1 root system 137279 Aug 18 17:40 zlib-1_1_3_2.exe
```

Use the following installp command to install the freeware.openssh.rte fileset. This will also install the freeware.openssl.rte and freeware.zlib.rte filesets automatically (because of the -g flag).

```
# installp -acgX -d . freeware.openssh.rte

Installation Summary
-----
Name                               Level      Part      Event      Result
-----
freeware.openssl.rte               0.9.5.1   USR       APPLY     SUCCESS
freeware.openssl.rte               0.9.5.1   ROOT     APPLY     SUCCESS
freeware.zlib.rte                   1.1.3.2   USR       APPLY     SUCCESS
freeware.openssh.rte               2.1.1.4   USR       APPLY     SUCCESS
freeware.openssh.rte               2.1.1.4   ROOT     APPLY     SUCCESS
```

Verify that all of the filesets are properly installed:

```
# lppchk -v
...
# lslpp -l | grep freeware.openssh.rte
...
#
```

After installation, you may want to include environment variables `MANPATH=/usr/local/man` and `PATH=/usr/local/bin` into `/etc/profile` to simplify access to the man pages and tools.

In addition to `ssh`, OpenSSH comes with the `scp` tool. It is functionally similar to `rcp`, but it has the added security of `ssh`. It can also be used as a replacement for `ftp`. Unfortunately, the current version of OpenSSH does not ship with the `sftp` tool, which is available in the SSH2 version of SSH from SSH Communications Security. This tool may be added in future versions of OpenSSH. To check, run:

```
#lslpp -f freeware.openssh.rte | grep sftp
```

5.1.5 OpenSSH using SSH1

By default, the OpenSSH installation sets the SSH daemon to SSH1, and an entry is added to `/etc/rc.tcpip` to automatically start the SSH daemon on system startup. When a user connects to the OpenSSH1 server, there are four supported methods of authentication:

1. The first is based on the `/etc/hosts.equiv` (or `/etc/shosts.equiv`) file or the `.rhosts` (or `.shosts`) file on the remote machine. This method carries the inherent security weaknesses of the traditional BSD `rsh` and `rlogin` commands (such as clear-text passwords) and should be disabled. By default, this authentication method is disabled in the `/etc/openssh/sshd_config` file (*RhostsAuthentication no*).
2. The second method is based on the first method combined with RSA-based host authentication. This means that on top of the `hosts.equiv/shosts.equiv/.rhosts/.shosts` files, the server verifies the client's host key in `/etc/ssh/known_hosts` and `$HOME/.ssh/known_hosts` files. This method also has security weaknesses since it still requires the use of `hosts.equiv/shosts.equiv/.rhosts/.shosts` files. By default, this authentication method is enabled in the `/etc/openssh/sshd_config` file (*RhostsRSAAuthentication yes*). The set up for this method is the same as that for the first. The `known_hosts` file is updated automatically by the SSH server. The SSH server prompts you whenever a new connection is added or modified.

3. The third method is based on user passwords as per normal telnet logins. The difference is that this authentication method uses encryption for the exchanges, so passwords are not sent in clear text as they are with standard protocols, such as telnet, ftp, rsh, and rlogin. This method is used when all other authentication methods fail.
4. The fourth method is based on RSA authentication. The `$HOME/.ssh/authorized_keys` file on the SSH server contains copies of the public keys for the remote clients that are allowed to log in under that account. When you generate a key-pair, the public key is stored in the `$HOME/.ssh/identity.pub` file, and the private key is stored in the `$HOME/.ssh/identity` file. You then copy the public key to the SSH server, storing it in your `$HOME/.ssh/` directory under the new name of `authorized_keys`. By default, this authentication method is enabled in the `/etc/openssh/sshd_config` file (*RSAAuthentication yes*). This is the method we recommend and the one we cover in detail in this book.

Configuring and using SSH1

By default, the SSH daemon listens on port 22. To find out which version of SSH (SSH1 or SSH2) you are running, telnet to localhost (0) on sshd port (22). Notice that the SSH version is 1.99.

```
# tn 0 22
Trying...
Connected to 0.
Escape character is '^T'.
SSH-1.99-OpenSSH_2.1.1
```

The SSH installation adds `/etc/rc.openssh` to `/etc/inittab` so that the SSH daemon will automatically start during system startup.

To manually start the SSH daemon (using `/etc/openssh/sshd_config` as the config file and `/etc/openssh/ssh_host_key` as the RSA host key file):

```
#!/usr/local/bin/opensshd -f /etc/openssh/sshd_config -h
/etc/openssh/ssh_host_key
```

To manually stop the SSH daemon:

```
#kill `cat /var/openssh/sshd.pid`
```

To refresh the daemon:

```
#kill -1 `cat /var/openssh/sshd.pid`
```

Configuration changes will take effect once the daemon has been refreshed.

To generate a public-key, private-key pair:

```
#ssh-keygen
```

By default, this will generate an SSH1 RSA key pair each with length of 1024 bits. It is also possible to generate the SSH1 RSA key pair each with length of 2048 bits by using the “-b 2048” flag.

During the creation process, you will be asked to enter a passphrase. A good passphrase is one that is 10-30 characters long, is almost impossible to guess, and has a hidden meaning only for the person generating it.

```
sp5en05 /home/khorck # ssh-keygen -b 2048
Generating RSA keys:
.....
.....ooo0.ooo0
Key generation complete.
Enter file in which to save the key (/home/khorck/.ssh/identity):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/khorck/.ssh/identity.
Your public key has been saved in /home/khorck/.ssh/identity.pub.
The key fingerprint is:
8e:f3:19:84:25:91:49:d3:fa:47:de:77:29:f0:15:5e khorck@sp5en05.
```

By default, the private key will be stored in the `$HOME/.ssh/identity` file, and the public key will be stored in the `$HOME/.ssh/identity.pub` file on the SSH client. You can specify a different name for the location of the keys, but you will also need to add an `IdentityFile <filename>` option line in the corresponding client configuration file. System-wide options are specified in the `/etc/openssh/ssh_config` file, and per-user options are specified in the `$HOME/.ssh/config` file in each user’s home directory.

Next, you need to send the public key (`$HOME/.ssh/identity.pub`) to the SSH server. (You may first want to rename the public key to identify the user and host from which the key was generated.) The public key needs to be stored (via rename or append) in the `$HOME/.ssh/authorized_keys` file on the SSH server. In this implementation, the file `authorized_keys` stores the actual keys themselves rather than just the file name(s) of where the keys are stored.

The use of `$HOME/.rhosts`, `$HOME/.shosts`, `/etc/hosts.equiv`, or `/etc/shosts.equiv` is not recommended. The use of these files will not be discussed further. For more information on their use, see the appropriate SSH man pages.

For applications, such as HACMP and PSSP, which require automatic remote login without prompting for password or passphrase, generate the keys without passphrase protection (using the `-N` flag):

```
#ssh-keygen -b 2048 -N ''
```

Use of this method requires further safeguards to ensure that the private key file is kept secure. You may want to use software, such as PGP (refer to Section 7.3.3, "Using PGP" on page 127), to protect the key by encrypting it when it is not in use. Be sure, however, to decrypt the key before you try to use it.

5.1.6 OpenSSH using SSH2

When someone attempts to connect to an OpenSSH server using SSH2, two different authentication methods are tried. The first method tried is *public key authentication*. If that method fails, the second method, *password authentication*, is tried.

To review, public key authentication uses an asymmetric scheme where each user has a *pair* of keys, one known as the *public* key and the other the *private* key. Public keys are public and can be distributed freely. Private keys are private and must be kept secure. The keys are related mathematically but it is computationally impracticable to derive the private key from the public key. Longer keys (more bits) are cryptographically stronger, and it is much harder to break them, but there is no such thing as a free lunch. You will pay a performance penalty for using longer keys. They are more computationally intensive, both to initially create them and to use them for data encryption and decryption.

SSH2 uses DSA authentication instead of the RSA authentication used by SSH1. The client uses the private key (which is a DSA key for SSH2) to sign a session identifier and sends the output to the SSH server. On receiving the data, the SSH server checks for a matching public key locally. If a matching key is found and the signature is correct, access is granted.

The client's private key is located in the `$(HOME)/.ssh/id_dsa` file on the client machine, and the client's public key is located in the `$(HOME)/.ssh/authorized_keys` file on the server machine.

If public key authentication fails, password authentication is tried. This method is similar to normal password authentication except that the passwords are encrypted rather than being transmitted in the clear. At present, this implementation of SSH does not support Kerberos or S/Key authentication.

Once authenticated, several encryption schemes are available to ensure the security of the data. The available schemes are 3DES (default), Blowfish, CAST128, or Arcfour. To further enhance security, data integrity is assured using hmac-sha1 or hmac-md5.

As part of the authentication process, SSH automatically checks to see if the host that the client is connecting to has been connected to previously. Any host that has been connected to in the past will be remembered. DSA host keys are stored in the `$HOME/.ssh/known_hosts2` file on the client. When connecting to any new host (DSA host key not found in the `known_hosts2` file), you will be informed and asked to accept or reject the connection. If you accept, the host DSA key will be automatically added into your `known_hosts2` file and will be reused in future connections.

Configuring and using SSH2

To run SSH2, both client and server need to be configured for this version of the protocol. By default, the OpenSSH client is configured to support both SSH1 and SSH2. However, by default, the OpenSSH server daemon (`sshd`) is configured only for SSH1. To reconfigure the OpenSSH server daemon (`sshd`) for SSH2, follow these steps:

1. Remove or rename the `/etc/openssh/ssh_host_key` and `/etc/openssh/ssh_host_key.pub` files.
2. Uncomment 'Protocol 2,1' in `/etc/openssh/sshd_config`
3. Restart the SSH server

By default, the SSH daemon listens on port 22. To find out which version of SSH (SSH1 or SSH2) you are running, telnet to localhost (0) on `sshd` port (22). Notice that the SSH version is now 2.0.

```
# tn 0 22
Trying...
Connected to 0.
Escape character is '^T'.
ssh-2.0-OpenSSH_2.1.1
```

The SSH installation adds `/etc/rc.openssh` to `/etc/inittab` so that the SSH daemon will automatically start during system startup.

The default command in `rc.openssh` to start the SSH daemon is:

```
#!/usr/local/bin/opensshd -f /etc/openssh/sshd_config -h
/etc/openssh/ssh_host_key
```

The `-h` flag is used to specify the RSA host key file. This will fail because we removed that file as part of the reconfiguration for SSH2. This failure triggers the SSH daemon to look for the `ssh_host_dsa_key` file instead and, if found, start the daemon with SSH2 support rather than SSH1. You can append the `-Q` flag to the end of the command to suppress the messages generated because of the missing `/etc/openssh/ssh_host_key` file.

To manually start the SSH daemon with SSH2 support (using `/etc/openssh/sshd_config` as the config file and suppress warning messages because of the missing `/etc/openssh/ssh_host_key` file):

```
#/usr/local/bin/opensshd -f /etc/openssh/sshd_config -Q
```

To manually stop the SSH daemon:

```
#kill `cat /var/openssh/sshd.pid`
```

To refresh the daemon:

```
#kill -1 `cat /var/openssh/sshd.pid`
```

Configuration changes will take effect once the daemon has been refreshed.

Because SSH2 uses DSA keys rather than RSA keys, you will need to generate a new public-key, private-key pair for SSH2:

```
#ssh-keygen -d
```

Notice the addition of the `-d` flag. This flag instructs `ssh-keygen` to generate DSA keys rather than RSA ones. By default, this will generate an SSH2 DSA key pair each with length of 1024 bits. It is also possible to generate the SSH2 DSA key pair each with length of 2048 bits by using the `“-b 2048”` flag or 3072 bits by using the `“-b 3072”` flag.

During the creation process, you will be asked to enter a passphrase. A good passphrase is one that is 10-30 characters long, is almost impossible to guess, and has a hidden meaning only for the person generating it.

```

$/usr/local/bin/ssh-keygen -d
Generating DSA parameter and key.
Enter file in which to save the key (/home/ali/.ssh/id_dsa):
Created directory '/home/ali/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ali/.ssh/id_dsa.
Your public key has been saved in /home/ali/.ssh/id_dsa.pub.
The key fingerprint is:
96:66:f5:5e:83:83:51:8e:f5:90:7c:90:a7:cc:39:3b ali@sp5en05

```

Refer to the man pages for more detail on the options.

By default, the private key will be stored in the `$HOME/.ssh/id_dsa` file, and the public key will be stored in the `$HOME/.ssh/id_dsa.pub` file on the SSH client. You can specify a different name for the location of the keys, but you will also need to add an `IdentifyFile2 <filename>` option line in the corresponding client configuration file. System-wide options are specified in the `/etc/openssh/ssh_config` file, and per-user options are specified in the `$HOME/.ssh/config` file in each user's home directory.

You next need to send the public key (`$HOME/.ssh/id_dsa.pub`) to the SSH server. (You may first want to rename the public key to identify the user and host from which the key was generated.) The public key needs to be stored (via rename or append) in the `$HOME/.ssh/authorized_keys2` file on the SSH server. In this implementation, the file `authorized_keys2` stores the actual keys themselves rather than just the file name(s) of where the keys are stored.

To ssh to the SSH server:

```

sp5en01 /home/khorck # ssh sp5en05
Enter passphrase for DSA key '/home/khorck/.ssh/id_dsa':
*****
*                                                                 *
*                                                                 *
* Welcome to AIX Version 4.3!                                     *
*                                                                 *
*                                                                 *
* Please see the README file in /usr/lpp/bos for information pertinent to *
* this release of the AIX Operating System.                       *
*                                                                 *
*                                                                 *
*****
Last login: Thu Aug 17 10:47:17 2000 on ssh from sp5en01
sp5en05 /home/khorck #

```

If public key method of authentication fails because the passphrase is wrong or the required files are not set up properly, the SSH server prompts for a password instead.

For applications, such as HACMP and PSSP, which require automatic remote login without prompting for password or passphrase, generate the keys without passphrase protection (using the `-N` flag):

```
#ssh-keygen -d -b 2048 -N ''
```

Use of this method requires further safeguards to ensure that the private key file is kept secure. You may want to use software, such as PGP (refer to Section 7.3.3, "Using PGP" on page 127), to protect the key by encrypting it when not in use. Be sure, however, to decrypt the key before you try to use it.

5.1.7 Other interesting SSH daemon configuration options

The following options can be set for the SSH daemon on the SSH server. They are applicable to both SSH1 and SSH2.

- The *StrictHostKeyChecking* option can be set to ensure that SSH will not automatically add host keys to the `known_hosts(2)` file. Connections to hosts with keys not in the `known_hosts(2)` file, or those whose keys have changed will be refused, and any changes to the `known_hosts(2)` file will need to be done manually. This protects against Trojan horse attacks. By default, this option is set to "no".
- The *AllowUsers*, *AllowGroups*, *DenyUsers*, or *DenyGroups* options can be set to allow only certain users and/or (primary) groups to connect to the SSH server. The default is to allow connections from any user and/or group.
- The *Ports* option is used to define the port on which the SSH daemon listens. By default, it is port 22, but you may want to change it to something else for security reasons.
- The *ListenAddress* option is used to control the network interfaces (IP addresses) on which SSH listens. The default is to listen on all available interfaces. Note that this option must not come before the *Port* option in the config file.
- The *PermitRootLogin* option is an option that can be used to further control root access. The arguments must be *yes*, *no*, or *without-password*. The default is "yes". For security reasons, you may not want root users to log in remotely. If so, in addition to this option, you should also disable remote access for things, such as telnet, ftp, rsh, and rlogin.

- The *SyslogFacility* option can be used to control the type of logging for sshd related messages. The values are DAEMON, USER, AUTH, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, and LOCAL7. The default is AUTH. This option is useful for debugging and audit trails.
- The *LogLevel* option controls the amount of logging done by sshd. The values are QUIET, FATAL, ERROR, INFO, VERBOSE, and DEBUG. The default is INFO. According to the man pages, DEBUG violates the privacy of users and is not recommended.
- The *X11Forwarding* option is used to control the forwarding of X11 traffic. The default is “no”. Use this option to protect X11 communications.

5.1.8 SSH2 interoperability between OpenSSH and SSH.Com

According to the OpenSSH README file, readme.openssh2, interoperability between OpenSSH and SSH from SSH Communications Security is possible. Users on a host running SSH2 from OpenSSH can connect to another host running SSH2 from SSH.Com. The steps are as follows:

1. On the OpenSSH host, run ssh-keygen as follows:

```
$ssh-keygen -f /key/from/ssh.com -X >> $HOME/.ssh/authorized
```

2. On the OpenSSH host, again run ssh-keygen, this time to generate a public key (mykey.pub in this example):

```
$ssh-keygen -f /privatekey/from/openssh -x > mykey.pub
```

3. Transfer the public key (mykey.pub) to the SSH.Com host, and update the authorization file:

```
$echo Key mykey.pub >> $HOME/.ssh2/authorization
```

5.1.9 SSH clients for the PC

An SSH client for PCs can be obtained from SSH Communications Security at:

<http://www.ssh.com>

The latest version at the time of this writing is SSHWin 2.2. It may be freely used by non-commercial users or used for a 30 day trial period by commercial users (after which time, it must be purchased). This version of the SSH client runs on Windows 95, 98, and NT, and can connect to an SSH daemon running either SSH1 or SSH2 protocol. A very interesting feature in this version of the SSH client is *secure file transfer* (sftp). The sftp tool enables file transfers to occur over an encrypted channel, thus, providing enhanced security.

Another tool worth looking at is `teraterm`. It is a terminal emulator that runs over SSH. It currently has support for SSH1 only and is available from:

<http://www.zip.com.au/~roca/ttssh.html>

5.1.10 Implications of having SSH

With SSH installed, you can securely communicate between hosts with encryption, authentication, and data integrity. Non-secure services, such as `ftp`, `telnet`, and `rsh`, can then be disabled or removed and replaced with SSH.

For added security, consider using some of the additional SSH configuration options described in Section 5.1.7, "Other interesting SSH daemon configuration options" on page 75. Specifically, the `AllowUsers`, `AllowGroups`, `DenyUsers`, `DenyGroups`, and `PermitRootLogin` options can help you control who has remote access to the SSH server, and the `ListenAddress` option lets you limit SSH traffic to specific network interfaces (IP addresses).

5.1.11 Alternatives to SSH

There are alternatives to SSH. The IP Security (IPSec) facility of AIX 4.3.x enables you to define firewall-like filter sets for network interfaces as well as create virtual private networks (VPNs) between hosts. IPSec comes standard with AIX 4.3.x, and is contained in the `bos.net.ipsec` fileset. The IPSec man pages are a good source of information. Additional detailed information on how to set up IPSec can be found in Section 7.3 of the redbook, *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962, and Chapter 9 of the redbook, *A Comprehensive Guide to Virtual Private Networks, Volume III*, SG24-5309.

5.2 TCP Wrapper

TCP Wrapper protects the `inetd` daemon by defining access controls to services provided by the daemon. When a connection comes in for a service and port combination, `inetd` first runs the TCP Wrapper program (`tcpd`). The `tcpd` program then checks the incoming connection request against the access controls to ensure that it is a legal request for the requested service and port combination. If it is, `tcpd` runs the requested server program to satisfy the original request. TCP Wrapper logs all connection requests to maintain an audit trail of both successful and unsuccessful connection attempts.

From the standpoint of `inetd`, the entire `tcpd` process is transparent. The only change to `inetd` that is required is to the `/etc/inetd.conf` file. It needs to be

modified to start `tcpd` instead of the requested server program. (The `tcpd` program will take care of starting the requested server program.) TCP Wrapper can be used to protect `telnet`, `finger`, `ftp`, `exec`, `rsh`, `rlogin`, `tftp`, `talk`, `comsat`, and other services that have a one-to-one mapping with an executable server program.

This tool is really useful on networks where you want to specify which hosts are allowed to connect to a particular service on a particular port and which hosts are not. For example, you may want to allow only certain hosts to connect to your FTP server while disallowing all others.

5.2.1 Obtaining and installing TCP Wrapper

You can obtain TCP Wrapper (and other tools, such as `SATAN` and `POSTFIX`, from the same author) from:

```
ftp://ftp.porcupine.org/pub/security/index.html
```

TCP Wrapper is also available from the Bull site in `installp` format. The version we installed and tested for this book is Version 7.6. Once downloaded, ensure the distribution has not been tampered with. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136, for more information on how to do this.

Inflate the distribution and ensure the filesets are all there and have the correct sizes:

```
# ./tcp_wrappers-7.6.1.0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: tcp_wrappers-7.6.1.0.bff
  inflating: tcp_wrappers-7.6.1.0.bff.asc
# ls -l
total 1032
-rw-r--r--  1 root system   358400 Jul 06 11:27 tcp_wrappers-7.6.1.0.bff
-rw-r--r--  1 root system    2643 Jul 06 11:28 tcp_wrappers-7.6.1.0.bff.asc
-rwx-----  1 root system   162727 Aug 18 10:49 tcp_wrappers-7.6.1.0.exe
```

Install the product, and verify that the installation completed successfully:

```
# installp -acgX -d . freeware.tcp_wrappers.rte

Installation Summary
-----
Name                      Level      Part      Event      Result
-----
freeware.tcp_wrappers.rte 7.6.1.0    USR       APPLY      SUCCESS

# lsllp -l freeware.tcp_wrappers.rte
Fileset                    Level     State      Description
-----
Path: /usr/lib/objrepos
freeware.tcp_wrappers.rte 7.6.1.0  COMMITTED  TCP/IP daemon security wrapper
package (with IPv6)
```

5.2.2 Configuring TCP Wrapper

Once installed, you need to decide which inetd-controlled services to protect with TCP Wrapper. For the appropriate service, replace the corresponding server program in `/etc/inetd.conf` with `tcpd`. For example, if you want to protect the `ftp` service, modify the `ftp` line as follows:

```
ftp      stream  tcp6    nowait  root    /usr/local/bin/tcpd      ftpd
```

Repeat for each service you want to protect with TCP Wrapper. When you have finished making modifications to `/etc/inetd.conf`, refresh `inetd`:

```
# refresh -s inetd
```

Next, configure `tcpd` to log appropriately. By default, TCP Wrapper uses the mail facility within `syslog` to log to `/var/adm/syslog`. We recommend that you modify the `/etc/syslog.conf` file to log to a different location. For example, to have it log to `/var/adm/tcp_wrapper.log`, modify `/etc/syslog.conf` as follows:

```
mail.debug                /var/adm/tcp_wrapper.log
```

You need to create the `/var/adm/tcp_wrapper.log` file and refresh the `syslog` daemon. Permissions of 600 for the log file will prevent normal users from reading the logs:

```
# touch /var/adm/tcp_wrapper.log
# chmod 600 /var/adm/tcp_wrapper.log
# refresh -s syslogd
```

You now need to configure the TCP Wrapper access control lists to define which hosts are allowed to execute the protected services and which are not. Access control is done through the `/etc/hosts.allow` and `/etc/hosts.deny` files. The search order is `/etc/hosts.allow` first, then `/etc/hosts.deny`. If neither

contains the match, access is granted. For additional access control options, see the man pages `hosts_options` and `hosts_access.5`.

Here is how we configured TCP Wrapper for our lab environment. Our server is `merlin.itso.ibm.com`. We set up TCP Wrapper to protect the telnet and ftp services. For telnet, all servers in the `itso.ibm.com` domain, with the exception of `sp5cw0`, are allowed to use the service. For ftp, `arthur` is the only server in the `itso.ibm.com` domain allowed to use the service. Access to either service from outside the `itso.ibm.com` domain is denied.

To enable this functionality, we first had to modify the telnet and ftp lines in `/etc/inetd.conf` to run `/usr/local/bin/tcpd` instead of `telnetd` and `ftpd`. We then created the `/etc/hosts.allow` and `/etc/hosts.deny` files with the required access control for these services. (Note that you can use IP addresses, host names, domain names, or a combination of all of them in these files. When specifying domain names, be sure to include the leading period in the name, as in `.itso.ibm.com`, and when specifying networks, be sure to include the netmask, as in `9.12.0.0/255.255.255.0`). Lastly, we verified the access controls with the `tcpdchk -v` command:

```

# cat /etc/inetd.conf

...
ftp      stream  tcp6     nowait  root    /usr/local/bin/tcpd    ftpd
telnet   stream  tcp6     nowait  root    /usr/local/bin/tcpd    telnetd -a
...

# cat /etc/hosts.allow
telnetd: .itso.ibm.com except sp5cw0.itso.ibm.com : allow
ftpd: arthur.itso.ibm.com: allow

# cat /etc/hosts.deny
ftpd: 9.12.0.0/255.255.255.0 : deny
ALL: ALL: deny

# tcpdchk -v
Using network configuration file: /etc/inetd.conf

>>> Rule /etc/hosts.allow line 1:
daemons: telnetd
clients: .itso.ibm.com except sp5cw0.itso.ibm.com
command: allow
access: granted

>>> Rule /etc/hosts.allow line 2:
daemons: ftpd
clients: arthur.itso.ibm.com
command: allow
access: granted

>>> Rule /etc/hosts.deny line 1:
daemons: ftpd
clients: 9.12.0.0/255.255.255.0
command: deny
access: denied

>>> Rule /etc/hosts.deny line 2:
daemons: ALL
clients: ALL
command: deny
access: denied

```

To test this setup, we used the `tcpdmatch` command. We first simulated a telnet test from `sp5cw0.itso.ibm.com`, followed by a telnet test from `arthur.itso.ibm.com`. As expected, access was denied on the first test but granted on the second:

```
# tcpdmatch telnetd sp5cw0.itso.ibm.com
client:  hostname sp5cw0.itso.ibm.com
client:  address  9.12.0.5
server:  process  telnetd
access:  denied

# tcpdmatch telnetd arthur.itso.ibm.com
client:  hostname arthur.itso.ibm.com
client:  address  9.12.0.18
server:  process  telnetd
access:  granted
```

5.2.3 Additional TCP Wrapper security features

By default, TCP Wrapper verifies that a client is indeed who it claims to be. TCP Wrapper (tcpd) does this by performing a name to IP address lookup and an IP address to name lookup to ensure that both are consistent. This double lookup helps ensure the authenticity of the client.

Chapter 6. Port and network scanning

Network and port scanners are tools typically used by attackers to find vulnerabilities in systems. They are used to map out your network and look for potential security weaknesses. Port scanners scan systems looking for open ports and known vulnerabilities. For example, if there is a new vulnerability for named (BIND), attackers may scan to see if that port is active, and, if so, attempt to exploit the new vulnerability.

Learning these tools gives you two advantages:

- You learn how these tools can be used against your network and what information can be gathered. You can then build defense mechanisms around this information.
- You use this knowledge to proactively scan your own network to find possible loopholes before an attacker discovers it.

Important

Even though this is considered ethical hacking, many organizations have very strict policies against it. Be sure to get the proper permissions in writing before using these tools on a production network.

In this chapter, we discuss *fping* and *nmap* which are network and port scanners. The *fping* tool is a useful for quickly mapping out the network by sending ICMP echo requests to the entire network or to an IP range. This technique is also called the *ping sweep*. With this tool, you can quickly scan the network and determine which hosts are alive, ensure that no new hosts have been introduced without your knowledge, and look for potential weak points.

The *nmap* tool is a premier port scanner. A port scanner connects to TCP or UDP ports on a machine to determine which ports are active and in LISTEN state (as can be seen with the `netstat -an` command). The *nmap* tool uses various scan techniques to determine if a port is active. The default scan is a TCP connect() scan that uses the standard TCP handshake (explained in Section 3.3.6.1, "SYNDefender" on page 37). The tool also includes many other types of scans, some of which are more subtle (like stealth scans) to prevent detection.

In this chapter, we also discuss other types of tools, such as *SAINT*, *PortSentry*, and *Isof*. These tools are not port or network scanners per se but

are included here because they are complimentary tools to port and network scanners, and can significantly help to secure your environment.

SAINT (Security Administrator's Integrated Network Tool) is a tool similar to the NSA tool discussed in Section 4.4, "Network Security Auditor (NSA)" on page 49. SAINT is an enhanced and updated version of SATAN. SAINT checks remote machines for a list of known vulnerabilities. To run SAINT, you first provide it with a list of target machines to scan. SAINT then checks the target machines for the vulnerabilities and reports its findings in an easy-to-read format. It also gives you recommendations on how to solve the vulnerabilities that it found.

It is also a good idea to have a tool to defend against port scans. PortSentry is just such a tool. It alerts you when it detects a port scan (both UDP and TCP), and can be configured to drop those connections. This tool is very useful as an early warning mechanism for suspected attacks, and, when configured to drop the connections, can help to thwart the attacker(s).

The last tool covered in this chapter is Isof. It is a good tool to help you identify ports which are unknown to you. When you see that your server has an active port, and you are unsure which application owns it, Isof can help you determine the owner. Note that the Isof tool is not really a security tool. You can think of it more as a very enhanced version of the `netstat -an` command.

6.1 fping

The fping tool is a ping sweep utility that can help map the network topology by sending ICMP echo requests to the entire network or to a range of IP addresses. The fping tool is similar to the ping tool except that it sends out ICMP echo request packets in a round-robin fashion. When executed, fping takes in a list of IP addresses to be scanned as arguments, and sends out the ICMP packets to each address without waiting for a response. This feature enables fping to be a quick tool for scanning an entire network.

The IP addresses passed to fping can be in several forms:

- They can be specified in a list on the command line.
- They can be specified in a list in a file.
- They can be specified in ranges either on the command line or in the file.

These options are covered in detail in Section 6.1.2, "Using fping" on page 86.

With `fping`, you can quickly determine which hosts are alive on the network and ensure that no new hosts have been introduced on the network without your knowledge. Keep in mind, however, that there may be machines on the network that have been configured specifically not to respond to ping requests. For example, firewalls are typically configured this way. In other cases, you may get a response back indicating that a machine is down. This may be a false positive caused by a slow network or an overloaded machine.

The `fping` tool is also very useful for verifying that the network is configured the way you think it is configured. Over time, things change, and documentation rapidly goes out of date. You can use `fping` routinely as an audit tool against your network documentation. This is especially important for early identification of potential loopholes in your security. For example, suppose a new machine with multiple network cards is introduced on the network without your knowledge. The danger here is that this machine, by virtue of its multiple network cards, could easily provide an alternate path into your network, a path that enables an attacker to circumvent a firewall.

Another good use of `fping` is to determine if broadcast addresses are configured to respond to pings. Denial of service attacks have been successfully launched against the broadcast addresses (both `.0` and `.255`). For this reason, it is a good idea to ensure that your broadcast addresses are not configured to respond to ping requests.

More information about denial of service attacks and how to prevent them can be found at:

www.pentics.net/denial-of-service/white-papers/smurf.cgi

6.1.1 Obtaining and installing `fping`

The `fping` tool is available for download from:

<http://www-frec.bull.fr> (install format)

<http://www.mirror.ac.uk/sites/ftp.kernel.org/pub/software/admin/mon/>

<http://www.svn.net/datamonk/outline.shtml>

For this redbook, we used `fping` version 2.2 from the Bull site.

Once downloaded, ensure that the distribution has not been tampered with. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136, for more information on how to do this.

Then, inflate the distribution:

```
# ./fping-2.2.1.0.exe
UnZipSFX 5.31 of 31 May 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
inflating: fping-2.2.1.0.bff
```

Verify that it was inflated properly:

```
# ls -l
total 384
-rw-r--r--  1 root    system   102400 Jan 14 1998  fping-2.2.1.0.bff
-rwx-----  1 root    system    92765 Aug 18 10:48  fping-2.2.1.0.exe
```

Now, install the fileset `freeware.fping.rte`, and verify that the installation completed successfully:

```
# installp -acqX -d . freeware.fping.rte

Installation Summary
-----
Name                               Level      Part      Event      Result
-----
freeware.fping.rte                 2.2.1.0   USR       APPLY      SUCCESS
```

6.1.2 Using fping

The `fping` tool scans a large number of hosts quickly. The list of hosts to scan can be specified on the command line or through scripts. Scripts to automate the generation of the host list are available for download on the Web. Also, a sample script that will scan an entire Class C network (or any network with netmask 255.255.255.0) is given in Appendix B, “Script used to scan a network with `fping`” on page 207. Three examples of using `fping` are given in this section.

The first example shows how to run `fping` with the host list provided on the command line:

```
# /usr/local/bin/fping 192.168.5.1 192.168.5.3 192.168.5.5
192.168.5.1 is alive
192.168.5.5 is alive
192.168.5.3 is unreachable#
```

In this example, `fping` is used to scan three address: 192.168.5.1, 192.168.5.3, and 192.168.5.5. The three addresses are used by `fping` as its

host list. It sends ICMP echo requests to each in round-robin fashion. Notice that the address, 192.168.5.3, did not respond after a timeout.

By default, `fping` uses:

- A retry count of three, excluding the initial ping (controlled by the `-r` flag)
- A 25ms minimum time between sending packets (controlled by the `-i` flag)
- A factor of 1.5 for increased retry wait times (controlled by the `-B` flag)

The second example shows how to run `fping` with the `/etc/hosts` file as input:

```
# cat /etc/hosts | fping
127.0.0.1 is alive
192.168.5.150 is alive
9.12.0.5 is alive
192.168.5.1 is alive
192.168.5.5 is alive
192.168.5.9 is alive
192.168.5.13 is alive
192.168.15.5 is alive
192.168.15.9 is alive
192.168.15.13 is alive
9.12.2.168 is unreachable
```

You can use this method as a quick way to ensure that all hosts in the hosts table are alive.

The third example shows how to run `fping` with the script found in Appendix B, “Script used to scan a network with `fping`” on page 207. In this example, we are scanning the 192.168.5.0 network (addresses 192.168.5.0 through 192.168.5.255, inclusive). Since 192.168.5.0 and 192.168.5.255 can act as broadcast addresses, we also want them scanned to ensure that they do not respond to ping requests because of the vulnerability to denial of service attacks.

```
# /tmp/fping.script 192.168.5.0
Please enter filename to output results
/tmp/fping.out
Output file is /tmp/fping.out

Obtaining list of hosts to scan and storing the list into /tmp/fping.data

Scanning hosts now. Please wait for prompt to return.
View output file /tmp/fping.out for results
```

The entire scan took less than a minute to run (including generating the list of hosts to be scanned). The output file contains the following results:

```
# cat /tmp/fping.out
192.168.5.0 : duplicate for [0], 84 bytes, 2.40 ms [<- 192.168.5.5]
192.168.5.255 : duplicate for [0], 84 bytes, 1.71 ms [<- 192.168.5.5]
192.168.5.0 is alive [<- 192.168.5.13]
192.168.5.1 is alive
192.168.5.5 is alive
192.168.5.9 is alive
192.168.5.13 is alive
192.168.5.150 is alive
192.168.5.255 is alive [<- 192.168.5.13]
192.168.5.2 is unreachable
192.168.5.3 is unreachable
192.168.5.4 is unreachable
...
192.168.5.253 is unreachable
192.168.5.254 is unreachable

    256 targets
      7 alive
    249 unreachable
      0 unknown addresses

    996 timeouts (waiting for response)
    1003 ICMP Echos sent
      9 ICMP Echo Replies received
      0 other ICMP received

    0.41 ms (min round trip time)
    1.41 ms (avg round trip time)
    2.40 ms (max round trip time)
    35.173 sec (elapsed real time)
```

As you can see, the broadcast addresses, 192.168.5.0 and 192.168.5.255, had responses from 192.168.5.5 and 192.167.5.13. A check of these two servers reveals that they are configured with the `bcastping no` option set to on (1). Refer to Section 8.4.2, "Broadcast protection" on page 187 for more information about this `no` option.

6.1.3 Protection against ping sweeps

In general, you can use intrusion detection software to protect your network from ping sweeps. This software implements a form of network sniffer that monitors the network traffic looking for this type of event. Keep in mind, however, that some port scanners (like `nmap`) can do a really slow scan (`-T Paranoid` flag) that may be missed by the intrusion detection software.

You can also protect your network by disabling ICMP requests from the Internet on your routers. There is also software available for download on the

Web that provides protection against port scanning. However, you should thoroughly test these packages prior to putting them in production to determine how sensitive they are to scans. Remember that some port scanners can be very subtle (in stealth mode) and use a number of different techniques to prevent detection.

To minimize your exposure, you should harden your operating system (refer to Chapter 8, “Securing AIX” on page 139) by removing unnecessary services. This reduces the number of potential targets that attackers have.

As always, your best defense is knowing how to think like an attacker. You should stay current with the latest techniques by reading articles posted on security sites and subscribing to security-related mailing lists. Learn how to view your network from both the defensive and offensive positions.

6.2 Network Mapper (NMAP)

The nmap tool was written by Fyodor and is well accepted as a premier tool for port scanning. It is very flexible, supports many scanning techniques, and enables you to do port scanning of hosts or entire networks. A port scanner connects to the TCP or UDP ports on a machine to determine which ports are active and in LISTEN state (as can be seen in the output of the `netstat -an` command).

To check for vulnerabilities on a system, run an nmap port scan on the system to see which services are running and exploitable. The nmap tool supports various scan techniques to determine if a port is active. The default scan is a TCP connect() scan which uses the standard TCP handshake (explained in Section 3.3.6.1, “SYNDefender” on page 37). If logging is enabled to track port scanning, this type of scanning will surely be logged. It is also visible in the output of the `netstat` command.

The nmap tool supports more subtle scans like SYN stealth (half open) scanning that may escape some logging facilities. Other techniques like TCP FIN, Xmas, or NULL (stealth) scanning, TCP ftp proxy (bounce attack) scanning, and UDP port scanning can also be used.

Other nmap features include ping scanning to determine the hosts that are running, remote OS identification to “guess” the remote operating system, and reverse ident scanning to determine the owner of the remote process if ident protocol is running.

Included in the package is the `/usr/local/lib/nmap-2.53/nmap.deprecated.txt` file that explains the various scan techniques and their use. Read that file for

a better understanding on what nmap can do. Alternatively, you can visit <http://www.insecure.org/nmap> for a more detailed description of the techniques and features supported by nmap.

6.2.1 Obtaining and installing nmap

The nmap tool in install format can be downloaded from <http://www-frec.bull.fr/>. For this redbook, we used nmap version 2.53 from the Bull site. The nmap tool can also be obtained from its homepage at <http://www.insecure.org/nmap>.

For nmap on AIX, you also need the `/usr/bin/lex` and `/usr/bin/yacc` utilities. You can find these in the `bos.adt.utils` fileset. If `lex` and `yacc` are not already installed, you will need to install them before installing nmap. To determine if the `bos.adt.utils` fileset is installed:

```
$ lspp -l bos.adt.utils
Fileset                Level  State   Description
-----
Path: /usr/lib/objrepos
bos.adt.utils          4.3.3.0 COMMITTED Base Application Development
                           Utilities - lex and yacc
```

Once you have downloaded nmap, make sure the distribution has not been tampered with. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136 for more information on how to do this.

Next, unpack the distribution, and install the nmap tool. (Notice that nmap requires `libpcap` as a prerequisite):

```
# ./nmap-2_53_0_0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: nmap-2.53.0.0.bff
  inflating: nmap-2.53.0.0.bff.asc

# ./libpcap-0_5_0_0.exe
UnZipSFX 5.41 of 16 April 2000, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: libpcap-0.5.0.0.bff
  inflating: libpcap-0.5.0.0.bff.asc

# installp -acgX -d . freeware.nmap.rte

Installation Summary
-----
Name                               Level      Part      Event     Result
-----
freeware.libpcap.rte               0.5.0.0   USR       APPLY     SUCCESS
freeware.nmap.rte                  2.53.0.0  USR       APPLY     SUCCESS
```

Once installed, enable the pse drivers. If you do not enable the pse drivers, you will get messages like this one:

```
# nmap -sS -O -P0 -v merlin > /tmp/merlin.nmap.sS
pcap_open_live: /dev/dlpi/tr0: No such file or directory
There are several possible reasons for this, depending on your operating system:
LINUX: If you are getting Socket type not supported, try modprobe af_packet or recompiling
your kernel with SOCK_PACKET enabled.
*BSD: If you are getting device not configured, you need to recompile your kernel with
Berkeley Packet Filter support. If you are getting No such file or directory, try creating
the device (eg cd /dev; MAKEDEV <device>; or use mknod).
SOLARIS: If you are trying to scan localhost and getting '/dev/lo0: No such file or
directory', complain to Sun. I don't think Solaris can support advanced localhost scanning.
You can probably use "-P0 -sT localhost" though.
```

To enable the drivers, edit the */etc/pse.conf* file, uncomment the appropriate entry for your type of network adapter (such as Ethernet or token ring), and add an entry for the loopback adapter:

```
#d+ dlpi en /dev/dlpi/en # streams dlpi ethernet driver
#d+ dlpi et /dev/dlpi/et # streams dlpi 802.3 driver
d+ dlpi lo /dev/dlpi/lo # streams dlpi loopback driver
d+ dlpi tr /dev/dlpi/tr # streams dlpi token ring driver
#d+ dlpi fi /dev/dlpi/fi # streams dlpi FDDI driver
```

Note

There seems to be some nmap functionality in the release that we used for this redbook that does not work properly with the token ring interface.

Now, activate the drivers:

```

# strload -f /etc/pse.conf
strload: cannot initialize stddev: Do not specify an existing file.
strload: cannot initialize stddev: Do not specify an existing file.
strload: 'spX' already loaded
strload: 'sc' already loaded
strload: 'stdmod' already loaded
strload: cannot initialize dlpi: Do not specify an existing file.
strload: cannot initialize dlpi: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: cannot initialize xtiso: Do not specify an existing file.
strload: 'timod' already loaded
strload: 'tirdwr' already loaded
# ls -l /dev/dlpi
total 0
crw-rw-rw-  1 root      system   11, 42 Sep 03 22:13 en
crw-rw-rw-  1 root      system   11, 43 Sep 03 22:14 lo

```

You are now ready to use nmap. Check out the `/usr/local/lib/nmap-2.53/nmap.deprecated.txt` file for highlights of the many different scanning techniques available and suggestions on when you might want to use them as well as the different types of information they provide.

By default, the permission of nmap is 755. In all likelihood, you probably do not want normal users scanning your network. We recommend that you change the permission to 500.

6.2.2 Nmap usage

The nmap tool reports one of three states for each port scanned. The states are:

- *Open* - The target machine will accept() connections on that port.
- *Filtered* - A device, such as a firewall, is protecting the port and preventing connection.
- *Unfiltered* - The port is unprotected.

Important

Some organizations have strict policies against network scanning by unauthorized personnel. Be sure to get the proper approval in writing before running tools, such as nmap, on a production network.

The following screen contains some examples of nmap usage along with the corresponding output.


```

# nmap -P0 -p23 -sT 172.16.30.0/24
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Strange error from connect (13):Permission denied
The 1 scanned port on (172.16.30.0) is: closed
Interesting ports on (172.16.30.1):
Port      State      Service
23/tcp    filtered  telnet
...
...
# nmap -P0 -p23,514,513 172.16.30.50
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on merlinen0 (172.16.30.50):
(The 2 ports scanned but not shown below are in state: closed)
Port      State      Service
23/tcp    open       telnet

# nmap -P0 -v 172.16.30.50
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
No tcp,udp, or ICMP scantype specified, assuming vanilla tcp connect() scan. Use -sP
if you really don't want to portscan(and just want to see what hosts are up).
Initiating TCP connect() scan against merlinen0 (172.16.30.50)
Adding TCP port 23 (state open).
Adding TCP port 111 (state open).
Adding TCP port 21 (state open).
Adding TCP port 53 (state open).
Adding TCP port 684 (state open).
Adding TCP port 683 (state open).
Adding TCP port 6000 (state open).
The TCP connect scan took 0 seconds to scan 1523 ports.
Interesting ports on merlinen0 (172.16.30.50):
(The 1516 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
23/tcp    open       telnet
53/tcp    open       domain
111/tcp   open       sunrpc
683/tcp   open       unknown
684/tcp   open       unknown
6000/tcp  open       XI1
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds

# nmap -sS -P0 -O sp5en01
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on sp5en01 (192.168.5.1):
(The 1502 ports scanned but not shown below are in state: closed)
Port      State      Service
7/tcp     open       echo
9/tcp     open       discard
13/tcp    open       daytime
19/tcp    open       chargen
21/tcp    open       ftp
2770/tcp  open       sometimes-rpc3
32771/tcp open       sometimes-rpc5
TCP Sequence Prediction: Class=truly random
                          Difficulty=9999999 (Good luck!)
Remote operating system guess: IBM AIX v3.2.5 - 4
Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds

```

Read the man pages and review the `/usr/local/lib/nmap-2.53/nmap.deprecated.txt` file to get a good overview of the nmap tool and its capabilities. There are numerous scan techniques that can be used under many different circumstances in different environments. The best way to get familiar with nmap is to play around with it in a lab environment. As you do, remember to also view what you see from the eyes of a potential attacker. We think you will be amazed at what you see.

6.2.3 Protection against port scanners

In general, you can use intrusion detection software to protect your network from port scanners. This software implements a form of network sniffer that monitors the network traffic looking for this type of event. Keep in mind, however, that some port scanners, such as nmap, can do a really slow scan (-T Paranoid flag) that may be missed by the intrusion detection software.

There is software available for download on the Web that provides protection against port scanning. However, you should thoroughly test these packages prior to putting them in production to determine how sensitive they are to scans. Remember that some port scanners can be very subtle (in stealth mode) and use a number of different techniques to prevent detection.

To minimize your exposure, you should harden your operating system (refer to Chapter 8, "Securing AIX" on page 139) by removing unnecessary services. This reduces the number of potential targets that attackers have.

As always, your best defense is knowing how to think like an attacker. You should stay current with the latest techniques by reading articles posted on security sites and subscribing to security-related mailing lists. Learn how to view your network from both the defensive and offensive positions.

6.3 Security Administrator's Integrated Network Tool (SAINT)

SAINT is a useful scanning tool for system vulnerabilities. It is an enhanced and updated version of SATAN, and is similar to the NSA tool discussed in Section 4.4, "Network Security Auditor (NSA)" on page 49. Even if you have other similar tools, it is always a good idea to have multiple tools so that you can cross-verify the results from each.

SAINT has many built-in tests for known vulnerabilities. See <http://www.wwdsi.com/cgi-bin/vulns.pl> for the full list of vulnerabilities known to SAINT. Note, however, that the list of vulnerabilities changes with each release of the tool. That is why it is important to make sure you are running

the latest version of the software (similar to what you are probably already doing with virus-protection software).

SAINT can be run from the command line or from a browser. The browser interface is nice in that it enables you to run the tool and view the results all within the browser. The online documentation is plentiful and provides explanations of how to use the tool. It even gives recommended procedures for solving the vulnerabilities it finds.

The SAINT homepage is at:

<http://www.wwdsi.com/saint/index.html>

This site provides a lot of detailed information about the tool. It also has a very cool demo/tutorial of the product.

6.3.1 Obtaining and installing SAINT

SAINT is only available in source-code form. Download it from <http://www.wwdsi.com/saint/index.html>. You will then need to compile it with a C compiler. At the time of this writing, the current version is SAINT 2.2.

Important

For security reasons, it is not a good idea to have a C compiler loaded on a production system. Compile the program on a non-production system, and only move the executable program to the production system. Be sure also to set the permissions as tightly as possible.

To install the package, you will need to the gzip utility (available for download from the Bull site), a copy of PERL, and a C compiler. We used the version 4.4 C compiler for AIX under AIX 4.3.3.

Download the package, and install it as shown in the following screen.

```

# ls -l
total 1352
-rw-r----- 1 root    system   687209 Sep 11 14:57 saint-2.2.tar.gz
# gzip -d saint-2.2.tar.gz
# tar -xvf saint-2.2.tar
...
...
# perl reconfig
Reconfiguring...Checking to make sure all the targets are here...
Trying to find Perl...

Perl is in /bin/perl
...
...
Changing paths in config/paths.sh...

# make
Usage: make system-type. Known types are:
aix osf freebsd openbsd bsdi dgux irix4 irix
hpux9 hpux10 hpux11 linux-old linux-new
solaris sunos4 sunos5 sysv4 tru64 unixware7-udk
make: 1254-004 The error code from the last command is 1.

Stop.
# make CC=cc aix
      cd src/misc; make "LIBS=" "XFLAGS=-DAUTH_GID_T=in
...
...
cc -o ../../bin/ddos_scan dds.o

```

Additional installation information is available in the README file that comes with the product.

To run SAINT, you can either use the command line (`saint -h`) or a browser. If you are going to use the browser interface, you need to ensure that your X11 environment is set up correctly to display the browser. The default browser was set automatically when you ran `perl reconfig` during the installation process.

Invoke the SAINT browser interface, shown in Figure 12 on page 97, by typing `saint`.

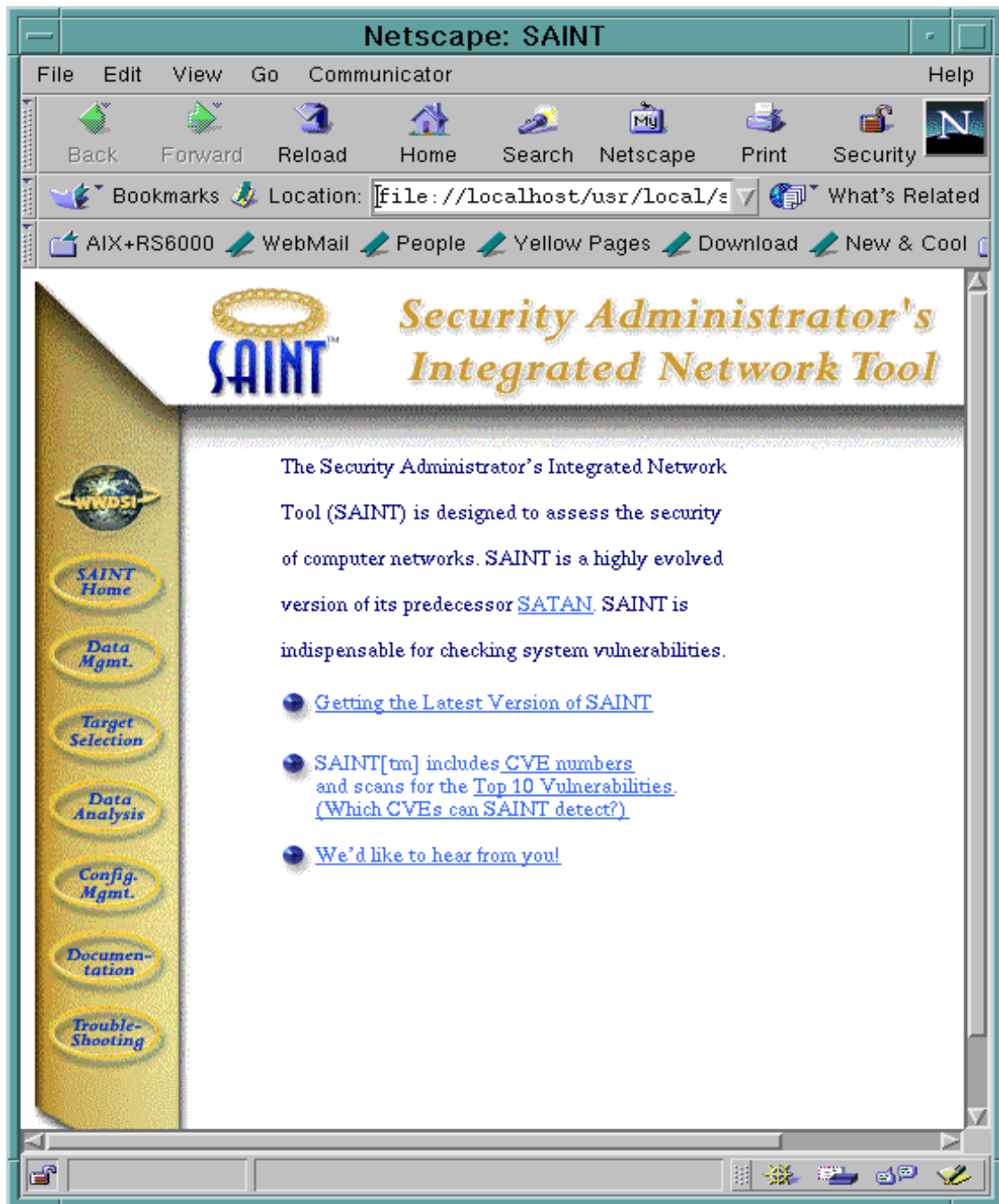


Figure 12. SAINT main page

6.3.2 Using SAINT

Important

Some companies have strict restrictions against doing such scanning. Ensure that you have the necessary authorization prior to running these scans. Also, you may want to play safe by ensuring that backups are done prior to scanning since crashes may occur under some severe conditions.

The steps for configuring SAINT, running a scan, and viewing the results are as follows:

1. Click the **Data Mgmt.** and **Config Mgmt.** buttons to configure SAINT for your requirements.
2. Click the **Target Selection** button to select the target systems to scan.
3. Select the scan method, then click the **Start the scan** button.
4. Click the *Data Analysis* button, then click the **By Approximate Danger Level** button to analyze the results. Select each item in the list for further details.
5. Click the **Documentation** button for access to the full product documentation for SAINT.

6.4 PortSentry

PortSentry from Psionic is a port scan detector. Since port scanning is a common part of an attacker's routine, having a defense against it will help secure your environment. When PortSentry detects a port scan, it can react by adding the IP address of the attacking host into the TCP Wrapper, `/etc/hosts.deny` file so that further connection attempts to protected services are denied. It can also kill off the route from the attacking machine. When used in conjunction with the LogCheck utility (also from Psionic), PortSentry can notify you through email when it detects a possible attack. (Without this added feature, you will need to regularly check the PortSentry logs yourself.)

The PortSentry homepage is can be found at the following address:

<http://www.psionic.com/abacus/>

Important

Be sure to thoroughly test your PortSentry rules and thresholds before putting them into production. It is amazingly easy to lock out legitimate clients. Also, once you have implemented PortSentry, continue to monitor it to ensure that it is really doing what you expected it to do.

6.4.1 Obtaining and installing PortSentry

PortSentry is only available in source code form. Download it from <http://www.psonic.com/abacus/port Sentry/>. Ensure the integrity of your download by verifying your signature posted on the site. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136.

To install the package, you need the gzip utility (available for download from the Bull site) and a C compiler. We used the version 4.4 C compiler for AIX under AIX 4.3.3.

Download the package, and unpack it as follows.

```
[/usr/local] # ls -l
total 408
drwxr-xr-x  2 bin      bin          512 Sep 12 14:52 bin
drwxr-xr-x  3 root    system      512 Aug 30 19:04 include
drwxr-xr-x 13 root    system      512 Sep 12 14:52 lib
drwxr-xr-x  6 root    system      512 Aug 30 19:04 man
-rw-r----- 1 root    system     184320 Sep 12 15:36 portsentry-1.0.tar.Z
drwxr-xr-x  4 root    system      512 Sep 12 14:52 share
drwxr-xr-x  6 root    system      3584 Sep 12 16:00 src
[/usr/local] # gzip -d portsentry-1.0.tar.gz
[/usr/local] # tar -xvf portsentry-1.0.tar
x portsentry-1.0
x portsentry-1.0/CHANGES, 6729 bytes, 14 media blocks.
x portsentry-1.0/CREDITS, 6133 bytes, 12 media blocks.
x portsentry-1.0/LICENSE, 2215 bytes, 5 media blocks.
x portsentry-1.0/Makefile, 5519 bytes, 11 media blocks.
x portsentry-1.0/README.COMPAT, 334 bytes, 1 media blocks.
x portsentry-1.0/README.install, 20191 bytes, 40 media blocks.
x portsentry-1.0/README.methods, 5475 bytes, 11 media blocks.
x portsentry-1.0/README.stealth, 7728 bytes, 16 media blocks.
x portsentry-1.0/ignore.csh, 2715 bytes, 6 media blocks.
x portsentry-1.0/portsentry.c, 52759 bytes, 104 media blocks.
x portsentry-1.0/portsentry.conf, 10403 bytes, 21 media blocks.
x portsentry-1.0/portsentry.h, 3896 bytes, 8 media blocks.
x portsentry-1.0/portsentry.ignore, 236 bytes, 1 media blocks.
x portsentry-1.0/portsentry_config.h, 2438 bytes, 5 media blocks.
x portsentry-1.0/portsentry_io.c, 19871 bytes, 39 media blocks.
x portsentry-1.0/portsentry_io.h, 2912 bytes, 6 media blocks.
x portsentry-1.0/portsentry_tcpip.h, 4688 bytes, 10 media blocks.
x portsentry-1.0/portsentry_util.c, 4160 bytes, 9 media blocks.
x portsentry-1.0/portsentry_util.h, 2325 bytes, 5 media blocks.
```

Additional installation information is available in the README.install file that comes with the product.

To enhance the protection provided by PortSentry, you should have TCP Wrapper installed and running. For more information on how to install and configure TCP Wrapper, see Section 5.2, "TCP Wrapper" on page 77.

The steps to install and configure PortSentry are as follows:

1. Edit the portsentry_config.h header file, and specify your choices for logging options. For example, to have PortSentry write to a dedicated log file, set SYSLOG_FACILITY to LOG_LOCAL0. Notice also that we installed PortSentry in /usr/local/portsentry, but you are free to install it somewhere else. If you do, be sure to modify the relevant fields in the header file to reflect your choice:

```
#define CONFIG_FILE "/usr/local/portsentry/portsentry.conf"

#define SYSLOG_FACILITY LOG_LOCAL0
#define SYSLOG_LEVEL LOG_DEBUG
```

2. Edit the /etc/syslog.conf file, and add a line for the PortSentry log file:

```
local0.debug    /var/adm/portsentry
mail.debug      /var/adm/tcp_wrapper
```

3. Create the PortSentry log file, and set the file permissions to prevent normal users from accessing it:

```
# touch /var/adm/portsentry
# chmod 600 /var/adm/portsentry
```

4. Edit the portsentry.conf file, and specify your PortSentry configuration choices. Refer to the comments in the portsentry.conf file and to the README.install file for more information about the various configuration options. In particular, pay special attention to the PORT_BANNER and KILL_ROUTE options:


```

# Un-comment these if you are really anal:
TCP_PORTS="1,7,9,11,15,70,79,80,109,110,111,119,138,139,143,512,513,514,515,540,
635,1080,1524,2000,2001,4000,4001,5742,6000,6001,6667,12345,12346,20034,30303,32
771,32772,32773,32774,31337,40421,40425,49724,54320"
UDP_PORTS="1,7,9,66,67,68,69,111,137,138,161,162,474,513,517,518,635,640,641,666
,700,2049,32770,32771,32772,32773,32774,31337,54321"
#
# Use these if you just want to be aware:
#TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,12346,
20034,31337,32771,32772,32773,32774,40421,49724,54320"
#UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,32770,32771,32772,32773,32774,3
1337,54321"
...
...
IGNORE_FILE="/usr/local/portsentry/portsentry.ignore"
# Hosts that have been denied (running history)
HISTORY_FILE="/usr/local/portsentry/portsentry.history"
# Hosts that have been denied this session only (temporary until next restart)
BLOCKED_FILE="/usr/local/portsentry/portsentry.blocked"
...
...
# Generic
#KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
KILL_ROUTE="/usr/sbin/route add $TARGET$ 127.0.0.1

```

5. Edit the portsentry.ignore file, and add the IP addresses of your local interfaces.
6. Edit the Makefile, remove the -Wall flag, and modify CFLAGS, INSTALLDIR, and CHILDDIR lines as follows:

```

CFLAGS = -O

INSTALLDIR = /usr/local
CHILDDIR=portsentry

```

7. Run make, make aix, and make install. In our case, the install directory is /usr/local/portsentry. This is where the portsentry executable will be located:

```

[/usr/local/portsentry-1.0] # make
Usage: make <systype>
<systype> is one of: linux, bsd, solaris, hpux, hpux-gcc,
freebsd, openbsd, netbsd, bsdi, aix, osf, generic

This code requires snprintf()/vsprintf() system calls
to work. If you run a modern OS it should work on
your system with 'make generic'. If you get it to
work on an unlisted OS please write me with the
changes.

Install: make install

NOTE: This will install the package in this
      directory: /usr/local

Edit the makefile if you wish to change these paths.
Any existing files will be overwritten.

[/usr/local/portsentry-1.0] # make aix
      SYSTYPE=aix
Making
      cc -O -o ./portsentry ./portsentry.c ./portsentry_io.c ./portsentry_util.c
...
...
/portsentry_util.c:

[/usr/local/portsentry-1.0]# make install
Creating psionic directory /usr/local
Setting directory permissions
      chmod 700 /usr/local
Creating portsentry directory /usr/local/portsentry
Setting directory permissions
      chmod 700 /usr/local/portsentry
Copying files
      cp ./portsentry.conf /usr/local/portsentry
      cp ./portsentry.ignore /usr/local/portsentry
      cp ./portsentry /usr/local/portsentry
Setting permissions
      chmod 600 /usr/local/portsentry/portsentry.ignore
      chmod 600 /usr/local/portsentry/portsentry.conf
      chmod 700 /usr/local/portsentry/portsentry

Edit /usr/local/portsentry/portsentry.conf and change
your settings if you haven't already. (route, etc)

WARNING: This version and above now use a new
directory structure for storing the program
and config files (/usr/local/portsentry).
Please make sure you delete the old files when
the testing of this install is complete.

```

8. Start PortSentry in either TCP or UDP mode.

```
[/usr/local/portsentry]# portsentry -tcp
[/usr/local/portsentry]# portsentry -udp
```

9. Test until you are satisfied that the tool is properly configured. (Note that KILL_ROUTE will only add routes to remote systems on different networks.)
10. For email alerts and better log management, you can download the LogCheck tool (<http://www.psionic.com/abacus/logcheck/>), or write your own scripts to do something similar.

6.4.2 Defense provided by PortSentry

PortSentry provides two major forms of defense:

- The first is based on TCP Wrapper. You enable this method with the WRAPPER_HOSTS_DENY option in the portsentry_config.h header file. Once enabled, IP addresses of suspected attack hosts will automatically be added to the TCP Wrapper access control file `/etc/hosts.deny`. To use this method, you must already have TCP Wrapper configured and running on the server. (For more information about installing and configuring TCP Wrapper, see Section 5.2.2, "Configuring TCP Wrapper" on page 79.) This method may, by itself, not provide sufficient defense against port scanning. Recall that TCP Wrapper only protects services started by `inetd` and then only the ones that you have wrapped with TCP Wrapper. In addition, you must make sure that the ports that you have wrapped in `/etc/inetd.conf` match what you have listed in the `portsentry.conf` file.
- The second is based on the loopback adapter. You enable this method with the KILL_ROUTE option in the `portsentry.conf` file. Once enabled, PortSentry will automatically add routes through the loopback adapter on your server to the IP addresses of the suspected attack hosts. This new route will effectively block a lot of normal port scans. Unfortunately, the AIX version of the tool that we tested did not have support for detecting and blocking stealth scans.

We recommend that you enable both methods.

6.5 List Open Files (lsof)

The `lsof` tool is quite literally used to list open files. Remember that in UNIX (and AIX) most things appear as files, including network sockets. If you find an unknown port with the `netstat` command, you can use `lsof` to get more information about it, such as ownership. The tool is available via anonymous ftp from `ftp://vic.cc.purdue.edu/pub/tools/unix/lsof`. It is also available in

installp format from the Bull site. The version we used during the writing of this redbook was version 4.50 from the Bull site.

6.5.1 Installing Isof

Download the package and ensure that it has not been tampered with. See Section 7.4.1, "Ensuring the integrity of downloads" on page 136, for more information on how to do this.

Once you have verified the integrity of the package, inflate it, and install it with either the `installp` command or SMIT:

```
# chmod u+x lsof_aix432-4.50.0.0.exe
# ./lsof_aix432-4.50.0.0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu) .
  inflating: lsof-4.50.0.0.bff
  inflating: lsof-4.50.0.0.bff.asc
# ls -l
total 2640
-rw-r--r--  1 root    system   972800 Jul  4  04:55 lsof-4.50.0.0.bff
-rw-r--r--  1 root    system    2580 Jul  4  10:38 lsof-4.50.0.0.bff.asc
-rwx-----  1 root    system  369866 Sep  1  14:48 lsof_aix432-4.50.0.0.exe
# rm .toc
# installp -acgX -d . freeware.lsof.rte
```

Alternatively, you can use SMIT to install the fileset:

1. Type **smitty installp**
2. Select **Install and Update** from ALL Available Software
3. Type **.** (dot) and press **Enter**
4. Type **F4** or **ESC+4** to list filesets
5. Select **Find freeware.lsof (F7 or ESC+7)** to select the fileset)
6. Press **Enter** three times to install

If the installation failed with a *fileset not found* message, try exiting SMIT, removing the `.toc` file (`#rm .toc`), and retrying the installation again.

Either way (`installp` or SMIT), verify that the installation completed successfully:

Installation Summary

| Name | Level | Part | Event | Result |
|-------------------|----------|------|-------|---------|
| freeware.lsof.rte | 4.50.0.0 | USR | APPLY | SUCCESS |

You can also check it with the `lslpp` command:

```
# lslpp -l freeware.lsof.rte
Fileset                Level State      Description
-----
Path: /usr/lib/objrepos
freeware.lsof.rte      4.50.0.0 COMMITTED List Open Files
```

6.5.2 Using Isof

The Isof tool is very versatile. We illustrate just one simple use for it here. Suppose you run the `netstat -af inet` command and notice some unknown ports. The output of the `netstat` command might look something like this:

```
# netstat -af inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4   0      2   arthur.telnet          merlin.32809           ESTABLISHED
tcp    0      0 *.ftp                  *.*                     LISTEN
tcp4   0      0 *.time                 *.*                     LISTEN
tcp    0      0 *.telnet               *.*                     LISTEN
tcp4   0      0 *.32768                 *.*                     LISTEN
tcp4   0      0 *.865                   *.*                     LISTEN
tcp4   0      0 *.864                   *.*                     LISTEN
tcp4   0      0 *.sunrpc                *.*                     LISTEN
udp4   0      0 *.time                 *.*                     *
udp4   0      0 *.*                     *.*                     *
udp4   0      0 *.*                     *.*                     *
udp4   0      0 *.32770                 *.*                     *
udp4   0      0 *.865                   *.*                     *
udp4   0      0 *.864                   *.*                     *
udp4   0      0 *.32768                 *.*                     *
```

Assume that UDP port 32768 is the unknown and that you want to find out which service owns it. You can run Isof as follows to find out:

```
# lsof -i udp -p 32768
lsof: WARNING: compiled for AIX version 4.3.2.0; this is 4.3.3.0.
COMMAND  PID USER  FD  TYPE  DEVICE SIZE/OFF NODE NAME
syslogd  3978 root   4u  IPv4  0x700f3a00      0t0  UDP *:syslog
inetd    5986 root    8u  IPv4  0x70030300      0t0  UDP *:time
rpc.statd 6200 root   20u  IPv4  0x700e9d00      0t0  UDP *:864
rpc.statd 6200 root   22u  IPv4  0x700e9a00      0t0  UDP *:865
```

From the output, you can see which services are using this port and take further action if necessary.

6.6 Intrusion detection

Intrusion detection software can help protect your network against attacks and alert you (via email) to take appropriate action. An attacker usually follows some form of pattern (also known as a *signature pattern*) when attacking a target. To protect your network, you can place a network sniffer (also known as sensor) on your network to monitor network activity. If an activity matches the signature of a known attack pattern, an alert can be raised forewarning you of the impending attack.

Keeping your intrusion detection software current will ensure that the latest attack patterns are known to the sensor. Remember that some forms of attack are very subtle (and slow); so, the sensitivity of the tools is key. Care should be taken to ensure that your intrusion detection software is properly configured.

Examples of intrusion detection tools are:

- ISS RealSecure
- NFR Network Flight Recorder
- Cisco NetRanger
- Computer Associates SessionWall
- Axent NetProwler

For a comparison of the different products, visit:

<http://www.stevenspublishing.com/stevens/secprodpub.nsf/PubHome/1F93DB21EE40EE8086256855005882EB?Opendocument>

As already mentioned, to protect against port scanners, such as nmap, you can use PortSentry discussed in Section 6.4, "PortSentry" on page 98. In this case, you would have PortSentry servers as sensors to alert you to possible attacks.

As another intrusion detection option, you may want to consider hiring an outside company with security experts to provide 24x7 watch over your network. In addition to intrusion detection support, these companies can also do an evaluation of your current security posture and recommend various ways to improve it. The biggest advantage to outsourcing this support is that in the unfortunate event of an attack and/or break-in, you have at your disposal security experts to help with the defense and postmortem activities. For more information, visit: <http://www.ers.ibm.com>

Chapter 7. System and data integrity

System and data integrity are important aspects of security. You need to be sure that files, be they system or data files, are not tampered with or accessed illegally. Once an attacker has penetrated the system, a common behavior is to leave backdoors and Trojan-horse programs in the system. System programs are replaced with modified versions that enable the attacker to gain easy access into the system again or that provide additional information to the attacker for future attacks on this or connected systems. Therefore, you need to have a security tool in place to give you assurance that your system files are intact.

System integrity also covers the area of password policy. Passwords should be very difficult to guess, expire regularly, be non-reusable after expiration, and never be shared. In general, shared accounts are a bad idea because of the loss of accountability, but, when they are necessary (such as with root), they should be configured to not allow direct logins. Instead, they should be configured to require the use of the `su` command. Password integrity is critical because it is almost impossible to differentiate between a valid user and an attacker who has purloined a user's password. Some scan programs (such as NSA) will check for obvious passwords, such as root for root, but to ensure password integrity, you really should have a program specifically designed for cracking bad passwords.

The privacy of your users also needs to be protected. Sensitive files need to be secured from unauthorized access. Standard UNIX file and directory permissions provide some measure of protection. However, the root user can still bypass these permissions and access the files. A better solution is to provide the users with encryption tools that they can use to protect their sensitive information (even from root).

Internet downloads are another area of concern when it comes to system and data integrity. Downloads are common in today's environment, and you need to provide a way for system users to ensure the integrity of their downloads. In other words, they need to be sure that what is downloaded is really what they expected and not something that has been tampered with by a would-be attacker.

In this chapter, we cover the following tools:

- Tripwire
- John the Ripper
- Pretty Good Privacy (*PGP*)
- MD5

Tripwire is a tool that provides assurance that systems and configuration files have not been modified. Put simply, Tripwire creates a database with the checksums of system binaries and configuration files. This database should be created on a known, clean system, such as one that has just been installed and has not yet been connected to an external network. Tripwire computes two (or more) checksums of the system files, stores them in the database, and uses them to periodically compare against the current copies of those files. If any of the files have been replaced or tampered with, Tripwire alerts you to this condition.

John the Ripper is a password cracker program. Ensuring strong passwords should be part of the security guidelines of any organization. AIX provides a number of facilities for implementing strong password policies. However, these only go so far in forcing users to pick good passwords. A tool, such as this one, will go the extra step and help you to uncover (and correct) weak passwords before an attacker does.

Pretty Good Privacy (PGP) is a popular encryption tool that you can make available for users to use to protect their sensitive files. It can also be used to securely communicate with remote users via encrypted email. PGP is a very popular tool due, in no small part, to its ease of use.

MD5 is a tool for ensuring the integrity of files downloaded from the Internet. MD5 computes a unique checksum against a file, and the algorithm that MD5 uses is very sensitive to even the slightest changes to the file. In order for this tool to be effective, the site from which you are downloading the file needs to have published the MD5 checksum. Once you have downloaded the file, run MD5 against it and compare the output with the published value. If the values are different, do not unpack and/or execute the file. Delete it, and notify the owners of the site.

7.1 Tripwire

Tripwire is a public domain tool that helps you ensure (with a high degree of certainty) that a list of system or configuration files has not been modified. A normal part of an attacker's routine is to install a backdoor into the system to allow for an easy return. This is achieved in part by replacing some system files with modified versions, such as Trojan horse programs. Tripwire is a tool to help you detect and correct this sort of thing.

During Tripwire configuration, a database is created to store all relevant security information about important files (as defined in the `tw.config` file). This database provides the security baseline for those files. Tripwire should

then be run regularly to check each of these system files to ensure that they have not been modified inappropriately. Tripwire checks the files listed in the configuration file against its database and reports any discrepancies found. To escape detection, an attacker would have to replace a system file with an identical file (size, checksums, timestamp, and so on), which is extremely difficult to do. (The reason this is so difficult is that Tripwire uses two different checksum algorithms to compute the checksums. Creating a modified system file to match both original checksums is no small task.)

For maximum protection, the database, the configuration file, and the Tripwire executable should be stored securely offline. They should only be restored to the system when you are ready to run Tripwire again. If any of the Tripwire components are compromised, Tripwire becomes worse than useless. For example, if attackers got hold of the Tripwire database, they could modify the checksums to match the altered system files that they have loaded on the system. Then, when you next run Tripwire, it appears that all is well.

Ideally, Tripwire should be run just after system installation is complete, while still in single-user mode and not connected to an external network. This ensures that you have a “known good” copy of the database. Tripwire should also be run just before and just after any changes are made to the system that alter the system files, for example, O/S upgrades, PTFs, APARs, and so on.

7.1.1 Obtaining and installing Tripwire

Tripwire was initially developed at Purdue University in 1992. In 1998, Purdue University transferred management of Tripwire to Tripwire Security Solutions, Inc. (TSS). The TSS Web site is located at <http://www.tripwiresecurity.com/>. They offer open-source and for-purchase versions of Tripwire. The for-purchase version has additional functionality that you may find useful. See <http://www.tripwiresecurity.com/products/index.cfm> for a comparison of the two different versions.

A version of Tripwire in installp format is available from the Bull site at <http://www.bull.de/pub/out/>. The version we downloaded for this redbook was Tripwire 1.2. Newer versions are available from TSS, but they are not in installp format. In other words, they will need to be compiled for AIX and manually installed.

After you download the package, ensure it has not been tampered with before installing it. Refer to Section 7.4.1, “Ensuring the integrity of downloads” on page 136, for more information about how to do this.

Inflate the package, and verify that it inflated properly:

```
# ./tripwire-1.2.1.0.exe
Archive:  ./tripwire-1.2.1.0.exe
  inflating: tripwire-1.2.1.0.bff

# ls -l
total 1480
-rw-r--r--  1 root    system  412672 Aug 09 1996  tripwire-1.2.1.0.bff
-rwx-----  1 root    system  336614 Aug 16 13:58  tripwire-1.2.1.0.exe
```

Install the fileset (with `installp` or `SMIT`), and verify that it was successfully installed:

```
# installp -acqX -d . freeware.tripwire.tripwire_bin

Installation Summary
-----
Name                               Level      Part      Event      Result
-----
freeware.tripwire.tripwire_1.2.1.0  USR        APPLY     SUCCESS
```

You can also use the `lsllp -l freeware.tripwire.tripwire_bin` command to verify the installation.

7.1.2 Configuring and using Tripwire

The important files for Tripwire are:

- The program file: `/usr/security/bin/tripwire`
- The configuration file: `/usr/adm/tcheck/tw.config`
- The database: `/usr/adm/tcheck/databases/tw.db_<hostname>`

The program file is the executable that should be run periodically to check system integrity. The configuration file contains the list of system files that you want to protect. This file needs to be created prior to the initial run of Tripwire, which uses it to populate the database. The database stores the following information about the files and directories listed in the configuration file:

- Permission and file mode bits
- Access timestamp
- Inode number
- Modification timestamp

- Number of links (ref count)
- Inode creation timestamp
- User ID of owner
- Group ID of owner
- Size of file
- Signature 1 (MD5 by default)
- Signature 2 (Snefru by default)

It is the combination of the two different signatures coupled with the file size that makes it extremely difficult to create another file with the same characteristics. Note that the default configuration file was created for AIX 4.1.5; so, you will most likely need to modify it for your systems.

Tripwire runs in one of four modes:

- Database Generation
- Database Update
- Integrity Checking
- Interactive Update

In *Database Generation* mode, Tripwire creates the database (/usr/adm/tcheck/databases/tw.db_<hostname>) based on the configuration file. To initialize the database:

```
# cd /usr/adm/tcheck
# /usr/security/bin/tripwire -initialize
```

Important

Be sure that your current working directory is set to /usr/adm/tcheck before initializing the database. Tripwire creates the `databases` subdirectory in the current working directory and then creates the database under the `databases` directory.

In *Database Update* mode, the database is updated with the current configuration information. This is required when legitimate changes are made to the system, such as changes to the system files or application of operating system patches:

```
# /usr/security/bin/tripwire -update <path_of_changed_file_or_directory>
```

In *Integrity Checking* mode, Tripwire generates a report of added, deleted, or changed files. This is done by comparing the current configuration information against that which is stored in the database (/usr/adm/tcheck/databases/tw.db_<hostname>):

```
# /usr/security/bin/tripwire | tee /tmp/tripwire.out
```

In *Interactive Update* mode, rather than just reporting things that have changed, Tripwire also prompts you to apply these changes to its database:

```
# /usr/security/bin/tripwire -interactive
```

7.1.3 Configuring Tripwire

The most important step in the configuration of Tripwire is creating an accurate and complete configuration file. The configuration file should contain the list of system files and directories that you want to have Tripwire protect. You can either start with the supplied Tripwire configuration file, which was created for AIX 4.1.5, or create your own from scratch. We used the former technique:

1. Use /etc/security/sysck.cfg to create an initial delta file for the Tripwire configuration file:

```
# cat /etc/security/sysck.cfg | grep ":" | sed s://g | \  
>sort > tripwire.list.sort  
  
# chmod 600 tripwire.list.sort
```

2. Modify the tripwire.list.sort file with additional files and directories that you deem important. We added the following:

```

/.kshrc      R      # may not exist
/usr/lib/boot/unix_mp  R
/usr/lib/boot/unix_up  R
/usr/lib/boot/unix_kdb R
/usr/lib/boot/unix_mp_kdb  R
/unix        R
/etc/security/limits  R
/etc/security/login.cfg R
/usr/lib/objrepos  L  # files frequently modified
/etc/objrepos      L
/usr/bin/whoami     R
/usr/bin/sh         R
/usr/bin/Rsh        R
/usr/bin/tsh        R
/usr/bin/bsh        R
/usr/bin/ls         R
/usr/adm/wtmp       L
## protect other binaries you installed in your system, Eg security tools
/usr/local/bin/mmap R
/usr/local/bin/ssh  R
/usr/local/bin/pgp  R
/usr/local/bin/lsof R

```

3. Merge the contents of this file with `tw.config` file that came with the Tripwire package.
4. Verify the configuration file with the Tripwire `CheckConfig` script. (See the `/usr/local/lib/tripwire-1.2/README.CheckConfig` file for more information about how to use this script.):

```

# /usr/local/lib/tripwire-1.2/CheckConfig -v /usr/adm/tcheck/tw.config
/:      directory
/.rhosts:  file
/.netrc:  not found
/.profile: file
/.cshrc:  not found
/.login:  not found
/.kshrc:  file
/usr/lib/boot/unix_mp: file
/usr/lib/boot/unix_up: not found
/usr/lib/boot/unix_kdb: not found
/usr/lib/boot/unix_mp_kdb:  file
/unix:    symlink
/dev:    directory
/sbin:   directory
/usr/local:  directory
/usr:     directory

```

5. Based on the output of the `CheckConfig` script, you may need to make additional changes to the `tw.config` file and run `CheckConfig` again.
6. Once you are satisfied with the output of `CheckConfig`, initialize the database:

```
[/usr/adm/tcheck] # /usr/security/bin/tripwire -initialize
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
### Phase 3:  Creating file information database
###
### Warning:  Database file placed in ./databases/tw.db_sp5en01.
###
###          Make sure to move this file file and the configuration
###          to secure media!
###
###          (Tripwire expects to find it in '/usr/adm/tcheck/databases')
```

7. When you first initialize the Tripwire database, it reports what it thinks are inconsistencies. Update the configuration file with the requested changes, and then reinitialize the database. (As long as your system is still off the network, you can be sure that these changes are a byproduct of normal system behavior.):


```

[/usr/adm/tcheck] # /usr/security/bin/tripwire
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
### Phase 3:  Creating file information database
### Phase 4:  Searching for inconsistencies
###
###                Total files scanned:          6667
###                Files added:                  0
###                Files deleted:                0
###                Files changed:                6111
###
###                After applying rules:
###                Changes discarded:            6108
###                Changes remaining:        5
###
changed: -rwxrwxr-- root      27029 Jul 20 14:16:31 1999 /sbin/rc.boot
changed: -rw-r--r-- root      1826 Sep 11 18:54:53 2000 /etc/inittab
changed: -rw-r----- root     1412 Sep 11 18:55:36 2000 /etc/security/lastlog
changed: -rw----- root         0 Sep 11 18:55:26 2000 /etc/pmd_lock2
changed: -rw----- root         0 Sep 11 18:55:26 2000 /etc/pmd_lock
### Phase 5:  Generating observed/expected pairs for changed files
###
### Attr          Observed (what it is)          Expected (what it should be)
### =====
/sbin/rc.boot
  st_ctime: Mon Sep 11 18:53:50 2000      Sun Sep 10 23:15:04 2000

/etc/inittab
  st_ino: 403                             405

/etc/security/lastlog
  st_mtime: Mon Sep 11 18:55:36 2000      Mon Sep 11 17:45:35 2000
  st_ctime: Mon Sep 11 18:55:36 2000      Mon Sep 11 17:45:35 2000
  md5 (sig1): 2noet1FpnqVTw1XwfZEL9f     25zTiUps0WezoeLB7nJUki
  snefru (sig2): 2egxLZbqhjIm6g64gpYOSR   2jbnbYfhkSkdJWLppnIFF.

/etc/pmd_lock2
  st_ino: 409                             476

/etc/pmd_lock
  st_ino: 476                             477

```

8. Repeat this process as many times as needed until Tripwire reports that the changes remaining are zero:

```

[/usr/adm/tcheck]# /usr/security/bin/tripwire
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
### Phase 3:  Creating file information database
### Phase 4:  Searching for inconsistencies
###
###                Total files scanned:      6667
###                Files added:              0
###                Files deleted:            0
###                Files changed:           6010
###
###                After applying rules:
###                Changes discarded:        6010
###                Changes remaining:        0
###

```

9. Once the Tripwire database has been successfully initialized, store the database, the configuration file, and the tripwire executable in a safe area, such as on another secure machine, on read-only media, or on the local machine encrypted with PGP. To run Tripwire again, you will need to put the database back in `/usr/adm/tcheck/databases/`.

7.1.4 Comments on configuration

A good deal of thought and planning need to go into the creation of the Tripwire configuration file (`tw.config`). Just one key system file missing from the configuration file could open the door for an attacker. Important system files should be explicitly called out in the `tw.config` file, rather than assuming they will be picked up by a directory specification, such as `/sbin` or `/usr/sbin`.

The goal is to minimize the Tripwire report of changed files to only those that change because of normal day-to-day activities. This makes it much easier to spot suspicious activities called out within the report. Be forewarned that this is an iterative process. You need to spend time running Tripwire, examining the report, modifying the configuration file, and running Tripwire again. It takes some time to refine it into a tool that works well in your environment. Be sure to safely store the configuration file, the database, and the executable program after each iteration. The strength of Tripwire depends solely on the integrity of these files.

7.1.5 When should Tripwire be run

Ideally, Tripwire should be initialized immediately after system installation and configuration while still in single-user mode and not connected to an external network.

Run Tripwire periodically to ensure that no unauthorized changes have been made to important system files or directories. Also, run it just before and just after any changes to the system for things, such as operating system upgrades, PTFs, APARs, new software installs, and so on. As always, be sure to secure the modified copies of the Tripwire configuration file and database.

7.1.6 Alternatives to Tripwire

Other alternatives to Tripwire are:

- Trusted Computing Base (TCB)
- COPS
- Tamu's Tiger

Trusted Computing Base (TCB) is a feature that comes with AIX. There are a number of drawbacks with TCB, not the least of which is that it can only be enabled during AIX installation (not after). For more information about TCB, see Chapter 6 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962.

For details on COPS and Tamu's Tiger, see Chapter 7 of *Exploiting RS/6000 SP Security: Keeping it safe*, SG24-5521.

7.2 John the Ripper

John the Ripper is a tool for password cracking. Weak passwords are one of the most common ways in which attackers gain unauthorized access to systems. This is one of the most insidious forms of attack, because it is very difficult to differentiate between a valid user and an attacker who has purloined a valid user's password. Having John the Ripper proactively scan the password file is one way to ensure that passwords are sufficiently strong.

Your first step in ensuring strong passwords is to educate the users of your system on what constitutes a strong password. Here are some guidelines taken from the *AIX 4.3 Elements of Security* redbook:

- Do not use your user ID or any permutation of it.
- If you use the same password on more than one system, be extra careful with it. Never use the same root password on multiple systems.
- Do not use any person's name.
- Do not use words that can be found in the online spelling-check dictionary, especially for a networked or larger multi-user system.
- Do not use passwords shorter than five or six characters.

- Do not use swear words or obscene words; these are among the first words tried when guessing passwords.
- Do use passwords that you can remember. Do not write down your password.
- Do consider passwords that consist of letters and numbers.
- Do use passwords that you can type quickly.
- Two words, with a number in between, make a good password.
- A word (with at least six characters), with a numeric digit inserted in the word, is an excellent password. But do not form the digit by changing an “l” to “1” or an “o” to “0.” A word with an internal digit is a better password than a word with a leading or trailing digit.
- A pronounceable password is easier to remember.
- AIX checks only the first eight characters of the password; however the word can be longer than eight characters.
- A good scheme is to memorize a sentence and use the first letter of each word for the password. For example, “the cat sat on the mat” = tcsotm.

Note that some of these guidelines can be enforced through facilities provided by AIX, but many of them still require good education and ongoing enforcement. John the Ripper is a very good tool to help you with the enforcement piece.

7.2.1 Obtaining and installing John the Ripper

The main site for of John the Ripper is <http://www.openwall.com/john/>. You can download the tool from this site in source-code format, or can download it in installp format from the Bull site. The version that we used for this redbook was version 1.6 from the Bull site.

Download the package and ensure that it has not been tampered with before installing it. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136 for more information on how to do this.

Inflate the package, and install it (via installp or SMIT):

```

# ls -l
total 1048
-rwx----- 1 root      system   535904 Sep 12 09:54 john-1.6.0.0.exe

# ./john-1.6.0.0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: john-1.6.0.0.bff

# installp -acgX -d . freeware.john.rte
Installation Summary
-----
Name                               Level           Part           Event           Result
-----
freeware.john.rte                  1.6.0.0        USR            APPLY           SUCCESS

# lsllp -l freeware.john.rte
Fileset                             Level  State      Description
-----
Path: /usr/lib/objrepos
freeware.john.rte                   1.6.0.0  COMMITTED  John the Ripper Password
                                         Cracker

```

Once the fileset has successfully been installed, change the ownership and permissions on the executable (`/usr/local/bin/john`) such that root is the only one allowed to run it.

Important

Many organizations have very strict policies against unauthorized use of tools like John the Ripper. Make sure you have the proper written authorization in place before running this tool.

7.2.2 Configuring John the Ripper

There are three modes that John the Ripper uses to crack passwords:

- Single Crack
- Wordlist
- Incremental

The *Single Crack* mode is the fastest. It uses the login/GECOS information from the `/etc/passwd` file as input.

The *Wordlist* mode takes as input a file containing one word per line. A wordlist file (`/usr/local/bin/john-1.6/password.lst`) is provided with the package. It is a rather short list, so you should augment it with additional words. For example, you can merge it with the `/usr/share/dict/words` file from the `bos.data` fileset (found on the AIX Installation CD), as follows:

```
#export JOHNDIR=/usr/local/lib/john-1.6

#cp /usr/share/dict/words $JOHNDIR

#cat $JOHNDIR/password.lst >> $JOHNDIR/words

#/usr/bin/tr A-Z a-z < $JOHNDIR/words | \
sort -u > $JOHNDIR/words.list.sort
```

Once merged, edit the Wordfile entry in the john.ini configuration file to reflect the new filename (/usr/local/lib/john-1.6/words.list.sort).

The *Incremental* mode is the most powerful. It uses all possible character combinations to ultimately crack all passwords. However, because of the length of time it needs to do this, it is generally not practical to have it run this way without modification. You can set parameters in the \$JOHNDIR/john.ini file to control this mode of operation and turn it into something more realistic and practical.

7.2.3 Using John the Ripper

Important

As has already been mentioned, ensure that you have the necessary written authorization in place *before* attempting to use John the Ripper.

John the Ripper requires that the password file be a single file. Since AIX implements a shadow password file, you need to merge the /etc/passwd and /etc/security/passwd files into a single file. Appendix C, “Script to merge the AIX passwd files” on page 209 provides a listing of the shadmrg.aix script (obtained from the Crack package) to merge these two files. (Alternatively, you can manually merge the contents of the two files by replacing the “|” mark in the /etc/passwd file with the encrypted password in the /etc/security/passwd file.):

```
# ./shadmrg.zix > /tmp/password
# chmod 600 /tmp/password
- OR -

# ./shadmrg.aix | egrep -v "useridA|useridB|useridC" > /tmp/password
# chmod 600 /tmp/password
```

This command runs the shadmrg.aix script to create the merged password file (/tmp/password). Use the egrep command to filter out the users you do not

want to scan. Also, be sure to change the file permissions for the `/tmp/password` file so that only root has access to it.

Important

After installation, change the file permissions of the John the Ripper executable (`/usr/local/bin/john`) so that only root has access to it. The same goes for the merged password file (`/tmp/password`).

Run John the Ripper as follows:

```
# /usr/local/bin/john /tmp/password
```

By default, John the Ripper uses the three modes (Single Crack, Wordlist, and Incremental, respectively) to attempt to crack the passwords. Successful attempts are sent to the screen and saved in the `$JOHNDIR/john.pot` file. If you want to see them again, re-run John the Ripper with the `-show` option:

```
# /usr/local/bin/john -show /tmp/password
```

Here is an example that illustrates the difference in time it takes to crack a password with a digit on the end versus one with a digit in the middle:

```
# /usr/local/bin/john /tmp/password
Loaded 2 passwords with 2 different salts (Standard DES [32/32 BS])
rootroot          (root)
merlin1           (khorck)
guesses: 2  time: 0:00:00:03 100% (2)  c/s: 22715  trying: menul - meterl
```

The passwords for root (password `rootroot`) and khorck (password `merlin1`) are cracked in 3 seconds. Now, let's change khorck's password from `merlin1` to `mer1n`, and re-run John the Ripper to see what happens:

```
# /usr/local/bin/john /tmp/password
Loaded 2 passwords with 2 different salts (Standard DES [32/32 BS])
rootroot      (root)
merlin        (khorck)
guesses: 2  time: 0:00:28:24 100% (3)  c/s: 23710  trying: Booms2 - hilsuh
```

Notice that Jack the Ripper now took 28.5 minutes to crack the new khorck password. This illustrates that having numbers in the middle of passwords, rather than on either end, makes them significantly harder to crack.

Note

A strong password is a good defense against brute force cracking methods. However, even the strongest password is vulnerable to network sniffing with protocols, such as telnet, ftp, rsh, and so on. Refer to Chapter 5, "Secure remote access" on page 59, for ways to protect against this type of attack.

Additional methods, such as secondary authentication or two-person login, are available to make authentication more difficult to crack. For more information, refer to *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962.

7.3 Pretty Good Privacy (PGP)

PGP is a popular tool used to communicate between two parties securely. This security is achieved by strong encryption and authentication. PGP is a de facto standard used in email to quickly and easily enable secure communications. It runs on various operating systems, including UNIX, Linux, and Windows, and it has plug-ins available for email programs, such as Lotus Notes, Eudora, and Outlook Express.

For more information about supported configurations, see the PGP FAQ at:

<http://www.pgpi.org/doc/faq/pgpi/en/#International>

7.3.1 PGP basics

PGP is based on public key cryptography (see Section 5.1.3, "Key concepts of SSH" on page 62). In a nutshell, files encrypted with the public key

(asymmetric cryptography) can only be decrypted by the matching private key and vice-versa.

By definition, a public key is “public” and can be freely distributed to the other individuals you want to communicate with using PGP. The private key is “private” and should be kept secure and private. The private key can be further protected with the use of a pass phrase.

Straight public-key encryption is much slower (sometimes by as much as a factor of 1000) than conventional single-key encryption (symmetric cryptography). It is for this reason that PGP uses a hybrid of both symmetric and asymmetric cryptography.

With PGP, messages are encrypted using conventional symmetric cryptography (the same key is used to encrypt and decrypt the message). However, this key is a random one-time key (*session key*) that is used for this session only. Now, the main weakness of symmetric cryptography is in how to share the same key securely. PGP overcomes this weakness by using asymmetric cryptography.

Here is how PGP does it:

1. Jack wants to communicate securely with Jill using PGP.
2. Jill sends Jack her public key.
3. Jack encrypts the data with the session key (symmetric cryptography), then uses Jill's public key to encrypt the session key (asymmetric cryptography). Once encrypted, Jack sends both the session key and the data to Jill.
4. Jill uses her private key to decrypt the session key (asymmetric cryptography) and then uses the session key to decrypt the data (symmetric cryptography).
5. From now on, Jack and Jill will just use the session key to encrypt and decrypt the data flowing back and forth between them.

In this case, both symmetric and asymmetric algorithms are used, the symmetric algorithm used to encrypt the data itself and the asymmetric algorithm used to securely transfer the encryption key. With this hybrid method, both speed and security are achieved.

PGP supports three symmetric ciphers: CAST (default), Triple DES, and IDEA. It also supports either DSS/DH keys or RSA (IDEA only) keys. A comparison of each can be found in the PGP man pages (`man pgp-intro`).

PGP also uses digital signatures for message authentication. Digital signatures serve the same function as that of real signatures, but in the digital world. A digital signature is created by encrypting the message digest (a strong one-way hash function of the message) with the sender's private key. The digital signature is sent together with the message. To verify the digital signature, the recipient uses the sender's public key to decrypt and obtain the message digest. Then, the recipient executes a one-way hash function on the message to obtain a message digest and compares this output with that sent by the sender (the output decrypted with the senders' public key). Both must be the same to ensure data integrity. At the same time, the author of the message is verified because the message was signed with the private key of the sender (thus the importance of protecting the private key). For more information on this process, see the PGP man pages (`man pgp-intro`).

7.3.2 Obtaining and installing PGP

PGP was created by Phil Zimmermann and is available from Network Associates (<http://www.pgp.com>). In addition to PGP, Network Associates has a host of other security-related products, ranging from firewalls to intrusion detection software to consultancy services. PGP is available in both commercial and non-commercial packages. See the Network Associates site for more details on the licensing agreements.

PGP is available for download in source-code form from the PGP International site (<http://www.pgpi.com>). Commercial licenses are available from <http://www.nai.com> (US or Canada only) or from <http://www.pgpinternational.com> (everywhere else).

PGP is also available in installp format from <http://www-frec.bull.fr>. The version we used for this redbook was version 5.0 from the Bull site.

Download the package and ensure that it has not been tampered with before installing it. Refer to Section 7.4.1, "Ensuring the integrity of downloads" on page 136, for more information on how to do this.

Inflate the package, install it (via installp or SMIT), and verify that it is successfully installed:

```

# ./pgp-5.0.0.0.exe
UnZipSFX 5.31 of 31 May 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: pgp-5.0.0.0.bff
# rm .toc
# ls -l
total 11776
-rw-r--r--  1 root    system  4403200 Mar 06 1998  pgp-5.0.0.0.bff
-rwx-----  1 root    system  1622544 Aug 24 14:49  pgp-5.0.0.0.exe
# rm .toc
# installp -acgX -d . freeware.pgp.rte
...
Installation Summary
-----
Name                               Level      Part      Event      Result
-----
freeware.pgp.rte                   5.0.0.0   USR       APPLY      SUCCESS

# lsllp -l freeware.pgp.rte
Fileset                            Level  State      Description
-----
Path: /usr/lib/objrepos
freeware.pgp.rte                   5.0.0.0  COMMITTED  Pretty Good Pricacy E-mail
system

```

You may also want to update the `MANPATH` environment variable to include `/usr/local/man` and the `PATH` environment variable to include `/usr/local/bin` in `/etc/profile`.

7.3.3 Using PGP

The first step to using PGP is to generate a public-private key pair. Like SSH, PGP can be used by anyone, not just root. Each user will have their own set of key files and configuration files.

Generate your public-private key pair with the `pgpk -g` command:

```

/ # pgpk -g
Creating /.pgp...complete.
No randseed file found.
Cannot open configuration file /.pgp/pgp.cfg
Choose the type of your public key:
  1) DSS/Diffie-Hellman - New algorithm for 5.0 (default)
  2) RSA
Choose 1 or 2: 1

Pick your public/private keypair key size:
(Sizes are Diffie-Hellman/DSS; Read the user's guide for more information)
  1) 768/768 bits- Commercial grade, probably not currently breakable
  2) 1024/1024 bits- High commercial grade, secure for many years
  3) 2048/1024 bits- "Military" grade, secure for foreseeable future (default)
  4) 3072/1024 bits- Archival grade, slow, highest security
Choose 1, 2, 3 or 4, or enter desired number of Diffie-Hellman bits
(768 - 4096): 2

You need a user ID for your public key. The desired form for this
user ID is your FULL name, followed by your E-mail address enclosed in
<angle brackets>, if you have an E-mail address. For example:
  Joe Smith <user@domain.com>
If you violate this standard, you will lose much of the benefits of
PGP 5.0's keyserver and email integration.

Enter a user ID for your public key: root <root@arthur.alibaba.ibm.com>

Enter the validity period of your key in days from 0 - 999
0 is forever (and the default):

You need a pass phrase to protect your private key(s).
Your pass phrase can be any sentence or phrase and may have many
words, spaces, punctuation, or any other printable characters.
Enter pass phrase:
Enter again, for confirmation:
Enter pass phrase:
Collecting randomness for key...

We need to generate 571 random bits. This is done by measuring the
time intervals between your keystrokes. Please enter some random text
on your keyboard until you hear the beep:
  0 * -Enough, thank you.
.***** '[o.....***** .
.....*****
.....*****
Keypair created successfully.

If you wish to send this new key to a server, enter the URL of the server,
below. If not, enter nothing.
/ # cd .pgp
/.pgp # ls -l
total 24
-rw----- 1 root system 0 Sep 05 11:50 pubring.bak
-rw----- 1 root system 889 Sep 05 11:51 pubring.pkr
-rw-r--r-- 1 root system 512 Sep 05 11:53 randseed.bin
-rw----- 1 root system 0 Sep 05 11:50 secring.bak
-rw----- 1 root system 976 Sep 05 11:51 secring.skr

```

The important files to note are the `$HOME/.pgp/pubring.prk` file and the `$HOME/.pgp/secring.skr` file, which contain the public and private keyrings, respectively. These are the default names and locations for these files, but you can configure PGP to use different names and locations if desired.

You can view the contents of your keyring using the `pgpk -l` command, and you can use the `pgpk -x <userid>` command to extract your public key to a file for distribution:

```

./pgp # pgpk -l
Cannot open configuration file /.pgp/pgp.cfg
Type Bits KeyID      Created Expires Algorithm      Use
sec+ 1024 0x0592C6DE 2000-09-05 ----- DSS                Sign & Encrypt
sub  1025 0x76580B38 2000-09-05 ----- Diffie-Hellman
uid  root <root@arthur.alibaba.ibm.com>

1 matching key found

./pgp # pgpk -x root > publickeyfile
Cannot open configuration file /.pgp/pgp.cfg

./pgp # more publickeyfile
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 5.0i for non-commercial use

mQGIBDm1FoQREAD8KcZ/FCH+QtQQRauJpBHLK2Ciwve1D7Zts8zqhsiGz+ywZxiB
lmsglopAW1w/LzpMLET0g4JhSSNnAi1H02KmBKrNCY4b2gB3UCX0bIxahamyTKU1
aDSFwthos5+PLbKSDB1WjVohLSei1WluFWSrO6B7sM0fO1+K0pYx8HrCEwCg/+aO
SDTzQp+BM2kBTYM6CKGbThMEAM+eE70adaiAm50KZRefSThuDgbvWpYkCZZs+i0X
L5KBRsnw26z1MF8xFPw+KtF8N+Tgn6td+qOxL1b5x2jOxoHA1Y2yp0yG6ExJfxN7
Ni6YcVrUHQBpbsOE9cJTLXgsugbwACQRTq0F/LkFquX01BguAltI.PH4FC87QEA0u
AutLBADBBrviGRVzVtSVYcb1hKBY4bRnEZswvThc0s710ZJGi54zH5RgvwItmsmd4
OgPBwZSaAuoaI5j3U4nHb1hJE41e/sWS+FO7Bgy6dFZ8qK/e5ibmE2nW29N6R9ZN
+E2KkWr7JNzbJN1vJAorJIYIYqgO9qfMXEKekGmLXCMk2r1Cv7Qi.cm9vdCA8cm9v
dEBhcnRodXluYwXpYmFiYS5pYm0uY29tPokASwQQEQIACwUCObUWhAQLAwECAAoJ
EHOk2gQFksbeL8wAcO1E50QLVmdpEFGCC3MUsSzLbZ0SAJ9TWFX0a4V9qIbHKYwu
tP+813BgnrkBDgQ5tRaREQAQEAQEVYvcxln++/zRqA2eQqNr4IrrHluZvHwyhhINo
6cCy+oANKzUjmXJajGIMCwT6wbO+koVSA8kYPz1SG5QpdEE9+2XjJaaB/csBMvN8
jvOjBBXrnGjEFAIwosAU1cvvmLwJPhGvh00cSbSX9ELj/iD0hqXABW6S1pdnRoRa
AR1PAAICA/98YOZxJl87uG1xdKhn4PDdECTEdMOqVE1oq3pECTQQgLa8WYZmfiiR
EPGVT7eBG101at2FrSDT0ac+Fd0LAAbRCw/0XRibPIoMbYE3ENUrX16tuCCQtT6G
EXRe20xLS2wKcG/QtyXyPH3TbF9gH/IyKqXUnCsFI552Xcda4WjgF64kAPwMFGDm1
FpFzpnqkBLZLG3hECWDYAoLco/XRRLjAlqKQR9yUQsyAdwCcZAKCwXGoKTFauuYuZ
DgnXLJjXybmxEA==
=hX2t

```

In this example, root's public key is extracted to a file named `publickeyfile`. The `publickeyfile` can then be distributed to other users with whom you need to communicate securely.

To distribute the public key, you can:

- Make the file to which you extracted the public key available to those who need it.
- Include the public key as attached text within an email message, and send it to those who need it.
- Distribute the public key through a public key server (details available in the PGP man pages).

To add a public key (`khork.asc` in this case) to your keyring, use the `pgpk -a` command:

```
# pgpk -a /home/khorck/.pgp/khor.asc
Cannot open configuration file /.pgp/pgp.cfg

Adding keys:

Key ring: '/home/khorck/.pgp/khor.asc'
Type Bits KeyID      Created Expires Algorithm      Use
pub  1024 0xF409CB5B 2000-09-05 ----- DSS                Sign & Encrypt
sub  1024 0xDC8183A1 2000-09-05 ----- Diffie-Hellman
uid  khor chune keat

1 matching key found

Add these keys to your keyring? [Y/n] y

Keys added successfully.

/.pgp # pgpk -l
Cannot open configuration file /.pgp/pgp.cfg
Type Bits KeyID      Created Expires Algorithm      Use
pub  1024 0xF409CB5B 2000-09-05 ----- DSS                Sign & Encrypt
sub  1024 0xDC8183A1 2000-09-05 ----- Diffie-Hellman
uid  khor chune keat

sec+ 1024 0x0592C6DE 2000-09-05 ----- DSS                Sign & Encrypt
sub  1025 0x76580B38 2000-09-05 ----- Diffie-Hellman
uid  root <root@arthur.alibaba.ibm.com>

2 matching keys found
```

You can use the `pgpk -ll` command to obtain detailed information of the keys on your keyring. The important information is the key length, the KeyID, and the Fingerprint. You can use this information to verify with the key owner (by phone or other means) that you are using the correct version of their public key:

```

# pgpk -ll
Cannot open configuration file /.pgp/pgp.cfg
Type Bits KeyID      Created Expires Algorithm Use
pub 1024 0xF409CB5B 2000-09-05 ----- DSS          Sign & Encrypt
f20 Fingerprint20 = 2334 815A FC8B ABEC 4AEF 8D3C 5284 2B3B F409 CB5B
sub 1024 0xDC8183A1 2000-09-05 ----- Diffie-Hellman
f20 Fingerprint20 = 135C F183 BFC8 2B1A C9CC 9BB8 618C 33E1 DC81 83A1
uid khor chune keat
sig 0xF409CB5B 2000-09-05 khor chune keat

sec+ 1024 0x0592C6DE 2000-09-05 ----- DSS          Sign & Encrypt
f20 Fingerprint20 = A594 B67E 5DFA BBEC 9FDE A59A 73A4 DAA4 0592 C6DE
sub 1025 0x76580B38 2000-09-05 ----- Diffie-Hellman
f20 Fingerprint20 = 83E2 8BC0 991F AC40 FBOE BE10 4183 8C47 7658 0B38
uid root <root@arthur.alibaba.ibm.com>
SIG 0x0592C6DE 2000-09-05 root <root@arthur.alibaba.ibm.com>

2 matching keys found

```

You can display even more information about the keys in your keyring (such as Trust level) with the `pgpk -c` command:

```

$ pgpk -c
Cannot open configuration file /home/khorck/.pgp/pgp.cfg
Type Bits KeyID      Created Expires Algorithm Use
sec+ 1024 0xF409CB5B 2000-09-05 ----- DSS          Sign & Encrypt
sub 1024 0xDC8183A1 2000-09-05 ----- Diffie-Hellman
uid khor chune keat
SIG! 0xF409CB5B 2000-09-05 khor chune keat

pub 1024 0x0592C6DE 2000-09-05 ----- DSS          Sign & Encrypt
sub 1025 0x76580B38 2000-09-05 ----- Diffie-Hellman
uid root <root@arthur.alibaba.ibm.com>
sig! 0x0592C6DE 2000-09-05 root <root@arthur.alibaba.ibm.com>

KeyID      Trust      Validity User ID
* 0xF409CB5B ultimate complete khor chune keat
khor chune keat
0x0592C6DE untrusted invalid root <root@arthur.alibaba.ibm.com>
root <root@arthur.alibaba.ibm.com>

```

Notice that the Trust value for root's public key is untrusted. That simply means that the public key that you imported for root did not come from a trusted source, such as a Certificate Authority. When this is the case, it is a very good idea to use the `pgpk -ll` command and verify the key length, KeyID, and Fingerprint with the public key owner to ensure that you have the correct key.

Once both parties have each other's public keys, they can communicate securely using PGP:

- The sender encrypts the message with the receiver's public key, then sends the message.
- The receiver uses their private key to decrypt the message, then reverses the process to send the reply.

To encrypt a file, you need to know the recipient's name. PGP then uses the corresponding public key to encrypt the file:

```
# pgpe -r "khor chune keat" -sat /tmp/secretfile
Cannot open configuration file /.pgp/pgp.cfg
A private key is required to make a signature.
Need a pass phrase to decrypt private key:
 1024 bits, Key ID 0592C6DE, Created 2000-09-05
"root <root@arthur.alibaba.ibm.com>"
Enter pass phrase:
Pass phrase is good.
 1024 bits, Key ID F409CB5B, Created 2000-09-05
"khor chune keat"
WARNING: The above key is not trusted to belong to:
khor chune keat

Do you want to use the key with this name? [y/N] y

Creating output file /tmp/secretfile.asc
```

In this case, the `pgpe -r "recipient name" -sat /tmp/secretfile` command encrypts and signs (`-s` flag) the `/tmp/secretfile` file. Since the message is signed with your private key, you also need to enter the pass phrase. The `-at` flags cause the output of the command to be in text format for portability between different operating systems. The `/tmp/secretfile.asc` output file is automatically created. This file contains the encrypted contents which you send to the recipient. The warning message simply means that the public key was not obtained from a trusted source, such as a Certificate Authority.

Note that signing and encrypting are independent events; so, you can sign a file without encrypting it, and you can encrypt a file without signing it. (Use the `pgps` command if you want to sign a file).

Once the recipient receives the encrypted file, they can decrypt it with the `pgpv` command:


```

$ pgpv /tmp/secretfile.asc
Cannot open configuration file /home/khorck/.pgp/pgp.cfg
Message is encrypted.
Need a pass phrase to decrypt private key:
 1024 bits, Key ID DC8183A1, Created 2000-09-05
Enter pass phrase:
Pass phrase is good.
Opening file "/tmp/aaa" type text.
Good signature made 2000-09-05 18:15 GMT by key:
 1024 bits, Key ID 0592C6DE, Created 2000-09-05
  "root <root@arthur.alibaba.ibm.com>"

WARNING: The signing key is not trusted to belong to:
root <root@arthur.alibaba.ibm.com>

```

Notice the prompt for the pass phrase. Because the file was encrypted with the recipient's public key, it must be decrypted with their private key, and use of the private key requires a pass phrase. Notice also the `Good signature` line, which is very important for ensuring the integrity of the message. The warning message simply means that the public key was not obtained from a trusted source, such as a Certificate Authority.

PGP can also be used to encrypt sensitive files for secure storage (locally or remotely). To use PGP to encrypt a file for yourself, simply encrypt it with you as the designated recipient:

```

# pgpe -r "root" -sat /tmp/secretfile
Cannot open configuration file /.pgp/pgp.cfg
A private key is required to make a signature.
Need a pass phrase to decrypt private key:
 1024 bits, Key ID 0592C6DE, Created 2000-09-05
  "root <root@arthur.alibaba.ibm.com>"
Enter pass phrase:
Pass phrase is good.
 1024 bits, Key ID 0592C6DE, Created 2000-09-05
  "root <root@arthur.alibaba.ibm.com>"

Creating output file /tmp/secretfile.asc

# rm /tmp/secretfile

```

Note that there is no reason to save the `/tmp/secretfile` file since what you really wanted was the signed, encrypted `/tmp/secretfile.asc` file.

PGP is widely used on the Internet to provide assurance (through digital signatures) that downloadable files have not been tampered with. For example, packages from the Bull site often include a file with a `.asc`

extension. Read this file for further instructions on how to verify the integrity of the download prior to installation of the fileset(s).

A downloaded signature file can contain the signature and the message in a single file, or just the signature without the message. In the latter case, you need to have the message file available (through a separate download) for PGP to be able to verify the signature. Most Internet sites provide the signature and message in separate files.

In the case of the Bull site, the signature file (.asc) and the message file (.bff) are included together in the same package, but the signature file does not contain the message file. To verify the message file (.bff), first, obtain the public key of the file's creator and add it to your keyring; then, run the `pgpv` command against the signature file (.asc) to verify the integrity of the download. Additional information on ensuring the integrity of downloads can be found in Section 7.4.1, "Ensuring the integrity of downloads" on page 136.

7.3.4 Protecting your private key

As you can see, private keys are the key to security within PGP (and all public-private key encryption schemes, for that matter). Private keys must be kept private and secure at all times. Just as important as the physical security of your private key is the choice of a good pass phrase. Pass phrases should be 10-30 characters in length, extremely difficult to guess, and have meaning only for you. The PGP man pages (`man pgp-intro`) contain additional suggestions on ways to protect your keys.

7.4 MD5

MD5 is a tool used to produce a unique checksum, known as the MD5 checksum. It is a strong, one-way *hash function*. Recall that a hash function takes an input of arbitrary length and produces an output of fixed length. *Any* change to the input will result in a totally different output (though of the same length). It is one way because you cannot reproduce the original input if given the hashed output. A detailed description of MD5 can be found in RFC 1321, which is included in the package.

The MD5 package is available in source-code form from Purdue University:

```
ftp://ftp.cerias.purdue.edu/pub/tools/unix/crypto/md5/
```

The site also contains a digital signature so that you can verify the integrity of the package. In addition to the package, you will need access to a C compiler. In our case, we used version 4.4 of the C compiler for AIX under AIX 4.3.3.

Important

For security reasons, it is not a good idea to have a C compiler loaded on a production system. Compile the program on a non-production system, and only move the executable program to the production system. Also, be sure to set the permissions as tightly as possible.

Unpack, compile, and install the package as follows:

```
# ls -l
total 72
-rw-r----- 1 root    system    35287 Sep 10 20:41 MD5.tar.Z
# uncompress MD5.tar.Z
# tar -xvf MD5.tar
x Makefile, 1644 bytes, 4 media blocks.
x README, 5568 bytes, 11 media blocks.
x global.h, 781 bytes, 2 media blocks.
x md5-announcement.txt, 1898 bytes, 4 media blocks.
x md5.1, 1288 bytes, 3 media blocks.
x md5.1.ps, 8572 bytes, 17 media blocks.
x md5.1.txt, 1503 bytes, 3 media blocks.
x md5.h, 1350 bytes, 3 media blocks.
x md5c.c, 10423 bytes, 21 media blocks.
x mddriver.c, 5354 bytes, 11 media blocks.
x rfc1321.txt, 35223 bytes, 69 media blocks.
# ls
MD5.tar          md5-announcement.txt  md5.h
Makefile         md5.1                 md5c.c
README          md5.1.ps              mddriver.c
global.h        md5.1.txt             rfc1321.txt
# vi Makefile
...
CC = cc
...
# make md5
cc -c -O -DMD=5 md5c.c
cc -c -O -DMD=5 mddriver.c
cc -o md5 md5c.o mddriver.o
#
```

Notice that we modified the Makefile to reflect the fact that we were using `cc` (AIX C compiler) rather than `gcc` (GNU C compiler).

Once the `make` has successfully completed, MD5 is ready for use. For example, to compute the MD5 checksum on the `nmap-2.53.0.0.bff` file:

```
# ./md5 nmap-2.53.0.0.bff
MD5 (nmap-2.53.0.0.bff) = 449244ff27fc11be06864ac6b0e8bb44
```

When downloading files from the Internet, execute the MD5 checksum on the downloaded files, and then verify the MD5 checksums with the MD5 checksums published on the Web site. If the checksums are identical, you know that what you downloaded matches what was put on the site. If they are not, delete the file(s), and notify the site owner. MD5 is also a good tool to use when sending files to a remote location across the network.

MD5 is but one of a number of available checksum tools. It is, however, also one of the most popular and widely-used ones.

7.4.1 Ensuring the integrity of downloads

Before downloading anything from the Internet, you first need to decide if you trust the site. If you do, you next need to ensure that the copy of the file(s) you download match the original version of the file(s) placed on the site by the owner. There are several ways you can do this, depending on what is available from the site.

Some sites will include a digital signature from the author for your verification. For example, the Bull site includes digital signatures for some of its filesets. An example package that contains a digital signature is the `nmap-2.53_0_0.exe` package. Once you inflate this package, you will find an `nmap-2.53_0_0.asc` (digital signature) file along with the installable `nmap-2.53_0_0.bff` file. To verify the signature, you need to download the public key of the author and run the `pgpv` command against the `nmap-2.53_0_0.asc` file. A good signature indicates that the package has not been tampered with. For additional instructions, see the `.asc` file.

```
# ./nmap-2_53_0_0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: nmap-2.53.0.0.bff
  inflating: nmap-2.53.0.0.bff.asc

# pgpv nmap-2.53.0.0.bff.asc -o nmap-2.53.0.0.bff
Cannot open configuration file /.pgp/pgp.cfg
Opening file "/dev/null" type text.
This signature applies to another message
File to check signature against [nmap-2.53.0.0.bff]:
Good signature made 2000-07-24 15:32 GMT by key:
  1024 bits, Key ID 22F02AED, Created 1999-03-15
  "Ciaran Deignan <ciaran.deignan@bull.net>"

WARNING: The signing key is not trusted to belong to:
Ciaran Deignan <ciaran.deignan@bull.net>
```

Some sites may not have all these files packaged together. When that is the case, you will need to download each of the files separately.

Other sites may not use PGP signatures at all. Instead, they publish the MD5 checksums for their downloadable files. Once you have downloaded the file(s), you use MD5 to generate the checksum(s), and then compare your results with those published on the site. If they are different, delete the file(s) and notify the owner of the site.

Finally, some sites just post information about the security measures in place on their site. After reviewing this information, you can decide whether or not you feel comfortable proceeding with the download.

Chapter 8. Securing AIX

Once AIX is installed on a server, there are a number of additional configuration changes that you can make to further tighten down security on the box. This chapter illustrates how to turn off unnecessary services and modify the configuration of the remaining services to make them even more secure. AIX supports a multitude of applications (each with their own unique requirements) in any number of conceivable environments making it impossible to create a single, default security environment that applies to all. Thus, the security level you set up in AIX needs to be tailored to the customer and application environment where the server is deployed. For example, an Internet Web server will most likely have different security requirements than a software development server located in a lab.

A common entry point for attackers is through vulnerabilities in network services. New exploits are discovered almost daily; so, what is secure today may not be secure tomorrow. The best rule of thumb is to turn off all services that are not needed. The best time to do this is before the server is in production. Begin by turning off as much as possible. If in doubt, turn it off. Then, selectively turn services back on only when there is proof that they are needed. It is far easier to turn on what is needed before going into production than it is to try and figure out what can be turned off once the server is live. You can then focus your security efforts toward making configuration changes and applying the latest security patches on the remaining services.

To secure a building, all entry points around the perimeter must be secure. The same holds true for securing a server. Take, for example, a Web server. Among other things, this server consists of the network, the operating system, and the Web server software (application). If you focus only on securing the Web server software piece, you leave the operating system and the network doors vulnerable. An entry gained through any door is an entry gained. For example, suppose the attacker takes advantage of an `rpc.statd` vulnerability to gain root access. At that point, the battle is lost. Not only has the OS been compromised, but everything else along with it. Worse yet, suppose that `rpc.statd` was a service not required for the proper functioning of the server. Unneeded doors should be removed and bricked over, and the remaining doors should be reinforced with the strongest materials at your disposal.

Securing AIX is not a one-time process. Even though you have taken steps to turn off unnecessary services and protect the remaining ones, you still need to maintain constant vigilance. New exploits are being invented almost daily, and changes to the system, such as software upgrades, potentially open new

holes. Apply security fixes to your system as soon as it is practicable, and strive to know the normal behavior of the system so that you can easily spot anomalies if they occur.

8.1 Overview

As mentioned earlier, there are two types of attackers: Those who are on the inside and those who are on the outside. Both can cause harm, but they each have slightly different profiles in the way they attack. For example, internal attackers may already have legitimate user accounts on the systems. In general, they do not, typically, have root access. However, having a normal, non-privileged account still gives them the freedom to snoop around looking for possible configuration errors and system vulnerabilities.

External attackers need to spend more time doing reconnaissance against their potential targets. For example, using tools, such as *nmap* (with the `-O` flag), external attackers attempt to find out which operating system runs on their target. They then use tools, such as *SAINT* or *nmap*, to attempt to determine which network service ports are open. With this information in hand, they next use a tool, such as *netcat* (*nc*), to attempt to grab banner information to determine which version of the service is running on each port.

If the attacker finds that you are running a certain version of a service, and there are known problems with this level of code, the attacker can then use known attack methods to penetrate the system. The importance of keeping your systems up-to-date with the latest security patches and configuring your systems to not leak any unnecessary information cannot be understated. One has simply to look at recent news headlines to understand that the threat is very real and that the reputation of the organization may very well hinge on the security of the systems you have been entrusted to protect.

The general steps to harden AIX (which are explained in detail in the remainder of this chapter) are as follows:

1. Remove unnecessary services.
2. Tighten configurations of remaining services.
3. Set proper network (`no`) options.
4. Tighten up user accounts.
5. Set up a strong password policy.
6. Install additional security tools (such as the ones mentioned in previous chapters).

7. Monitor logs, audit trails, and system behavior.

Note

Prior to making any change to a production system, be certain that you have a known-good backup just in case the change does not work the way you expect.

In the examples that follow, we assume a newly-installed system. Our environment (AIX 4.3.3 on an RS/6000 F50) is meant to simply illustrate what is possible. Your environment is almost certainly different and more complex; so, you will need to take this information and tailor it to your unique needs.

Important

Many organizations have very strict policies in place against running certain types of tools, such as sniffers and scanners, on a production network. Make sure you have the proper written authorization in place prior to running any of these tools on your systems.

8.2 Step 1: Remove unnecessary services

You can see which services have active listening ports on your system with the `netstat -af inet` command. Alternatively, you can do this with scanning tools, such as NSA, SAINT, or nmap. The advantage to using scanning tools are threefold:

- You can run them against a multitude of servers from a single location. With `netstat`, you need to run it on a server-by-server basis.
- Tools, such as NSA (Section 4.4, "Network Security Auditor (NSA)" on page 49) and SAINT (Section 6.3, "Security Administrator's Integrated Network Tool (SAINT)" on page 94), have a list of known vulnerabilities that they use to test against.
- Attackers use similar tools to scan your network. By knowing what they know and seeing what they see, you can devise strategies to defend against such scans and minimize the amount of information that is available to would-be attackers. The less information they have to work with, the harder it is to penetrate your network.

Important

Many organizations have very strict policies in place against running certain types of tools, such as sniffers and scanners, on a production network. Make sure you have the proper written authorization in place prior to running any of these tools on your systems.

Additionally, in rare cases, scanning tools, such as NSA, can cause system crashes. Ensure that you have current known-good backups, that you have fully tested the scanning tool on a non-production server and network, and that you have informed the relevant parties before launching a scan on the production network.

It is also possible to permanently remove a service by uninstalling the appropriate filesets. This is not required and should only be undertaken by someone with sufficient AIX skills since there is a high risk involved with removing system-level filesets. We recommend disabling services, rather than removing them.

8.2.1 Removing entries from `/etc/inittab`

Important

Prior to making any changes to `/etc/inittab`, be sure to make a backup copy of it. It is also a good idea to have a known-good system backup.

The `/etc/inittab` file varies from system to system depending on the applications that are installed on the server. Verify that the entries in `/etc/inittab` reflect the software that is installed on the server. Applications, such as TSM, Oracle, HACMP, and so on, typically have entries in `/etc/inittab`. If the software is no longer used on the server, remove the `/etc/inittab` entries.

The following default entries in `/etc/inittab` are candidates for removal since they are usually not required:

```

piobe:2:wait:/usr/lib/lpd/pio/etc/pioint >/dev/null 2>&1 # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
writesrv:2:wait:/usr/bin/startsrc -swritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd
httpdlite:2:once:/usr/IMNSearch/httpdlite/httpdlite -r /etc/IMNSearch/httpdlite/
httpdlite.conf & >/dev/console 2>&1
dt:2:wait:/etc/rc.dt
immss:2:once:/usr/IMNSearch/bin/immss -start immhelp >/dev/console 2>&1
imgss:2:once:/usr/IMNSearch/bin/img_start >/dev/console 2>&1

```

If you decide to remove these entries, you can do so with the `rmitab` command as follows:

```

# for i in piobe qdaemon writesrv uprintfd httpdlite dt immss imgss \
>do \
>rmitab $i \
>done

```

Removing the entries from `/etc/inittab` will disable them from automatically starting. Reboot the server after the modification to test the change and ensure that everything else still properly starts. You may also want to alter the permissions (`chmod 0000 <filename>`) on the executables to further disable the service. You can even go so far as to uninstall the filesets (particularly the Dt filesets used for CDE), but great care must be taken when doing this.

The following sections provide detailed descriptions of the default `/etc/inittab` entries. This information is provided to help you determine whether or not these entries are needed on your systems.

8.2.1.1 The piobe and qdaemon entries

The `qdaemon` program is used to schedule jobs for printing. The `piobe` program is the spooler backend program called by the `qdaemon` program. It processes the print job and serves as the print job manager. These two entries are required if the server is set up for printing.

8.2.1.2 The httpdlite, immss, and imgss entries

The `httpdlite` program is the default Web server for the docsearch engine and is installed by default. For docsearch to function properly, a Web server is required, and `httpdlite` is the version that comes with the `IMNSearch.rte.httpdlite` fileset. If you have another Web server, you can remove `httpdlite`, and configure docsearch to use your installed Web server. For more information on how to do this, see:

www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/topnav/topnav.htm

In most cases, the man pages are sufficient, and docsearch is not really required. Storing local copies of man pages is rarely an issue with the disk sizes today, and the man pages provide much faster access to information than does the docsearch engine. The one advantage that docsearch provides over the man pages is its search capabilities.

8.2.1.3 The dt entry

The `/etc/rc.dt` script is used to start Common Desktop Environment (CDE). Unless absolutely essential, CDE should not be run on servers because of the security issues inherent in it as well as the X window system on which it sits. Remove the `/etc/rc.dt` entry, and use a tool, such as Tripwire, to ensure that it is not put back in. Uninstalling CDE (Dt) filesets is another option.

8.2.1.4 The writesrv entry

The `writesrv` daemon enables users to communicate back and forth between systems by means of the `write` command. In order for this communication to work, the receiving system(s) must have the `writesrv` daemon running.

The `writesrv` daemon receives incoming requests from a `write` command and creates a server process to handle the request. This server process communicates with the client process (`write`) and provides whatever services are requested. Removing the `writesrv` entry prevents use of the `write` command between systems but does not effect its use between users on the local system.

8.2.1.5 The uprintfd entry

The `uprintfd` daemon retrieves, converts, formats, and writes kernel messages to the controlling terminals of processes. Kernel messages are submitted through the `NLuprintf` and `uprintf` kernel services. This daemon is generally not required.

8.2.2 Removing entries from `/etc/rc.tcpip`

The `/etc/rc.tcpip` script starts the TCP/IP daemons. By default, the `syslog`, `sendmail`, `portmap`, `inetd`, `snmpd`, and `dpid2` daemons are started. You may not need some of these (especially `dpid2`). For the ones that you do need, be sure to modify their configuration files to further tighten down security, and always keep them up-to-date with the latest patch levels.

To determine which TCP/IP services are actively listening on the system, use the `netstat -af inet` command. If you find a service listening on an unknown port, you can use tools, such as `lsof` (see Section 6.5, "List Open Files (`lsof`)" on page 103), to help determine the ownership.

To prevent a service from starting in `/etc/rc.tcpip`, either comment it out (add a `#` at the beginning of the line) or delete the entry from the file. We suggest the latter because it is all too easy for another administrator to uncomment the line during a problem determination session and unknowingly open a potential and unnecessary security vulnerability.

To stop the unwanted TCP/IP daemons, you can either wait for a maintenance reboot of the system, or you can run the `/etc/tcp.clean` script to stop all TCP/IP daemons, then run `/etc/rc.tcpip` to restart only those that are still uncommented in the file. Note that all TCP/IP connections will be severed when you run `/etc/tcp.clean`. This is very disruptive and should not be done on a production server outside of a maintenance window. In addition, ensure that you are locally connected to the system because all telnet connections will be dropped as well.

In most cases, the following daemons do not need to be started:

```

# Start up dhcpd daemon
#start /usr/sbin/dhcpd "$src_running"

# Start up autoconf6 process
#start /usr/sbin/autoconf6 ""

# Start up ndpd-host daemon
#start /usr/sbin/ndpd-host "$src_running"

# Start up the ndpd-router daemon
#start /usr/sbin/ndpd-router "$src_running"

# Start up print daemon
#start /usr/sbin/lpd "$src_running"

# Start up routing daemon (only start ONE)
#start /usr/sbin/routed "$src_running" -q
#start /usr/sbin/gated "$src_running"

# Start up Domain Name daemon
#start /usr/sbin/named "$src_running"

# Start up time daemon
#start /usr/sbin/timed "$src_running"

# Start up Network Time Protocol (NTP) daemon
#start /usr/sbin/xntpd "$src_running"

# Start up rwhod daemon (a time waster)
#start /usr/sbin/rwhod "$src_running"

# Start up the Simple Network Management Protocol (SNMP) daemon
# start /usr/sbin/snmpd "$src_running"

# Start up the DHCP Server
#start /usr/sbin/dhcpd "$src_running"

# Start up the DHCP Relay Agent
#start /usr/sbin/dhcprd "$src_running"

# Start up the DPID2 daemon
start /usr/sbin/dpid2 "$src_running"

# Start up the mouted daemon
#start /usr/sbin/mouted "$src_running"

```

The following sections provide detailed descriptions of these entries. This information is provided to help you determine whether or not these entries are needed on your systems.

8.2.2.1 The dhcpd, dhcpsd, and dhcprd entries

DHCP (Dynamic Host Control Protocol) is a protocol that enables network clients to receive their IP addresses from a server pool of IP addresses. DHCP is an extension of the bootp protocol. When a client machine is

configured to use DHCP boots, it sends out a request for an IP address to a DHCP server. The DHCP server responds with an IP address assigned from its pool of addresses. The client then uses this assigned IP address to configure the remainder of its network services and finish booting.

The entries correspond to the client, server, and relay components of the protocol. If you do not use DHCP with your AIX servers, leave these entries commented.

8.2.2.2 The autoconf6, ndpd-host, and ndnpd-router entries

The autoconf6 process is run at boot time to configure the IPv6 network interfaces and to add necessary routes. The ndp entries are also for IPv6 and are not yet commonly used.

If you are not using IPv6, leave these entries commented.

8.2.2.3 The lpd entry

The lpd daemon is required if this machine acts as a print server to other machines. In other words, this machine has an attached printer that is used by remote machines for printing.

8.2.2.4 The routed and gated entries

Routed and gated are two daemons used to implement dynamic routing. You can run either routed or gated but not both. Generally, in a secure environment, you want to keep things deterministic. So, instead of using dynamic routing, you define static routes to the system. When this is the case, leave the entries commented, and be sure to set the network (no) options accordingly. For more information on the no options, see Section 8.4, "Step 3: Set proper network (no) options" on page 186.

8.2.2.5 The named entry

The named daemon is used when the local server acts as a Domain Name System (DNS) server providing name resolution services to other machines on the network. For more information on DNS, see Section 8.3.1, "Domain Name System (DNS)" on page 153.

8.2.2.6 The timed and xntpd entries

Timed and xntpd are two different time synchronization daemons. They are used by servers to synchronize their time with a reliable clock source. These daemons are typically used in time-sensitive environments, such as DCE/DFS cells, SP complexes (NTP is used by default), and some firewalls (to synchronize connection tables).

8.2.2.7 The rwhod entry

This daemon is used to provide remote users with information about who is logged in to the server. It is a very big security exposure, and should not be used.

8.2.2.8 The snmpd entry

SNMP is used for network administration. It has security vulnerabilities and should be removed if not needed. If it is needed, at a minimum, you should change the community names to unique, hard-to-guess names. For more information on configuring SNMP, see Section 8.3.4, "Simple Network Management Protocol (SNMP)" on page 180.

8.2.2.9 The dpid2 entry

DPID is a legacy SNMP daemon. It is rarely used, is a security exposure, and should be removed. Delete the entry from `/etc/rc.tcpip`, and modify the permissions of the executable (`chmod 0000 /usr/sbin/dpid2`).

8.2.2.10 The mroued entry

Mroued is used to forward multicast datagrams. If you are not using multicasting, leave it commented.

8.2.3 Removing entries from `/etc/inetd.conf`

The `inetd` daemon acts as a master server that invokes other daemons (specified in the `/etc/inetd.conf` configuration file) as needed. This design is meant to reduce system load. Each new daemon started by `inetd` results in another active port being available for remote client connection. This also means additional opportunities for attack.

Unneeded services should be commented out. In some cases, you may also be able to remove services that are normally considered necessary. For example, you can replace `telnet`, `rsh`, `rlogin`, and `ftp` with SSH equivalents (`ssh`, `scp`, `sftp`), which are more secure. (For more information about SSH, see Section 5.1, "Secure Shell (ssh)" on page 59.) If you decide to go with SSH, comment out the `telnet`, `rsh`, and `ftp` services in `/etc/inetd.conf`. You might also consider replacing the `ftp`, `ftpd`, `telnet`, `telnetd`, `rsh`, `rshd`, `rlogin`, and `rlogind` executables with their SSH equivalents. The only drawback to this approach is that they will almost certainly be overwritten by OS upgrades and patches requiring you to redo the replacements each time.

For those services that are required, consider protecting them with TCP Wrapper. (For detailed information about how to set up TCP Wrapper, see Section 5.2, "TCP Wrapper" on page 77.) TCP Wrapper protects these services through access control and logging. Unfortunately, there is still a

problem with many of these services exchanging authentication information in clear text across the network. TCP Wrapper does nothing to solve this; so, again, only enable those services that are absolutely required.

In an SP environment, kshell, klogin, bootps, and tftp are required. However, you can selectively turn them off when they are not required and reenable them when they are.

For our basic setup, we removed the following services from the `/etc/inetd.conf` file. This is not meant to be a definitive list. You need to decide which services to run and which to remove based on your environment. A lot of these services are nice to have but may present a security problem.

To prevent inetd from starting a service, comment it out in the `/etc/inetd.conf` configuration file, and then refresh inetd (`refresh -s inetd`).

```

#shell stream tcp6 nowait root /usr/sbin/rshd rshd
#kshell stream tcp nowait root /usr/sbin/krshd krshd
#login stream tcp6 nowait root /usr/sbin/rlogind rlogind
#klogin stream tcp nowait root /usr/sbin/krlogind krlogind
#exec stream tcp6 nowait root /usr/sbin/rexecd rexecd
#comsat dgram udp wait root /usr/sbin/comsat comsat
#uucp stream tcp nowait root /usr/sbin/uucpd uucpd
#bootps dgram udp wait root /usr/sbin/bootpd bootpd /etc/bootp
tab
#finger stream tcp nowait nobody /usr/sbin/fingerd fingerd
#sysstat stream tcp nowait nobody /usr/bin/ps ps -ef
#netstat stream tcp nowait nobody /usr/bin/netstat netstat -f inet
#
#tftp dgram udp6 SRC nobody /usr/sbin/tftpd tftpd -n
#talk dgram udp wait root /usr/sbin/talkd talkd
#ntalk dgram udp wait root /usr/sbin/talkd talkd
#rquotad sunrpc_udp udp wait root /usr/sbin/rpc.rquotad rquota00011 1
#rexcd sunrpc_tcp tcp wait root /usr/sbin/rpc.rexcd rexd 100017 1
#rstatd sunrpc_udp udp wait root /usr/sbin/rpc.rstatd rstatd 1000
01 1-3
#usersd sunrpc_udp udp wait root /usr/lib/netsvc/users/rpc.ruser
sd usersd 100002 1-2
#rwalld sunrpc_udp udp wait root /usr/lib/netsvc/rwall/rpc.rwalld
rwalld 100008 1
#sprayd sunrpc_udp udp wait root /usr/lib/netsvc/spray/rpc.sprayd
sprayd 100012 1
#pcnfsd sunrpc_udp udp wait root /usr/sbin/rpc.pcnfsd pcnfsd 1500
01 1-2
#echo stream tcp nowait root internal
#discard stream tcp nowait root internal
#chargen stream tcp nowait root internal
#daytime stream tcp nowait root internal
#time stream tcp nowait root internal
#echo dgram udp wait root internal
#discard dgram udp wait root internal
#chargen dgram udp wait root internal
#daytime dgram udp wait root internal
#time dgram udp wait root internal
#instsrv stream tcp nowait netinst /u/netinst/bin/instsrv instsrv -r /tmp/n
etinstalllog /u/netinst/scripts
#ttbserver sunrpc_tcp tcp wait root /usr/dt/bin/rpc.ttbserver
er rpc.ttbserver 100083 1
#dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd
#cmsd sunrpc_udp udp wait root /usr/dt/bin/rpc.cmsd cmsd 100068
2-5
#imap2 stream tcp nowait root /usr/sbin/imapd imapd
#pop3 stream tcp nowait root /usr/sbin/pop3d pop3d
#ssalld sunrpc_tcp tcp wait root /usr/sbin/rpc.ssalld rpc.ssalld
300667 1
#kfcli stream tcp nowait root /usr/lpp/ssp/install/bin/kfserver kfserver
#xmquery dgram udp wait root /usr/bin/xmservd xmservd -p3

```

The following sections provide detailed descriptions of these entries. This information is provided to help you determine whether or not these entries are needed on your systems.

8.2.3.1 The shell (rshd) and login (rlogind) entries

These entries are for rsh, rlogin, and rcp command functionality and should be disabled for security reasons. See Section 5.1, "Secure Shell (ssh)" on page 59, for more information. The rshd daemon uses TCP port 514, and the rlogind daemon uses TCP port 513.

8.2.3.2 The kshell (krshd) and klogin (krlogind) entries

These entries are similar to the shell and login services but are used when Kerberos authentication is used. They are typically required in SP environments. It is important to note that in an SP environment, even though the sessions are authenticated, data from commands, such as rsh and rcp, are transferred in the open without encryption. The krshd daemon uses TCP port 544, and the krlogind daemon uses TCP port 543.

8.2.3.3 The exec, rstatd, ruserd, rwalld, and rquotad entries

The exec (rexecd) entry is for rexec command functionality, which is used for remote command execution support. As with other "r" services, it should be disabled. The rexecd daemon uses TCP port 512. The other daemons listed here use rpc (portmap) for their port allocation.

8.2.3.4 The comsat entry

The comsat daemon is the server that receives reports of incoming mail and notifies those users who have enabled this service (with the `biff` command) accordingly. The comsat daemon uses UDP port 512.

8.2.3.5 The uucp (uucpd) entry

The `uucp` command is a Basic Networking Utilities (BNU) command. It is used to copy one or more source files from one system to one or more destination files on another UNIX system. The uucpd daemon uses TCP port 540.

8.2.3.6 The bootps (bootpd) and tftp (tftpd) entries

The bootpd daemon implements an Internet Boot Protocol server to enable remote boot of clients. The `tftp` command uses trivial file transfer protocol to transfer files from one system to another. These services are required in an SP when installing or network-booting an SP node. The tftp service, in particular, has a number of security issues that you need to be aware of if you plan to leave it enabled. For more information, see Section 8.3.7, "File Transfer Protocol (ftp)" on page 184. The bootpd daemon uses UDP port 67, and the tftpd daemon uses UDP port 69.

8.2.3.7 The finger (fingerd) entry

The `finger` command displays information about the users currently logged in to a system (either local or remote). It should be disabled, as it is commonly

used for social-engineering attacks and buffer-overflow attacks. The fingerd daemon uses TCP port 79.

8.2.3.8 The systat (ps) entry

The systat entry provides support for the `ps` command. It uses TCP port 11.

8.2.3.9 The netstat entry

The netstat entry provides support for the `netstat` command. It uses TCP port 15.

8.2.3.10 The talk (talkd) and ntalk (talkd) entries

The `talk` command allows two users on the same system or on different systems to have an interactive conversation. The talkd daemon uses TCP port 518.

8.2.3.11 The rexd entry

With the rexd daemon running, any program emulating the `on` command can obtain shell access, which, in turn, provides access to files on the remote system including things, such as the password file. Because of the security risks, it should be disabled. The rexd daemon uses rpc (portmap) for its port allocation.

8.2.3.12 The sprayd entry

The sprayd daemon provides support for the `spray` command. Because the `spray` command can be used for both buffer overflow and denial of service attacks, this entry should be disabled. The sprayd daemon uses rpc (portmap) for its port allocation.

8.2.3.13 The pcnfsd entry

The pcnfsd daemon provides NFS service to PC clients. It has a history of attackers using untrapped-meta-character vulnerabilities to execute unauthorized commands on remote systems. For this reason, it should be disabled. The pcnfsd daemon uses rpc (portmap) for its port allocation.

8.2.3.14 The echo, chargen, time, and daytime entries

All these services can be spoofed into sending data from one service on one system to another service on another system resulting in an infinite loop and creating a denial of service attack. The port usage is as follows: echo uses TCP and UDP port 7, chargen uses TCP and UDP port 19, time uses TCP and UDP port 37, and daytime uses TCP and UDP port 13.

8.2.3.15 The discard entry

As its name implies, discard discards whatever is sent to it by a client. In all likelihood, you do not need it. It uses TCP and UDP port 9.

8.2.3.16 The ttdbserver, dtspc, and cmsd entries

These entries are used for the CDE environment. CDE should not be used on a server unless absolutely required. If you are disabling CDE, disable these entries also. All of these services use rpc (portmap) for their port allocation.

8.2.3.17 The imap2 (imapd) and pop3 (pop3d) entries

Both of these daemons are used to retrieve mail. Each has had buffer overflow problems in the past. If this system is not serving as an imap or pop3 server, disable these entries. The imapd daemon uses TCP and UDP port 143 for imap2, and the pop3d daemon uses TCP port 110.

8.3 Step 2: Tighten configurations of remaining services

No doubt, some services will still be required to support the normal production activities of the server. These services typically include things, such as DNS, NFS, SMTP, and SNMP. For any remaining active services, you should adjust the default configurations to further tighten their security. The extent of the configuration changes will depend on the security level that you are trying to achieve.

8.3.1 Domain Name System (DNS)

DNS is used to map a host name to an IP address and vice-versa. Basically, it is a host name to IP lookup protocol. Other means of host name resolution include the local host table (`/etc/hosts`) and NIS. The local host table solution does not scale well, and NIS has a number of well-known security issues. For these reasons, DNS is the most commonly used solution.

On AIX, DNS is managed by BIND (the Berkeley Internet Name Daemon). DNS and BIND are so closely related that the terms are interchangeable.

Two different BIND versions (BIND4 and BIND8) are in common use. BIND4 is the older protocol, and support for it has been dropped. BIND8 is the newer protocol, which includes more capabilities, built-in security, and support for bug fixes.

The BIND homepage is:

<http://www.isc.org/products/BIND/>

8.3.1.1 DNS principles

When asked to resolve a host name to IP address (for example, `sales.west.france.acme.com.`), the DNS name server tries to answer the following three questions:

1. Is the answer in the my local DNS cache?
2. If not, does this name belong to my DNS domain?
3. If not, to which other DNS name server should I escalate this request?

Escalating the request and receiving the answer can be quite time consuming. Assuming the worst case scenario, let us say that your DNS name server belongs to a completely different DNS domain (for example, `abc.xyz.fr.`). The following domain name servers will be queried in sequence (assuming no cache hits along the way):

1. . (“dot”, this is the top of the DNS tree)
2. `com.`
3. `acme.com.`
4. `france.acme.com.`
5. `west.france.acme.com.`

To reduce both network load and delays, each DNS name server manages a local cache of the answers to its most recent queries. Once your name server finally gets the IP address for `west.france.acme.com.`, it will not have to make the preceding queries again until the information in its cache expires.

The use of DNS cache typically speeds up name resolution by anywhere from one to three orders of magnitude. However, with the volatile nature of the Internet, the question of how to keep DNS cache up-to-date with the thousands of address changes that occur every day remains. (If the address associated with a host name changes but the local DNS cache still has the old information, host name resolution will be hindered by the cache rather than helped.) We will return to the question of DNS cache consistency shortly.

8.3.1.2 DNS spoofing

According to www.dictionary.com, spoofing means: Nonsense; a hoax; a gentle satirical imitation; a light parody.

A client knows its DNS name server by IP address only. This IP address can be faked by an attacking machine on the LAN (for instance, a portable computer). If that false name server answers faster than the real name server,

it can misdirect the querying client to other hostile computers. However, this technique has the following drawbacks from the point of view of the attacker:

- It requires a machine to be connected to the LAN.
- Success is not guaranteed. Even if it is faster, the false name server will only fool the clients that are querying DNS as they boot. The other clients on the LAN already have the MAC address for the real DNS name server in their ARP cache.
- A single IP address associated with two different MAC addresses is easy to detect by system and network administrators because it generally causes additional problems on the network.

For these reasons, attackers designed a new and improved version of DNS spoofing.

8.3.1.3 The new and improved DNS spoofing

We saw in Section 8.3.1.1, "DNS principles" on page 154, that cached addresses need to be updated whenever they change. The whole idea behind DNS spoofing is to send counterfeit update requests to a DNS name server. By definition, the DNS name server sending the update requests does not belong to the same DNS domain as the local DNS name server. In other words, the remote DNS name server is not a trusted host, and, in principle, the update should not be allowed. Prior to BIND 8.2, there was no way to prevent this.

How the DNS spoofing exploit works

Here is an example of how the DNS spoofing exploit works:

The attacker starts by sending a bogus update to a DNS name server. The update changes the IP address of a fully-qualified host name in the DNS name server cache. Since it appears as nothing more than a routine update of the IP address associated with a given host name, everything seems fine, and since it is an update to a fully-qualified host name, there is no need to disturb anything further up the DNS hierarchy.

Now, imagine that the IP address that was updated was the IP address to a Web site that provides secure content. Suppose further that the attackers have managed to duplicate the home page of that site on the host whose IP address they substituted for the legitimate one. To you, it looks and feels just like the real site, and you unknowingly attempt to log in with your account number and password.

The attackers now employ a technique known as *blackholing*. Blackholing means intentionally giving no answer to a request. This technique is intended to consume time from the requester and is used both by attackers against their victims and by system administrators against suspected attackers. So, the attackers blackhole you, then put up some fake message (such as “site busy” or “requested page not found”) and ask you to try again later. While you are waiting for all of this to happen, they turn around and use your identity and password to access the legitimate site. By the time you realize what has happened, they will be long gone. Worse yet, because they accessed the site with your legitimate account number and password, the site will probably not accept any responsibility for any damage done to you.

Protection against the DNS spoofing exploit

One form of protection against DNS spoofing involves a software modification to do an extra DNS lookup for each request. After translating the incoming request from an IP address to a host name, a second translation from host name back to IP address is performed. If the two IP addresses do not match, the request is denied and a warning message issued. (Faking an address in a direct access table is easy, but inserting a fake address at the correct location in a sequential, reverse access table is much more difficult.)

To address the growing problem of DNS (and IP) spoofing, RFC 2065 defines the DNS Security Extensions (DNSSEC). These security extensions enable DNS resolvers and DNS name servers to cryptographically authenticate the source and, through digital signatures, guarantee the integrity of the DNS queries and responses. They are incorporated into BIND 8.2.1, which is included in AIX 4.3.3.

DNS spoofing compared to IP spoofing

To illustrate the difference between DNS spoofing and IP spoofing, we use a telephone system analogy:

With IP spoofing, a skilled technician takes over the phone number (IP address) of the destination. From then on, calls (requests) made to that number (IP address) go to the technician (bogus server), rather than to the legitimate destination (real server).

DNS spoofing is more subtle. With DNS spoofing, the skilled technician keeps their original phone number (IP address) unchanged. Instead, the technician modifies a phone book (DNS server) somewhere in the system and substitutes their phone number (IP address) in place of a legitimate one. Then, when someone looks up the name for that entry, they get the phone

number (IP address) of the technician, rather than that of the legitimate owner.

DNS spoofing requires IP spoofing at the very beginning to fake the identity of a DNS server. Once that is done, however, the attacker can revert back to their “legitimate” IP address (which, in all likelihood, is probably the address of another site the attacker has broken into). At this point, nothing appears out of the ordinary on the network. The security extensions provided in BIND 8.2.1 are meant to thwart these kinds of attacks.

8.3.1.4 DNS on AIX

The DNS daemon, `named`, is started by default in `/etc/rc.tcpip`. The version of BIND is dependent on the version of AIX as shown in Table 4:

Table 4. AIX versions of BIND

| AIX Version | BIND Version |
|---|--|
| AIX 3.2.x | BIND 4.8.3 |
| AIX 4.1.x | BIND 4.9.3 |
| AIX 4.2.x | BIND 4.9.3 |
| AIX 4.3.x | ipv4 BIND 4.9.3 ipv6 BIND 8.1.2 |
| AIX 4.3.3 (with <code>bos.net.tcp.server 4.3.3.13</code>) | ipv4 BIND 4.9.3 ipv6 BIND 8.2.2 (patch 5) |

To check which version of BIND (BIND4 or BIND8) is running on the system, look at how the `/usr/sbin/named` file is symbolically linked (either to `/usr/sbin/named4` or `/usr/sbin/named8`):

```
$ ls -l /usr/sbin/named
lrwxrwxrwx  1 root  system      16 Mar 08 1999 /usr/sbin/named ->
/usr/sbin/named4
$ ls -l /usr/sbin/named-xfer
lrwxrwxrwx  1 root  system      21 Mar 08 1999
/usr/sbin/named-xfer -> /usr/sbin/named4-xfer
```

The `named` daemon is contained in the `bos.net.tcp.client` fileset, as shown in the next screen.

```

$ type named4
named4 is /usr/sbin/named4

$ lsllp -w /usr/sbin/named4
File                               Fileset                               Type
-----
/usr/sbin/named4                   bos.net.tcp.server                   File

$ lsllp -f bos.net.tcp.server|more
Fileset                               File
-----
Path: /usr/lib/objrepos
     bos.net.tcp.server 4.3.3.0
                             /usr/sbin/trpt
                             /usr/sbin/named8-xfer
                             /usr/samples/srmpd/view.my
                             /usr/sbin/rsvpd
                             /usr/sbin/XNSquery
                             /usr/samples/srmpd/unix.my
                             /usr/sbin/mosy
                             /usr/samples/tcpip/named.hosts
                             /usr/sbin/named -> /usr/sbin/named4
                             ...

```

Converting from BIND4 to BIND8 is fairly straightforward. First, stop named. Then, convert the BIND4 configuration file (`/etc/named.boot`) into the BIND8 configuration file (`/etc/named.conf`). The easiest way to do this is with the `/usr/samples/tcpip/named-booconf.pl` PERL script. Next, relink the `named`, `named-xfer`, and `nsupdate` files. Finally, restart named:

```

# stopsrc -s named
0513-044 The named Subsystem was requested to stop.
# cd /usr/samples/tcpip
# chmod u+x named-bootconf.pl
# ./named-bootconf.pl /etc/named.boot > /etc/named.conf
#rm /usr/sbin/named
#rm /usr/sbin/named-xfer
#rm /usr/sbin/nsupdate
#cd /usr/sbin
#ln -s named8 named
#ln -s named8-xfer named-xfer
#ln -s nsupdate8 nsupdate
#startsrc -s named
0513-059 The named Subsystem has been started. Subsystem PID is 12346.

```

If you are setting up BIND8 on a system that previously just used `/etc/hosts` for name resolution, you can use the `h2n` PERL script to convert the `/etc/hosts` file into the appropriate named files. The `h2n` PERL script is available via anonymous FTP from:

`ftp://ftp.uu.net/published/oreilly/nutshell/dnsbind/dns.tar.Z`

8.3.1.5 Securing DNS on AIX

The following configuration changes will help make DNS more secure on AIX:

Restrict zone transfers

Restricting zone transfers enables you to control which servers (trusted servers) may request and receive zone transfers from your DNS server. The information provided by a DNS zone transfer is of great value to a would-be attacker. It provides information about things, such as mail servers, Web servers, server aliases, and network mapping. This option is available with BIND 4.9 and later.

Any user on a remote system that is able to query the zone data of your DNS name server is only able to look up data (such as IP addresses) for hosts whose domain names they already know, and, then, only one at a time. Any user on a remote system that is able to request and receive zone transfers from your DNS name server is able to see all of the DNS data for all of the hosts in your zone(s). More information about querying DNS name servers with the `nslookup` tool can be found in Section 8.3.1.6, "Nslookup" on page 165.

Note

The `ls` subcommand of `nslookup` is implemented as a zone transfer.

The `allow-transfer` substatement in BIND 8 and the `xfrnets` directive in BIND 4.9 enables you to apply access lists to zone transfers. The `allow-transfer` substatement can be used as a zone substatement to restrict transfers of a particular zone, or it can be used as an options substatement to restrict all zone transfers. It takes as its argument a list of addresses to match against.

Important

Be sure to restrict zone transfer access on both your primary and slave DNS server.

For example, suppose the DNS slave servers for your `acmebw.com` zone have IP addresses `192.168.0.1` and `192.168.1.1`. The following zone statement (in `/etc/named.conf`) allow only those slave servers to transfer `acmebw.com` from the primary master name server:

```
zone "acmebw.com" {
    type master;
    file "db.acmebw";
    allow-transfer { 192.168.0.1; 192.168.1.1; };
};
```

Note that since the default in BIND 8 is to allow any IP address (including ones belonging to attackers) to transfer zones, you should also have a zone statement, such as the following, on your DNS slave servers:

```
zone "acmebw.com" {
    type slave;
    masters { 192.168.0.4; };
    allow-transfer { none; };
};
```

With BIND 8, you can also establish a global access list for zone transfers. This list applies only to those zones that do not already have explicit access lists defined (via zone substatements).

For example, suppose you want to limit all zone transfers to hosts on your internal network (192.168.0.0):

```
options {
    allow-transfer { 192.168/16; };
};
```

With BIND 4.9, the `xfrnets` directive may have arguments that are the networks or IP addresses that are allowed to do zone transfers from your primary master DNS name server. Networks are specified in dotted-octet format (in `/etc/named.boot`), for example, allowing only hosts on the Class A network 15.0.0.0 and the Class B network 128.32.0.0 to do zone transfers:

```
xfrnets 15.0.0.0 128.32.0.0
```

Unlike secure zones, this restriction applies to any zones for which the server is authoritative.

If you want to specify just a part of the network down to a single IP address, you can add a network mask. The syntax for including a network mask is

network&netmask. Note that no spaces are allowed either between the network and the ampersand or between the ampersand and the netmask.

Suppose you want to pare down the addresses allowed to transfer zones in the previous example to just the IP address 15.255.152.4 and the subnet 128.32.1.0:

```
xfrnets 15.255.152.4&255.255.255.255 128.32.1.0&255.255.255.0
```

For a primary master name server accessible from the Internet, you will probably want to limit zone transfers to just your slave name servers. For a primary master name server on your internal network (behind a firewall), you will probably only want to limit zone transfers to your slave name servers if you are concerned about your own people listing your zone data.

Restrict dynamic updates

Dynamic DNS is a protocol extension to DNS. It defines a lightweight mechanism that allows for zone data modifications (low overhead) without the need for a complete reloading of zone data. The update can theoretically originate from a remote host.

Only BIND 8 has the dynamic update capability. By default, it does not allow dynamic updates. To enable dynamic updates, you must add the allow-update substatement to the name server configuration file (`/etc/named.conf`):

```
zone "acmebw.com" {  
    type master;  
    file "db.acmebw"  
    allow-update { localhost; };  
};
```

If authentication is not being used, dynamic updates should either not be enabled or enabled only to localhost. In some cases, dynamic updates are necessary, for example, with DHCP servers.

Restrict recursion

Recursion is where DNS servers contact other DNS servers on your behalf and relay the final answer back to you. This is different from giving a referral, where one DNS server refers you to the next DNS server and so on. The advantage to recursion is that the servers along the way cache the answers, potentially speeding up the response time for future queries.

With recursion disabled, DNS reverts to referrals. When queried about an unknown host, the DNS server attempts to answer the query. If it is not able to answer it, the DNS server sends back a referral to a more authoritative DNS server. The DNS client then contacts this next server in the chain, and the process repeats. The disadvantage is that this may slow down response time, but the advantage is that it makes it much more difficult to spoof.

Accepting recursive queries from the Internet makes you DNS name server susceptible to DNS spoofing. Suppose an attacker has gained (temporary) control over some remote DNS zones. The attacker then queries your name server for information about those zones. With recursion on, your name server queries the information on your behalf from the commandeered name server. The response from the query contains bogus data that is then stored in the cache of your name server, and the DNS spoof is complete. The attacker can now relinquish control of the commandeered name server since your name server is now set up to route requests to their servers instead of the legitimate ones.

If it is not practical for you to turn recursion off, you should:

- Restrict the addresses to which your name server responds
- Restrict the addresses to which your name server responds to recursive queries

The method for turning off recursion is version-dependent.

For BIND 4.9:

```
options no-recursion
```

For BIND 8:

```
options {  
    recursion no;  
};
```

Restricting queries

The purpose of a DNS name server is to answer questions about the zones for which it is authoritative. For an internal DNS server (one sitting on your internal network behind the firewall), it is a good idea to configure your name server to only respond to queries originating from within your internal network. In other words, your DNS name server should be configured to

respond to only those queries originating from within the same zones for which it is authoritative. In BIND 8, this is done with the `allow-query` substatement inside the zone statement.

For example, suppose `acmebw.com` is on the 192.168.0 network. To restrict queries to your name server from only those hosts on the 192.168.0 network:

```
zone "acmebw.com" {
    type slave;
    allow-query { 192.168.0/24; };
};
```

Note

This option can also be set at a global level rather than at the zone level.

In BIND 4, restricting queries is accomplished with the `secure_zone` record. It limits access to both individual resource records and zone transfers. (In BIND 8, the two are done separately.) With BIND 4, this option only works for those zones for which the DNS name server is authoritative.

To use secure zones, include one or more special TXT (txt) records in the zone data on the primary master name server. The records are automatically transferred to the slave servers:

```
secure_zone      IN      TXT      "192.249.249.0:255.255.255.0"
```

Restrict DNS cache update

The DNS feature that allows for address updates within cache is known as *glue fetching*. It is an open door for DNS spoofing and should be turned off (`fetch-glue no`). This slows the performance of your DNS server (since no cache is built), but it helps you avoid many potential problems, such as spoofing.

The method for turning off glue fetching is version-dependent.

For BIND 4.9:

```
options no-fetch-glue
```

For BIND 8:

```
options {
    fetch-glue no;
};
```

Split DNS configuration

Consider having two kinds of name servers:

- *Advertising name servers* that are open to the world and non-recursive.
- *Internal resolution name servers* that can only be queried by known resolvers (internal and secure). If desired, glue fetching and recursive queries can also be allowed on these servers.

Configuration examples for each type are as follows:

Advertising name server

```
acl slaves { 207.69.231.3; 209.86.147.1; };

options {
    directory /var/named;
    recursion no;
    fetch-glue no;
    allow-query { any; }; // the default
};

zone "acmebw.com" {
    type master;
    file "db.acmebw.com";
    allow-transfer { slaves; };
};
```

Internal resolution name server


```

acl internal { 192.168.0/24; };

options {
    directory "/var/named";
    recursion yes; // the default
    allow-query { internal; };
};

zone "." {
    type hint;
    file "db.cache";
};

zone "acmebw.com" {
    type slave;
    masters { 207.69.231.2; };
    file "bak.acmebw.com";
    allow-transfer { internal; };
};

```

8.3.1.6 Nslookup

The nslookup tool is used to query DNS name servers for information about hosts or domains. On AIX, it is packaged in the bos.net.tcp.client fileset. Nslookup can be used two ways: Interactively and non-interactively (batch). It is commonly used to query a DNS server to obtain the IP address of a server, but it has far more functionality than that.

As a simple example, suppose you want to find the IP address of `www.redbooks.com`:

```

#nslookup www.redbooks.com
Server:  joyce.sg.ibm.com
Address:  9.184.8.1

Non-authoritative answer:
Name:    www.redbooks.com
Address: 198.245.191.153

```

Conversely, suppose you want to find the name associated with IP address `198.245.191.153`:

```
#nslookup -type=ptr 198.245.191.153
Server:  joyce.sg.ibm.com
Address:  9.184.8.1

Non-authoritative answer:
153.191.245.198.in-addr.arpa  name = redbooks.com

Authoritative answers can be found from:
191.245.198.in-addr.arpa      nameserver = nis.ans.net
191.245.198.in-addr.arpa      nameserver = ns.ans.net
nis.ans.net                    internet address = 147.225.1.2
ns.ans.net                      internet address = 192.103.63.100
```

Notice the listing of name servers that are authoritative for the site. You can use that information to query each for more information.

You can also use nslookup to query DNS name servers specific kinds of DNS records. For example, to query for the Start Of Authority (SOA) record from yahoo.com:

```
#nslookup -type=SOA yahoo.com
Server:  joyce.sg.ibm.com
Address:  9.184.8.1

Non-authoritative answer:
yahoo.com
      origin = ns0.corp.yahoo.com
      mail address = hostmaster.yahoo-inc.com
      serial = 2000080812
      refresh = 1800 (30 mins)
      retry = 900 (15 mins)
      expire = 1209600 (14 days)
      minimum ttl = 9 (9 secs)

Authoritative answers can be found from:
yahoo.com      nameserver = ns1.yahoo.com
yahoo.com      nameserver = ns3.europe.yahoo.com
yahoo.com      nameserver = ns5.dcx.yahoo.com
ns1.yahoo.com  internet address = 204.71.200.33
ns3.europe.yahoo.com  internet address = 194.237.108.51
ns5.dcx.yahoo.com  internet address = 216.32.74.10
```

You can also use nslookup to query for the DNS name servers that serve a specific domain. For example, to find the DNS name servers serving the yahoo.com domain:

```

# nslookup -type=NS yahoo.com
Server:  joyce.sg.ibm.com
Address:  9.184.8.1

Non-authoritative answer:
yahoo.com      nameserver = ns3.europe.yahoo.com
yahoo.com      nameserver = ns5.dcx.yahoo.com
yahoo.com      nameserver = ns1.yahoo.com

Authoritative answers can be found from:
ns3.europe.yahoo.com  internet address = 194.237.108.51
ns5.dcx.yahoo.com     internet address = 216.32.74.10
ns1.yahoo.com         internet address = 204.71.200.33

```

You can also use nslookup to list all servers in a given DNS domain. Start by connecting to the DNS server (in interactive mode) with the `server` subcommand in nslookup:

```
> server <DNS server IP address>
```

Then, use the `ls` subcommand of nslookup to list the servers in the domain:

```
> ls <domain name>
```

Recall that the `ls` subcommand of nslookup does a zone transfer; so, zone transfers must be allowed in order for the `ls` subcommand to work.

```

# nslookup
Default Server:  LOCALHOST.0.0.127.in-addr.arpa
Address:  127.0.0.1
>server 9.12.0.5
Default Server:  sp5cw0.itso.ibm.com
Address:  9.12.0.5
> ls alibaba.ibm.com
[merlin.itso.ibm.com]
*** Can't list domain alibaba.ibm.com:Unspecified error
> ls alibaba.ibm.com
[merlin.itso.ibm.com]

merlin          10M IN NS      merlin
aixfw           10M IN A      9.12.0.50
arthur          10M IN A      172.1.1.4
loopback        10M IN A      9.12.0.18
cws             10M IN A      127.0.0.1
                10M IN A      9.12.0.5
Success

```

Notice that the first time the `ls alibaba.ibm.com` command was issued, it failed. That is because we had zone transfers disabled. We then enabled zone transfers and reissued the `ls alibaba.ibm.com` command. As you can see, the entire domain listing was returned this time.

In addition to general-purpose DNS debugging, nslookup is also very useful for ensuring that your DNS is properly configured to not give out any more information than is necessary. For example, if you have a DNS name server that is accessible to hosts on the Internet, you should check to make sure that only those servers that need to be visible outside of your DMZ show up and all others are hidden.

8.3.2 Network File System and Network Information Service

The Network File System (NFS) provides local access to remote file systems; The Network Information Service (NIS) provides central administration of key system files, such as `/etc/passwd` and `/etc/hosts`. Both protocols have a tight history with each other. However, NIS will not be covered in detail here. Instead, we refer you to Section 7.6 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962, or the *Network Information Service (NIS) Overview for System Management in AIX Version 4.3 Network Information Services (NIS and NIS+) Guide* for detailed information on NIS and its derivatives.

NFS and NIS both use Sun's Remote Procedure Call (RPC) protocol, which uses portmapper to allocate ports. The portmap daemon converts RPC program numbers to Internet port numbers. By querying portmapper (TCP/UDP port 111) with the `rpcinfo` program, you can determine on which port an RPC service is listening. Applications do the same thing to locate the port for a given service. Historically, the portmap daemon has been subject to buffer overflow attacks. Be sure to keep it at the latest security patch level, or disable it if it is not needed.

The RPC protocol supports several authentication methods. The one used by NFS and NIS is `AUTH_UNIX` or, sometimes, `AUTH_SYS`. With `AUTH_UNIX`, the client sends user ID and group ID information along with the request. The server assumes this is legitimate information from the client and grants or denies access based on the access control list stored on the server. In the case of NFS, the access control list is the `/etc/exports` file.

You can use the `rpcinfo` command to query the portmap daemon on a remote system to determine if NFS is active. Notice that any user can use `rpcinfo` to query the portmap daemon on a remote system to determine which RPC services are active. If your server is not using any of these RPC services, disable the portmap daemon. Knowing that no RPC services are active on a server still provides useful information to a would-be attacker. It gives them information about what not to attack.

Suppose you want to see which RPC services are active on host, `sp5cw0`:

```
$ rpcinfo -p sp5cw0
```

```
program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100001 1 udp 32780 rstatd
100001 2 udp 32780 rstatd
100001 3 udp 32780 rstatd
100002 1 udp 32781 rusersd
100002 2 udp 32781 rusersd
100008 1 udp 32782 walld
100012 1 udp 32783 sprayd
150001 1 udp 32784 pcnfsd
150001 2 udp 32784 pcnfsd
100083 1 tcp 32769 ttldbserver
100068 2 udp 32785 cmsd
100068 3 udp 32785 cmsd
100068 4 udp 32785 cmsd
100068 5 udp 32785 cmsd
300667 1 tcp 32770
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
200006 1 udp 2049
200006 1 tcp 2049
100005 1 udp 32795 mountd
100005 2 udp 32795 mountd
100005 3 udp 32795 mountd
100005 1 tcp 32774 mountd
100005 2 tcp 32774 mountd
100005 3 tcp 32774 mountd
100024 1 udp 746 status
100024 1 tcp 746 status
200001 1 udp 755
200001 1 tcp 755
200001 2 tcp 755
100021 1 udp 32818 nlockmgr
100021 2 udp 32818 nlockmgr
100021 3 udp 32818 nlockmgr
100021 4 udp 32818 nlockmgr
100021 1 tcp 32775 nlockmgr
100021 2 tcp 32775 nlockmgr
100021 3 tcp 32775 nlockmgr
100021 4 tcp 32775 nlockmgr
100099 1 udp 32839 autofs
1342177279 4 tcp 37661
1342177279 1 tcp 37661
1342177280 4 tcp 49690
1342177280 1 tcp 49690
```

From the output, you can see that NFS (port 2049) is active on the target system.

Knowing that NFS is active, you can now use the `showmount` command to see which filesystems are exported from host `sp5cw0`:

```
$ showmount -e sp5cw0

export list for sp5cw0:
/spdata/sys1/install/pssplpp      (everyone)
/usr/sys/inst.images              (everyone)
/spdata/sys1/install/aix433/lppsource (everyone)
```

Notice that any user can use the `showmount` command to find out which directories are exported from a remote server, and, depending on how the exports are set up, they may also be able to find out the individual server names for the servers that are allowed access to the exported directories. All of this is useful information for a would-be attacker.

There is a very useful NFS client utility (`nfs`) available from:

```
ftp://ftp.cs.vu.nl/pub/leendert/nfsshell.tar.gz
```

The `nfs` utility has an ftp-like interface. With it, you can easily see the dangers of exporting a file system with root access.

You need access to both the `gzip` utility (available from the Bull site) and a C compiler. In our case, we used the version 4.4 C compiler (`/usr/vac/bin/cc`) on AIX 4.3.3. Download the package, uncompress it, and verify that it has not been tampered with (see Section 7.4.1, "Ensuring the integrity of downloads" on page 136). You also need to modify the Makefile by commenting out the Solaris section, uncommenting the AIX section, and specifying the appropriate C compiler. Finally, run `make` to compile the `nfs` utility:

```

# ls -l
total 96
-rw-r----- 1 root      system    29393 Aug 29 09:52 nfsshell.tar.gz
# /usr/local/bin/gzip -d nfsshell.tar.gz
# tar -xvf nfsshell.tar
x nfs
x nfs/Makefile, 1935 bytes, 4 media blocks.
x nfs/mount.h, 4144 bytes, 9 media blocks.
x nfs/mount.x, 3088 bytes, 7 media blocks.
x nfs/nfs.c, 53673 bytes, 105 media blocks.
x nfs/nfs_prot.x, 6405 bytes, 13 media blocks.
x nfs/steal.c, 7750 bytes, 16 media blocks.
x nfs/mount_xdr.c, 2259 bytes, 5 media blocks.
x nfs/mount_clnt.c, 2579 bytes, 6 media blocks.
x nfs/mount_svc.c, 5744 bytes, 12 media blocks.
x nfs/READ_ME, 828 bytes, 2 media blocks.
x nfs/mail
x nfs/mail/Friedrichs, 1687 bytes, 4 media blocks.
x nfs/nfs_prot_xdr.c, 12153 bytes, 24 media blocks.
x nfs/nfs_prot.h, 12107 bytes, 24 media blocks.
x nfs/nfs_prot_clnt.c, 6358 bytes, 13 media blocks.
x nfs/nfs_prot_svc.c, 8287 bytes, 17 media blocks.
# cd nfs
# ls
Makefile      mount.x      nfs.c        nfs_prot_svc.c
READ_ME      mount_clnt.c  nfs_prot.h   nfs_prot_xdr.c
mail         mount_svc.c  nfs_prot.x   steal.c
mount.h      mount_xdr.c  nfs_prot_clnt.c
# vi Makefile
# uncomment the following 4 lines for Solaris 2.x
#CC          = gcc
#CFLAGS      = -DSYSV -DREADLINE -I/usr/local/include
#LIBS        = -lsocket -L/usr/ucblib -R/usr/ucblib -lrpcsvc -lnsl \
              -L/usr/local/lib -lreadline -lhistory -ltermplib

# uncomment the following 3 lines for AIX
CC          = cc
CFLAGS      = -DAIX
LIBS        =
# make nfs
cc -DAIX -c mount_clnt.c
cc -DAIX -c mount_xdr.c
cc -DAIX -c nfs_prot_clnt.c
cc -DAIX -c nfs_prot_xdr.c
cc -DAIX -c nfs.c
cc -g -o nfs mount_clnt.o mount_xdr.o nfs_prot_clnt.o nfs_prot_xdr
.o nfs.o

```

Important

For security reasons, it is not a good idea to have a C compiler loaded on a production system. Compile the program on a non-production system, and only move the executable program to the production system. Be sure also to set the permissions as tightly as possible.

Now, let us take a look at how to use the `nfs` utility to see the dangers of exporting an NFS filesystem with root access. For this example, we used our lab SP system. We started by NFS exporting the `/node5` filesystem from host `node5` using defaults and giving root access to host `node13`.

On `node13`, we run `nfs` from a non-privileged user account. With the `status` subcommand, we see that we currently have permissions as user `nobody` (-2) and group `nobody` (-2). Next, we open a connection to `node5` with the `host` subcommand. Now comes the really scary part. With the `uid` and `gid` subcommands, we change our IDs to root, even though `nfs` was not started as root. From the `export` subcommand, we see that the `/node5` filesystem is exported; so, we mount it with the `mount` subcommand. Then, with the `ls` subcommand, we see that there is a file (`aaa`) with access permissions for root only. The `cat` subcommand proves that we now have root access in the `/node5` file system. Lastly, we change our user ID back to `nobody` (-2) to show that we no longer have access to the file (`aaa`):

```
$ nfs
nfs> status
User id      : -2
Group id     : -2
Transfer size: 0
nfs> host node5
Using a privileged port (1023)
Open sp5en05 (192.168.5.5) TCP
nfs> uid 0
nfs> gid 0
nfs> export
Export list for sp5en05:
/node5      everyone
nfs> mount /node5
Using a privileged port (1022)
Mount `/'node5', TCP, transfer size 8192 bytes.
nfs> ls -l
-rwx----- 1      0      3      17 Aug 29 10:01 aaa
nfs> cat aaa
this is file aaa
nfs> uid -2
nfs> cat aaa
aaa: Permission denied
```


If the `/node5` file system had not been exported with root access, we would not have been able to view the file (`aaa`) with the `nfs` utility. There are two lessons here. First, do not export file systems with root access unless absolutely necessary. Second, if you load the `nfs` utility, make sure you set the permissions so that only root can run it. For more information about using the `nfs` utility, run the `help` subcommand.

In an SP environment, PSSP uses NFS during node installation and customization. In most SP environments, NFS is also used for normal day-to-day support activities (automounted home directories, nightly `mksysb` backups to the control workstation, and so on). If you are dependent upon having NFS active within the SP environment, be sure to limit it to only the internal SP networks (administrative Ethernet and switch) by being very explicit in your export list. Also, in an SP environment, avoid using NIS for user management because of the security exposure. More secure alternatives include: `supper`, DCE user management, and NIS+. Both `supper` and DCE user management are supported directly by PSSP.

Newer alternatives to NFS and NIS are Secure NFS and NIS+. Secure NFS and NIS+ use Sun's RPC protocol, just as NFS and NIS do, but they differ in the authentication protocol used. NFS and NIS use the `AUTH_UNIX` method; Secure NFS and NIS+ use the more secure `AUTH_DES` method.

The `AUTH_DES` protocol used by Secure RPC employs DES-based, public-key, private-key encryption. This ensures proper authentication for all requests and, therefore, provides a secure way of sharing data (Secure NFS) and performing user management (NIS+). To use Secure RPC, you need to set up the public and private keys and create principals for all users. The most straightforward way of doing this is with NIS+, but its setup is not trivial. NIS+ is included with AIX 4.3.3.

For more information about Secure NFS, see Section 7.5.4 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962. For more information about setting up and administering NIS+, see the system manual, *Network Information Service (NIS) Overview for System Management in AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*.

Protecting NFS

With NFS, the most common security mistake is exporting directories with excessive privileges. By default, the NFS export on AIX gives read-write access to all hosts. Where possible, change this to read-only, and explicitly specify only those hosts who need access to the directories on a directory by

directory basis. Export only the needed directories, and specify the exports as low in the directory as possible. For example, on an SP control workstation, the `/spdata/sys1/install/pssplpp` directory is exported to the nodes instead of the top-level `/spdata` directory. You should also keep an eye on the content in the directories that are NFS exported to ensure that no sensitive data is inadvertently exposed.

To provide further protection, consider blocking the portmap port (port 111) at the firewall. You may still want to allow internal users to have portmap (and NFS) access while preventing it for external users. Since attackers can still ferret out the fact that you are using portmap (and NFS) by scanning non-privileged ports (above 1023), you should also set up protection against port scanning. For more information about how to protect against port scanning attacks, see Chapter 6, "Port and network scanning" on page 83.

By default, NFS is started on system startup with the `/etc/rc.nfs` script out of `/etc/inittab`. NFS is comprised of five separate daemons: *biod*, *nfsd*, *rpc.mountd*, *rpc.statd*, *rpc.lockd*. Each are briefly described here:

- *Biod* - The *biod* daemon handles client requests and runs on the NFS client (and usually also the NFS server). When a user (client) wants access to a file on an NFS-mounted file system, *biod* sends the request to NFS server.
- *Nfsd* - The *nfsd* daemon services requests from clients for file system operations.
- *Rpc.mountd* - The *rpc.mountd* daemon answers requests from clients for file system mounts. It is required when you mount an NFS file system.
- *Rpc.statd* - The *rpc.statd* daemon provides crash recovery functions for the NFS locking services (provided by the *rpc.lockd* daemon).
- *Rpc.lockd* - The *rpc.lockd* daemon processes lock requests for concurrent access to files.

NFS supports the UNIX advisory locking functions with the *rpc.lockd* and *rpc.statd* daemon. If you are not running applications that require this locking functionality, disable *rpc.lockd* and *rpc.statd*. (It is also possible to disable *rpc.mountd* after all client mounts have successfully completed, but this is much harder to do and runs the risk of causing additional problems for clients that need to remount the file systems.) These daemons have been successfully exploited through buffer overflow attacks to gain root access on the system. If you do need them, ensure that you are at the latest patch level for fileset `bos.net.nfs.client`.

To disable `rpc.lockd` and `rpc.statd`, comment out their entries in `/etc/rc.nfs`, and stop the daemons (in the order shown):

```
# stopsrc -s rpc.lockd; stopsrc -s rpc.statd;
```

To unexport a single directory, use the `exportfs -u <directory>` command. To unexport all directories in the `/etc/exports` file, use the `exportfs -au` command.

8.3.3 Simple Mail Transfer Protocol (SMTP)

In AIX, the default SMTP server software is sendmail. Sendmail has gotten a bad reputation for having had many security vulnerabilities in the past. For that reason, you should have as current a version as possible, and always keep it at the latest patch level. A wealth of information about sendmail can be found at the sendmail homepage:

<http://www.sendmail.org>

With AIX, the sendmail version you receive depends on the AIX version as shown in Table 5.

Table 5. AIX versions of sendmail

| AIX version | Sendmail version |
|-------------|------------------|
| AIX 3.2.5 | Sendmail 5.6.4 |
| AIX 4.1.4 | Sendmail 5.6.4 |
| AIX 4.1.5 | Sendmail 5.6.4 |
| AIX 4.2.0 | Sendmail 8.7 |
| AIX 4.2.1 | Sendmail 8.7 |
| AIX 4.3.0 | Sendmail 8.7 |
| AIX 4.3.1 | Sendmail 8.8 |
| AIX 4.3.2 | Sendmail 8.8 |
| AIX 4.3.3 | Sendmail 8.9 |

The sendmail configuration file is `/etc/sendmail.cf`. One of the most common security mistakes made with the configuration file is to allow the use of the `VERFY` and `EXPN` built-in commands. These should be disabled by adding the following line in the `/etc/sendmail.cf` file:

```
O PrivacyOptions=noexpn,novrfy
```

VRFY is used to verify names of valid users, and EXPN is used to obtain the actual delivery addresses. Not only do these commands expose sensitive information to would-be attackers, but they can sometimes also be used by the attacker to forge mail from the server. This is such a common security mistake that scan programs, such as NSA (Section 4.4, "Network Security Auditor (NSA)" on page 49), have built-in tests for it.

There a couple of other security-related options that you might want to consider adding to the sendmail `PrivacyOptions` line:

- `restrictmailq` restricts ordinary users from viewing the status of all queued mail.
- `restrictqrun` restricts ordinary users from causing sendmail to process its queue.
- `goaway` restricts SMTP status queries. Note: This option takes care of setting `novrfy`, `noexpn`, `authwarnings`, `needmailhelo`, `needexpnhelo`, and `needvrfyhelo`.

To include all of the above options in addition to the `novrfy` and `noexpn` options, add the following line to the `/etc/sendmail.cf` file (instead of the one shown earlier):

```
O PrivacyOptions=goaway,restrictmailq,restrictqrun
```

In the following example, we show both the VRFY and the EXPN commands. Sendmail listens on TCP port 25. To connect to the local sendmail daemon, we simply telnet to port 25 on localhost (0):

```
# tn 0 25
Trying...
Connected to 0.
Escape character is '^T'.
220 sp5en0.msc.itso.ibm.com ESMTP Sendmail AIX4.3/8.9.3/8.9.3; Tue, 29 Aug 2000 17:08:15
-0500
vrfy khorck
250 <khorck@sp5en0.msc.itso.ibm.com>
expn khorck
250 <khorck@sp5en0.msc.itso.ibm.com>
```

Notice also how much information is provided by the Sendmail banner. It gives the version of AIX (4.3) as well as the version of sendmail (8.9.3). This is a wealth of information for would-be attackers. They now know the operating system type (AIX), the operating system version (4.3) that sendmail is running on this server, the sendmail version (8.9.3), and the timezone

where the server is located. Taken individually, each of these pieces of information is damaging enough, but taken together, the damage can be much worse. The would-be attacker can now perform a simple search on the Web to find all known exploits for this version of sendmail running on this version of the operating system. For that reason, you should turn off the banner information.

To remove it, edit `/etc/sendmail.cf`, and look for the `SMTPGreetingMessage` line:

```
O SmtpgreetingMessage=$j Sendmail $v/$Z; $b
```

The `$v` is AIX version, the `$Z` is sendmail version, and the `$b` shows the time and date (unless you have defined the macro). Remove the fields as required, and then refresh the sendmail daemon for the changes to take effect.

Beginning with version 8.8, sendmail supports the option to run as a user other than root. This can reduce the impact of an attacker using a remote exploit to gain access to the local system through a sendmail vulnerability. However, this option should only be used on systems that are either dedicated sendmail servers or bastion hosts in a DMZ.

This option is ideal for sendmail relays or hubs but does not work very well for servers with normal user accounts and access. For the latter type of server, the unprivileged ID used by sendmail needs read access to the `.forward` and `:include:` files owned by the normal users. In general, the problems involved with opening access to these files for the unprivileged sendmail ID outweigh the benefits.

Note that sendmail in daemon mode still runs as root so that it is able listen on TCP port 25. However, when a connection is opened, the child forked by the sendmail daemon to process the request runs with the unprivileged ID.

To set this up, we recommend that you create a special unprivileged user ID of either `postman` or `mailnull` and a matching group for the exclusive use of sendmail. Lets assume that you use `postman` for the user ID and group ID. Once these are created, add the following two lines to your `/etc/sendmail.cf` file:

```
O RunAsUser=postman:postman
```

```
O DefaultUser=postman:postman
```

The `DefaultUser` option is used by sendmail when it is unable to run as root and has no other choices. (This option should be set even if you do not plan to use the `RunAsUser` option.) By default, sendmail will always try to not run as

root unless needed. For example, it does not need root to write to `$HOME/.deadletter`.

Additionally, with the `RunAsUser` option, you need to give the unprivileged sendmail id access to the sendmail queue:

```
# chown postman /var/spool/mqueue
```

Note that the commercial version of sendmail, Sendmail Switch (<http://www.sendmail.com>), automatically configures sendmail to run as an unprivileged user.

Important

Do not run sendmail from the command line with the `RunAsUser` option because it may cause sendmail to lose its root authority.

As has already been mentioned, you probably do not want to use the sendmail `RunAsUser` option on a server that has normal user account access. However, you can still remove the root SUID from the sendmail executable:

1. Create an unprivileged user account, such as `mailnull`, that is not allowed login privileges on the system.
2. Change ownership of the sendmail files:

```
# chown mailnull /usr/sbin/sendmail
# chown mailnull /etc/sendmail.cf
# chown mailnull /etc/sendmail.st
# chown mailnull /var/spool/mqueue
```

3. Redo the SUID to the unprivileged sendmail ID:

```
# chmod 4511 /usr/sbin/sendmail
```

If you are running sendmail on a server where users have shell access, consider setting the `SafeFileEnvironment` option in `/etc/sendmail.cf`. One type of denial of service attack involves symbolically linking bogus files to system configuration files or system device files and then getting sendmail to overwrite them. The `SafeFileEnvironment` option, among other things, ensures that the file being written to is a plain file (as opposed to a symbolic link or device file). Consider setting this option as follows in the `/etc/sendmail.cf` file:

O SafeFileEnvironment=/
/

This causes sendmail to do the extra level of checking on the files but does not go so far as to require that all files to which it appends exist in a chrooted directory tree set up specifically for sendmail. For more information about setting up a chrooted environment for sendmail, see *Sendmail*, ISBN 1-5659-2222-0 (aka “The Bat Book”).

Another security option that you can use with sendmail is the sendmail restricted shell (`smrsh`). With this special shell, you can restrict sendmail to a specific directory tree from which it can execute programs. This enables you to control the programs that sendmail can execute, such as those that are specified in the `.forward` files of normal users. This also seriously hinders the damage that a would-be attacker can cause by exploiting a hole in sendmail. Without it, the attacker can execute arbitrary programs and issue commands as root on the system. You can get more information about how to set up and use `smrsh` by downloading the latest sendmail source tree from ftp.sendmail.org. After expanding the distribution, look for the README file in the `smrsh` subdirectory.

It is also possible to run sendmail in a chrooted environment, which limits sendmail to a specific directory tree. This is similar to `smrsh` but takes things a step further by completely disallowing sendmail any access outside of its own (chrooted) directory tree. There are numerous articles on the Internet about how to set this up. In addition, there are tools, such as the Firewall Toolkit (http://www.tis.com/research/software/fwtk_over.html), that can set it up for you.

Whenever you make sendmail configuration changes, always verify them with:

```
# sendmail -v -bi
```

Running this will alert you to any permissions errors on the files and directories needed by sendmail.

AIX 4.3.3 has various sendmail enhancements, which are covered in Section 7.27 of *AIX Version 4.3 Differences Guide*, SG24-2014. You should also visit the sendmail homepage to get the latest security information as well as additional methods for securing sendmail. Two special points of interest are the sendmail FAQ, at <http://www.sendmail.org/faq>, and the information about how to set up anti-spam measures, at <http://www.sendmail.org/antispam.html>.

You may also want to consider restricting access to SMTP TCP port 25. There are a number of ways of doing this. You can control direct access to the port by blocking it at the firewall, or you can block access to it at the server level with IPsec, which is included in the AIX 4.3 bos.net.ipsec files. For more information on IPsec, refer to Section 7.3 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962, and Chapter 9 of *A Comprehensive Guide to Virtual Private Networks, Volume III*, SG24-5309.

A slightly different approach is to protect access to sendmail rather than just SMTP TCP port 25. One way to do this is by placing sendmail under the control of inetd, and protecting it with TCP Wrapper. For more information on how to do this, see Section 7.4.4 of *Exploiting RS/6000 SP Security: Keeping it Safe*, SG24-5521.

It is also possible to just replace sendmail. One alternative is a product known as Postfix. Postfix is essentially a re-architecture of sendmail that was done with both security and speed in mind. The developer of Postfix, Wietse Venema, is also the developer of TCP Wrapper and SATAN. For more information about Postfix, see Section 7.4.4 of *Exploiting RS/6000 SP Security: Keeping it Safe*, SG24-5521.

Postfix is available from:

<ftp://ftp.porcupine.org/pub/security/index.html>

(At one time, Postfix was known as IBM Secure Mailer; so, you may find additional information about Postfix under that name as well.)

8.3.4 Simple Network Management Protocol (SNMP)

AIX ships with SNMPv1. The only security mechanism supported by SNMPv1 is through the use of community names (see the `/etc/snmpd.conf` file). An SNMP community consists of one or more hosts grouped under a community name. The community name is used for authentication in the SNMP request packet. When a request is received, the SNMP agent verifies that the IP address of the host making the requests matches one of the hosts belonging to the community name. If the requesting host is a member of the community, the SNMP agent then determines if the requesting host is authorized for the requested access to the specified MIB.

By default, AIX actively listens for SNMP requests on TCP and UDP port 161. Therefore, to further secure SNMP, you should change the community names from their defaults (such as *public*) to names that are harder to guess. You should also disable MIB read and write permissions as much as possible.

8.3.5 Trivial File Transfer Protocol (TFTP)

TFTP is typically used to boot diskless workstations and network devices. It is basically a stripped-down version of FTP and has very little built-in security. TFTP uses UDP port 69. On SP systems, both `tftp` and `bootps` are required for node installation and customization activities.

The following screen shows an example of `tftp` use within our lab environment. (Server `sp5cw0` is the control workstation for the SP, and server `arthur` is a standalone server.):

```
[sp5cw0] $ cat /etc/tftpacces.ct1
# PSSP and NIM access for network boot
allow:/tftpboot
allow:/usr/lpp/ssp
allow:/etc/SDR_dest_info
allow:/etc/krb.conf
allow:/etc/krb.realms

tn arthur

...

[arthur] $ tftp sp5cw0
tftp> ?
Commands may be abbreviated.  Commands are:

connect      Connect to remote tftp
mode         Set file transfer mode
put          Send file
get          Receive file
quit         Exit tftp
verbose      Toggle verbose mode
trace        Toggle packet tracing
status       Show current status
binary       Set mode to octet
ascii        Set mode to netascii
timeout      Set total retransmission timeout
?            Print help information

tftp> get /etc/krb.conf
Received 36 bytes in 0.1 seconds
tftp> quit
```

TFTP access control is provided through the `/etc/tftpaccess.ct1` file. The default TFTP access control for an SP system allows access to the `/tftpboot` directory, the `/usr/lpp/ssp` directory, the `/etc/SDR_dest_info` file, the `/etc/krb.conf` file, and the `/etc/krb.realms` file. Notice that from `arthur`, a standalone server that is completely unrelated to the SP system, that we were able to grab a copy of the `/etc/krb.conf` file without having to provide any authentication information whatsoever.

You may want to consider enabling tftp on the SP system only when it is needed for node installations or customizations. Of particular interest to a would-be attacker is the `/tftpboot` directory, which contains information about the nodes in the SP system. It is also a good idea to make sure that tftp access is blocked at your firewalls as well.

8.3.6 Securing X11

If you are running X11, you need to be especially concerned about security. An insecure X11 poses a major security risk. If access control is improperly set, attackers with simple programs (built on the X libraries) can capture keystrokes (such as account name, password, and other sensitive data), take control of the X windows display, and monitor the applications that are run. There are utilities available on the Internet that scan networks looking for active X servers, and, once found, attempt to connect to them for the sole purpose of monitoring and capturing keystrokes. In fact, utilities, such as `xwd` and `xwud` (found in the `X11.apps.clients` filesset), can be used to silently monitor a remote X server:

```
# xwd -root -display <server to be snooped>:0.0 > /tmp/tempfile
# xwud -in /tmp/tempfile
```

X provides two forms of access control:

- `xhost` - provides access control through IP address and/or host name
- `xauth` - provides access control through cookies

A comparison of the two can be found at:

<http://www.finnigan.de/products/icisnotes/xsecurity.htm>

Of the two, `xhost` is the more commonly used method for access control. Running `xhost` with out parameters lists the hosts that have been granted access to the X server. A fairly common user mistake is to issue the `xhost +` command. This disables all access control for the X server, which means that any host (even one belonging to an attacker) anywhere can connect to it. Once connected, programs, such as `xwd`, `xwud`, or worse, can easily be run to gather sensitive information and monitor activities. Some port scanners are able to detect that access control is turned off for an X server. For example, when NSA detects this, it reports it as follows:

```
o X server has no access control [port 6000].
```

If you want to secure the X server, start by running the `xhost -` command to turn on access control. You can safely run this command even if you are not sure if access control is already enabled. Running this command will not remove any existing connections, it will only prevent new, unauthorized connections from occurring. You can then augment your access control list by iteratively running the `xhost + <hostname>` for each additional host that is to have access to the X server. For example:

```
# xhost -
access control enabled, only authorized clients can connect

# xhost
access control enabled, only authorized clients can connect
INET:loopback
INET:merlinen0
INET:firewall
LOCAL:

# xhost +sp5cw0
sp5cw0 being added to access control list

# xhost -sp5cw0
sp5cw0 being removed from access control list
```

You can obtain a finer grain of access control with `xauth`. For information on how to set this up, see *The X Windows System Administrator's Guide, Volume 8*, ISBN 0-9371-7583-8.

Unattended workstations should always have their screens locked. This not only applies to PCs but also to workstations running X. The `xlock` command provides a password protected screen lock, and the `xss` command enables you to automatically activate the screen lock after some period of inactivity. These features are also built in to the CDE desktop.

The X server uses TCP port 6000. This port should either be blocked at the firewall or filtered on the server with the IPsec facility (included in the AIX 4.3 `bos.net.ipsec` fileset). For more information on IPsec, refer to Section 7.3 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962, and Chapter 9 of *A Comprehensive Guide to Virtual Private Networks, Volume III*, SG24-5309.

Terminal emulators are another area of concern for security and X. When using `xterm` (or other similar terminal emulators), use the secure keyboard option activated from the main menu. To activate the main menu, place the mouse pointer in the terminal window, then hold down the control key and mouse pointer button simultaneously. Once activated, the secure keyboard

option directs all keyboard inputs to the `xterm` command only, thus, preventing keystroke snooping. When secure keyboard is activated, the `xterm` foreground and background colors are interchanged to provide a visual reminder that the option is in force. Be aware that only one X client at a time can use this option, so you will need to enable and disable it as you move from X client to X client. For more information on the secure keyboard option, see the `xterm` man page. Also, note that the secure keyboard option does not protect against network sniffing since user ids and passwords are still transferred in the open. To protect against this, you can run X over SSH. For more information on SSH, see Section 5.1, "Secure Shell (ssh)" on page 59.

8.3.7 File Transfer Protocol (ftp)

The `ftp` protocol is used for transferring files between two hosts. As mentioned in Section 5.1, "Secure Shell (ssh)" on page 59, `ftp` transfers passwords (for authentication) in the open without encryption and is, thus, susceptible to spoofing. As such, running `ftp` over the Internet poses a huge security risk. Your first, best option is to replace `ftp` with a more secure variant. If that is not possible, steps should be taken to further secure `ftp`.

If possible, consider using `sftp` or `scp` from the SSH package to replace `ftp`. SSH is based on a public-key, private-key authentication scheme and provides encryption for secure communication channels. Refer to Section 5.1, "Secure Shell (ssh)" on page 59 for more details.

Another option is to protect `ftp` with TCP Wrapper. TCP Wrapper provides access control to services (such as `ftpd`) controlled by the `inetd` daemon (`/etc/inetd.conf`). Refer to Section 5.2, "TCP Wrapper" on page 77, for more details.

By default, the `ftp` command displays a banner of information when run:

```
# ftp sp5cw0
Connected to sp5en0.
220 sp5en0 FTP server (Version 4.1 Sun Jul 9 18:28:14 CDT 2000) ready.
Name (sp5cw0:root):
```

Of particular interest to would-be attackers is the `ftp` version information. Lists of vulnerabilities are readily available on the Internet. With the banner information in hand, the attacker can then tailor the attack to that specific version of `ftp`, saving many hours of trial and error. You can remove the banner as follows:

```

# dspcat -g /usr/lib/nls/msg/en_US/ftpd.cat > /tmp/ftpd.msg

# vi /tmp/ftpd.msg

$delset 1
$set 1
$quote "
...
...
"signal: %m"
9      "%s FTP server ready."
...

# gencat /tmp/ftpd.cat /tmp/ftpd.msg

# cp -p /tmp/ftpd.cat /usr/lib/nls/msg/en_US/ftpd.cat

```

Here is how it looks after the banner replacement:

```

# ftp sp5cw0
Connected to sp5en0.
220 sp5en0 FTP server ready.
Name (sp5cw0:root):

```

There are two modes of ftp: active and passive. Some organizations require that only passive ftp be allowed. For any file transfer, ftp uses two channels, one for control and one for data. In active ftp, the server initiates the data channel. In passive ftp, the client initiates the data channel. With active ftp, it is possible for a malicious ftp server on the Internet to open a data port on an internal server for attack purposes. Passive ftp prevents this from happening.

Passive ftp is the mode which is supported by most Web browsers and many GUI-based ftp clients. For passive ftp to work, both ftp server and ftp client must support it. The ftp server code in AIX supports both active and passive modes, as does the AIX 4.3.3 ftp client code (via the `passive` subcommand). Active ftp is the default in AIX.

For more information on the differences between active and passive ftp, visit:

<http://www.geek-speak.net/ftp/ftp1.html>

Another alternative for ftp is to use SOCKS (see Section 2.2.2, "Circuit level firewalls" on page 15) as a security measure. There are two different ways in which SOCKS can be implemented to further secure ftp. Either the TCP stack can be socksified, or the ftp client can be socksified. Socksified clients are the more common solution. You can obtain a socksified ftp client (`rftp`) from

`ftp://isscftp.raleigh.ibm.com/IBM/socks/AIX/bin/`. There are also socksified ftp clients incorporated in Web browsers, such as Netscape or Internet Explorer. To socksify the entire TCP stack, refer to the following, Section 8.3.8, "Protecting TCP services using SOCKS".

8.3.8 Protecting TCP services using SOCKS

SOCKS is a common means for clients behind a firewall to connect to servers on the Internet. Among other things, SOCKS protects the clients by hiding their IP addresses from the servers on the Internet. Only the Internet-facing IP address of the firewall is exposed to the servers. SOCKS also enables you to define your own IP address ranges (such as 192.168.130.0) for machines behind the firewall, since these IP addresses do not need to be registered externally.

SOCKS servers are similar to proxy servers. The main difference is that a SOCKS server identifies the user and redirects the application through to the server, whereas a proxy server actually performs the functionality on behalf of the client. SOCKS is also more generalized with little limitation on the supported protocols (dependent, of course, on the specific SOCKS implementation).

To use SOCKS, you either need a socksified client or a socksified TCP stack. There are a number of socksified clients available for AIX, such as `rftp` for ftp, `rtelnet` for telnet, and `rtn` for tn. To use these clients, you will also need access to a SOCKS server. (There are AIX versions available for download from the Bull site.) The advantage of a socksified TCP stack is that you do not need to replace your clients. They automatically and transparently make use of the SOCKS support provided in the TCP stack. AIX 4.3.3 contains support for automatic socksification (SOCKS v5). Configuration can be found in the *AIX Version 4.3 Differences Guide*, SG24-2014. Even with a socksified TCP stack, you will still need access to a SOCKS server.

8.4 Step 3: Set proper network (no) options

Network options control how TCP, UDP, and ICMP behave on an AIX machine. Default settings are insufficient for the Internet world. You use the `no` command to configure network attributes. Some of the default settings need to be changed in order to further tighten security on the machine. The changes you make are dependent on the level of security required for your environment. The options along with their default settings (for AIX 4.3.3) are explained here to help you make those choices.

8.4.1 SYN attack protection

The `clean_partial_conns` option is used to avoid SYN attacks. In a legitimate 3-way TCP/IP handshake, the client sends a SYN packet (TCP packet with SYN bit set) to the server. The server replies with a SYN/ACK packet. Finally, the client acknowledges the SYN/ACK packet with an ACK packet. This completes the 3-way handshake.

In a SYN attack, a client with a spoofed IP address sends many repeated SYN packets to the server. For this attack to work, the spoofed IP address must not be reachable by the server. When the server receives the SYN packets, it responds by sending the SYN/ACK packets back to the spoofed IP address. Since the spoofed IP address is unreachable, the server never receives the corresponding ACK packets in return, and the pending connection table on the server eventually fills up. This is a denial of service attack because once the connection table is full, the server is unable to service legitimate requests.

Setting `clean_partial_conns` to 1 (the default is 0) causes the server to periodically clear incomplete 3-way handshake connections, thus, reducing the likelihood of a successful SYN attack and allowing legitimate client connections to proceed.

8.4.2 Broadcast protection

The `bcastping` option is used to control the response to ICMP echo packets directed at the broadcast address. Systems are vulnerable to smurf attacks when `bcastping` is enabled. Smurf attacks occur when a client sends huge amounts of spoofed ICMP traffic to the broadcast addresses. With broadcast address ping allowed, each host on the network will respond with an ICMP echo reply. This worthless network traffic wastes bandwidth causing normal operations to slow or, in extreme cases, to halt. Smurf attacks are another form of denial of service attacks.

More information on smurf attacks can be found at:

<http://www.pentics.net/denial-of-service/white-papers/smurf.cgi>

Setting `bcastping` to 0 (the default is 0) causes the hosts to not respond to broadcast ping requests. However, smurf attacks are more effectively dealt with by routers. The routers should be configured to not allow ICMP echo packets from the Internet to reach the broadcast addresses.

Similarly, `directed_broadcast` should be set to 0 (the default is 1) to prevent directed packets from reaching the broadcast address of a remote network through a gateway.

Note that for SP systems, the `hats` topology daemon requires `bcstping` to be set to 1 unless APAR IX86922 has been installed throughout the SP system. This solution also requires the Topology Services daemon to prepare an all-known-IP list for `netmonAdapterHealth`.

8.4.3 IP routing options

The `ipignoreredirects` option is used to control ICMP redirects. When servers or routers find a more efficient way of routing packets, they send back ICMP redirect packets to the originating host informing it of the more efficient route. This opens a potential security loophole because it can be used to alter a legitimate route to redirect traffic to an unintended destination. The recommended alternative is to set `ipignoreredirects` to 1 (the default is 0) and add static routes.

Similarly, the following `no` options should be set to prevent illegal access through source routing:

- Set `ipsendredirects` to 0 (the default is 1).
- Set `ipsrcroutesend` to 0 (the default is 1).
- Set `ipsrcrouterrecv` to 0 (the default is 0).
- Set `ipsrcrouteforward` to 0 (the default is 1).
- Set `ip6srcrouteforward` to 0 (the default is 1).
- Set `icmpaddressmask` to 0 (the default is 0).
- Set `nolocsroute` to 0 (the default is 0).
- Set `tcp_pmtu_discover` to 0 (the default is 0).
- Set `udp_pmtu_discover` to 0 (the default is 0).

Source routing attempts to control the path through the network to the destination. It can be manipulated such that the source is directed to reach the destination through an insecure interface rather than a secure one. Setting the `no` options as outlined prevents source routing control from external sources. For maximum security, add only static routes and do not allow routes to be dynamically set by dynamic routing daemons (that can be abused).

Lastly, you should set `ipforwarding` to 0 (the default is 0) unless the server is acting as a gateway. For example, the control workstation for a large SP

system with multiple administrative Ethernet segments will typically have `ipforwarding` set to 1 so that traffic can pass from segment to segment. This setting is used to allow packets to pass from one interface to another in a dual-homed server. It is a very big security concern on servers attached to two different networks (such as in a DMZ). Not only should `ipforwarding` be turned off (set to 0), but you should also have some form of monitoring or auditing to ensure that it does not get inadvertently turned back on (set to 1).

8.5 Step 4: Tighten up user accounts

Remove unnecessary default accounts, and tighten the default attributes for the remaining ones. Unneeded default accounts are an open invitation to would-be attackers, and insecure normal user accounts pose one of the most insidious threats to security on a system. An attacker who has commandeered a legitimate user account is a very difficult animal to track and capture. Much better to have and enforce strong account policies so that the attacker never gains this toehold in the first place.

In this step we give recommendations about default accounts that can possibly be removed and account options that can be tightened up. Remember that even though your server may not contain sensitive data, once compromised, it can certainly serve as a launch point for attacks on other servers.

8.5.1 Removing unnecessary default accounts

By default, several users and groups are created on the system during installation. They are usually not required and can be safely removed.

The default users that can be removed are:

- `uucp` and `nuucp` - for use by the `uucp` protocol
- `lpd` - for use by the printing subsystem
- `imnadm` - for use by the IMN search engine
- `guest` - for use by users without accounts on the system

The default groups that can be removed are:

- `uucp` - for use with the `uucp` and `nuucp` accounts
- `printq` - for use with the `lpd` account
- `imnadm` - for use with the `imnadm` account

You can remove these users and groups as follows:

```
# for u in uucp nuucp lpd imnadm guest; do rmuser -p $u; done
# for g in uucp printq imnadm; do rmgroup $g; done
```

Once you have removed these users and groups, verify the correctness of the user, group, and password files:

```
# usrck -y ALL
# grpck -y ALL
# pwdck -y ALL
```

These commands should be run on a regular basis to ensure consistency and detect problems.

8.5.2 Setting user attributes

User attributes are set in the `/etc/security/user` file. They can either be set globally or on an individual user basis. This file is in stanza format, and the `default` stanza contains the global settings. A setting in an individual user stanza overrides the corresponding setting in the `default` stanza. The user attributes along with the recommended and default settings are listed in Table 6.

Table 6. AIX account attributes

| Attribute | Recommended value | Default value |
|--------------|-------------------|---------------|
| loginretries | 3 | 0 |
| registry | files | |
| rlogin | false (for root) | true |
| tpath | on | nosak |
| ttys | tty0 (for root) | ALL |
| umask | 077 | 022 |

Descriptions of each of the attributes follows:

loginretries

This attribute controls the number of invalid login attempts before the account is locked. The possible values are positive integers or 0 to disable it. Once the number of retries is exceeded, the account is locked. To be usable again, the

account will need to be unlocked by the system administrator (`smitty failed_logins`).

registry

This attribute controls where the account is administered. A value of *files* means that the account will be administered with local files rather than remotely with something, such as NIS.

rlogin

This attribute controls whether or not the account can be accessed by remote logins. The `rlogin` and `telnet` commands support this attribute. The possible values are `true` and `false`. The value should be set to `false` for `root`, thus, requiring users to use the `su` command from their normal user accounts to gain root access. Not only is this more secure, but it also provides an audit trail with the `/var/adm/sulog` file.

tpath

This attribute defines the trusted path characteristics for the account. The possible values are:

- `nosak` - The Secure Attention Key (SAK) key (^X^R) has no effect.
- `notsh` - The SAK key logs you out. You can never be on the trusted path.
- `always` - When you log in you are always on the trusted path.
- `on` - The trusted path is entered when the SAK key is hit.

Note

This attribute only takes effect if the `sak_enabled` attribute (in the `/etc/security/login.cfg` file) is set `true` for the port you are logging into.

ttys

This attribute controls which terminals can be used by this account. Terminals can either be explicitly allowed (`tty0`) or explicitly denied (`!tty0` - notice the exclamation point preceding the terminal name). Consider setting this attribute to `tty0` (console) for `root`.

umask

This attribute defines the default umask for this account. It is specified as a three-digit octal mask. For tight security, a value of 077 is recommended. This option is critically-important for securing permissions within the file systems.

Note

For SP node installation and customization, ensure that the umask is set to 022 (at least temporarily during the process).

Two additional account attributes to consider are:

expires

This attribute defines the expiration time for the account. It is useful in situations, such as classes or schools, where the account should expire at a known point in time in the future.

logintimes

This attribute defines the times when the account may be used. It is useful for limiting the times when normal user activity can occur on the system. For example, you could use it to prevent normal users from logging in during regularly-scheduled maintenance windows.

8.5.3 Securing root

The root account requires special attention. The first step is to limit access to as few people as possible; the fewer the better. Because AIX accounts support the function of roles, it is possible to allow selected accounts to execute certain administrative commands without having to have access to root. Examples include managing accounts, performing backups, and rebooting the system. This functionality is covered in detail in Chapter 3 of *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962.

A good way to protect root is by disabling remote login for the account. This forces the authorized users to first log in with their normal user accounts and then use the `su` command to gain root access. This provides accountability through the audit trail in the `/var/adm/sulog` file (which, itself, should be protected with some tool, such as Tripwire).

In addition to the recommendations for securing normal user accounts, consider these additional recommendations for securing root:


```
Escape character is '^T'.
```

```
telnet (merlin)
```

```
Unauthorized Access Prohibited.  
login:
```

Notice how little useful information is revealed in the new herald.

Lastly, the `TMOUT` and `TIMEOUT` environment variables should be set in the `/etc/profile` file. These variables control how long the account can remain idle with no activity before being automatically logged off.

For example, to set these timeouts to 10 minutes (600 seconds), include the following line in the `/etc/profile` file:

```
TMOUT=600 ; TIMEOUT=600 ; export readonly TMOUT TIMEOUT
```

8.6 Step 5: Set up strong password policy

AIX provides a set of attributes that enable you to implement strong password policy on the systems. Among other things, you can use these attributes to force users (including root) to choose passwords that are difficult to guess and are not available in the standard UNIX dictionary. However, use of these attributes does not alleviate the need to run cracking tools, such as John the Ripper (see Section 7.2, "John the Ripper" on page 119). The AIX attributes are listed in Table 7 on page 195 and discussed in the next section.

8.6.1 Modifying user password attributes

The AIX attributes used to define strong password policy are contained in the `/etc/security/user` file. Attributes specified for the `default` stanza apply globally but can be overridden for individual user accounts. the attribute

name, the recommended value, the default value, and the max value for the AIX attributes are shown in Table 7.

Table 7. Recommended values for password settings

| Attribute | Recommended value | Default value | Max value |
|-------------|-----------------------|---------------|-----------|
| dictionlist | /usr/share/dict/words | | |
| histexpire | 26 | 0 | 260 |
| histsize | 8 | 0 | 50 |
| minage | 0 | 0 | 52 |
| maxage | 12 (5 for root user) | 0 | 52 |
| maxexpired | 4 | -1 (no limit) | 52 |
| minalpha | 4 | 0 | 8 |
| minother | 1 | 0 | 8 |
| minlen | 6 (8 for root user) | 0 | 8 |
| mindiff | 3 | 0 | 8 |
| maxrepeats | 3 | 8 | 8 |
| pwdwarntime | 14 | 0 | 0 |

Descriptions of each of the attributes follows:

dictionlist

This attribute ensures that passwords are not words from the standard UNIX dictionary (under AIX, this is the `/usr/shared/dict/words` file, which is part of the `bos.data` fileset). When passwords are changed, the dictionary file is checked to make sure that the chosen password is not contained in the dictionary. As root, you can add or delete words from this file to tailor for your environment (for example, by adding common computer terms, such as *hostname*).

In addition to the `bos.data` fileset, you will also need the `bos.txt.spell` fileset (required for text processing), neither of which get installed by default.

histexpire

This attribute specifies the number of weeks before a user is allowed to reuse a password.

histsize

This attribute specifies the number of password iterations required before a user is allowed to reuse a password.

minage

This attribute specifies the minimum number of weeks that must pass before a password can be changed. Since it is common for there to be situations where a password must be changed immediately (for example, to prevent the account from being compromised), we recommend that you do not use this attribute.

maxage

This attribute specifies the maximum number of weeks that can pass before a password must be changed. Consider using smaller values for privileged users, such as root and members of the system group. A smaller value limits the amount of time that an exposed or disclosed password can be used.

There is debate about whether or not a rigid password expiration policy is a good thing. A user suddenly confronted with an expired password does not have the luxury of considering their choice for a new one at length. More often than not, they will either pick something that is very close to the expired password or, worse yet, something trivial just so that they can get logged in to the system to do their work. The `pwdwarn` attribute (described later) causes AIX to start warning the user shortly before the password is set to expire. This permits the user the time needed to consider a strong replacement for their current password.

maxexpired

This attribute specifies the maximum number of weeks beyond `maxage` in which a password can be changed (after which administrative action will be required to effect the change). The root password is exempt.

minalpha

This attribute specifies the minimum number of alphabetic characters that the password must contain.

minother

This attribute specifies the minimum number of non-alphabetic characters that the password must contain. These (other) characters are defined as any

ASCII printable characters that are non-alphabetic and not national language code points.

minlen

This attribute specifies the minimum number of characters that the password must contain.

Note

The minimum length of a password is determined either by `minlen` or by the sum of 'minalpha + minother', whichever is greater. The sum of 'minalpha + minother' should never be greater than 8. If it is, then `minother` is reduced to '8 - minalpha'.

mindiff

This attribute specifies the minimum number of characters in the new password that must be different from the characters in the old password.

maxrepeats

This attribute specifies the maximum number of times a character can appear in the password.

pwdwarntime

This attribute (specified in days) causes AIX to start warning the user that their password is about to expire.

8.6.2 Password cracker utility

There are a number of good tools available that enable you to test the strength of passwords. One such tool is John the Ripper, which is covered in detail in Section 7.2, "John the Ripper" on page 119.

8.7 Step 6: Install additional security tools

Throughout this book, we cover a number of third-party security tools. Our intent is to give you a representative sample of the tools that are available to help you with the various areas of security. To recap, the tools covered in this book are:

- Check Point FireWall-1

- IBM Secureway Firewall
- SSH
- TCP Wrapper
- fping
- nmap
- SAINT
- PortSentry
- Tripwire
- PGP
- MD5
- John the Ripper

Note

A number of these tools are licensed products even though they are available for download on a “try before you buy” basis. If you plan to use them beyond the evaluation period, ensure that you have the proper licenses in place.

A brief summary of each is given here:

Firewalls are an essential security enforcement mechanism. Together with routers, they provide the first line of defense against attacks. In addition to protecting the network, they also log illegal access attempts, thus, providing a means of early warning for potential attacks. Only those services that are essential to the proper functioning of the organization should be allowed to cross the firewall. All others should be blocked.

SSH and TCP Wrapper are used to secure remote access. SSH is a secure replacement for less secure protocols, such as rsh. It provides encryption, authentication, and data integrity to ensure that remote hosts can communicate securely (even across the Internet). Different implementations of SSH provide different sets of tools. For example, some provide tools, such as sftp and scp, which are secure replacements for ftp and rcp. TCP Wrapper provides access control protection to the services that are under the control of the inetd daemon. This includes things, such as the ftpd and telnetd daemons, but can even include things, such as sendmail. There are two different TCP Wrapper access control lists. In one, you specify which servers are allowed access to which services. In the other, you specify which servers

are denied access to which services. Typically, you start by denying access to all services from all servers, and then grant access to individual services on a server-by-server basis. TCP Wrapper also provides logging of both successful and unsuccessful connection attempts.

The `fping` and `nmap` tools are network and port scanners. The `fping` tool does a ping sweep of an entire network or range of IP addresses by quickly sending out ICMP echo requests and tracking the return responses. It is a good tool for verifying that the topology of your network is what you think it is and that hosts that should respond to these types of requests actually do respond. The `nmap` tool is a port scanner. It is used to scan a large number of hosts looking for active services. It supports many different scan techniques, including *stealth scan*, which quietly scans a server in a way that makes it very hard to detect. These tools are invaluable for providing you with insight into what a would-be attacker sees when probing your network.

Important

Many organizations have very strict policies in place against running certain types of tools, such as sniffers and scanners, on a production network. Make sure you have the proper written authorization in place before running any of these tools on your systems.

The `PortSentry` tool provides protection against port scanners, such as `nmap`. `PortSentry` can automatically add the IP address of the suspected attacker to the TCP Wrapper `/etc/hosts.deny` file, thus, preventing access to any `inetd` controlled services that are protected by TCP Wrapper. `PortSentry` can also be configured to kill the route (by aliasing it against the loopback adapter) to the suspected attack server. Unfortunately, the AIX version of the tool does not currently support protection against stealth scans.

The `SAINT` tool is another port scanner. It is an enhanced version of the `SATAN` tool. In addition to scanning for active services, it also tests those services against a list of known vulnerabilities. `SAINT` produces a report of the potential vulnerabilities it found along with recommendations for fixing them. A similar, commercial tool is `NSA`, which comes with IBM Secureway Firewall. `NSA` provides much more additional functionality, but requires a valid license from IBM to run it. Both are comprehensive tools for overall security auditing.

It is helpful to view the collection of `fping`, `nmap`, `SAINT`, and `NSA` as a suite of tools for proactively verifying the security of your network. Each tool brings with it its own set of strengths and weaknesses, but the sum of the tools used together is much greater than the individual parts. Run in concert with each

other, these tools can cross-verify the results from the others. Be sure to keep the tools current with the latest versions, patches, and vulnerability lists.

Tripwire provides a means of verifying with a high degree of certainty that essential system files and directories have not been tampered with. It is a tool for assuring that system integrity has not been compromised. Attackers usually leave backdoors and Trojan horses after compromising a system so that they can return to it in the future. Tripwire protects against this threat by means of a secure database containing a snapshot of the system configuration. Without a tool, such as Tripwire, it is extremely difficult, tedious, and often impossible to ferret out the holes left behind by attackers.

PGP provides encryption services for users of your systems. It enables them to secure sensitive information from others (including root) and to communicate securely with others, locally or remotely. It is commonly used with email systems to provide secure authentication and data integrity for sensitive messages or files. Many Internet Web sites support it as a means of verifying that files downloaded from their site have not been tampered with.

MD5 is a tool for verifying data integrity. Like PGP, it is a commonly used method for ensuring that files downloaded from an Internet Web site have not been tampered with. This subject is further discussed in Section 7.4.1, "Ensuring the integrity of downloads" on page 136.

John the Ripper is a password cracking tool. Having a strong password policy is one of the most important defenses you can have within your network. With AIX, you can create and enforce strong password policy, such as requiring passwords to meet certain criteria, forcing passwords to be changed on a regular basis, and ensuring that passwords are not chosen from words available in the dictionary. However, even with a policy like this in place, it is still a good idea to use a tool, such as John the Ripper, to proactively search for weak passwords. Remember that an attacker who has gained access to a legitimate user account is very hard to distinguish from the real user. Finding and correcting weak passwords is one good way to combat against this.

8.8 Step 7: Monitor logs, audit trails, and system behavior

Routine and regular system monitoring is another important aspect of security. Through it, you are able to discover system breaches and attempted system breaches. The various elements of this type of monitoring can be roughly categorized as follows:

- Monitor system logs
- Enable auditing

- Monitor files and directories
- Monitor `cron` and `at` jobs

In addition to monitoring, you should also consider running regular scans with the security auditing tools described in the previous section. These tools help you ensure that your security posture remains unchanged. Often, a once secure network becomes insecure because someone introduces a change that inadvertently and unknowingly opens a security vulnerability. Running regular security audits greatly reduces the risks posed by such activities.

8.8.1 Monitor system logs

Regularly monitor the following logs for evidence of breaches or attempted breaches:

`/var/adm/sulog`

This file logs the use of the `su` command. It identifies the account that initiated the command, the account that was the target of the command, whether the command was successful or not, and the time and date when the command was run. This file is especially important if you have disabled remote login for root because it tracks the accounts that are being used to gain (or attempt to gain) root access along with the time when the access was performed.

`/var/adm/wtmp`

This file stores information about current and previous system logins and logouts. You access this file with the `last` command. The information is kept in this file only until the `acctcon1` and `acctcon2` commands are run.

`/etc/security/failedlogin`

This file captures all failed login attempts. You can access the information in this file by running: `who /etc/security/failedlogin | more`

`/etc/utmp`

This file stores information about the users who are currently logged in to the system. You can access the information in this file by running the `who` command.

8.8.2 Enable auditing

You can also set up system auditing to monitor security-related activities. A list of pre-defined audit events can be found in the `/etc/security/audit/events`

file. To streamline the audit and limit the amount of generated data, define just the events of interest to you. Once auditing is turned on, system activities are monitored and logs generated for the events that you have defined. Scan these logs regularly for evidence of any suspicious activities. For more information about setting up auditing, see "Setting up Auditing" in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

8.8.3 Monitor files and directories

First, begin by cleaning up unneeded files, which pose their own form of security risk. The `skulker` utility enables you to easily remove unwanted or obsolete files. It is typically run from `crontab` on a daily basis. Examples of candidate files include:

- files in the `/tmp` directory
- files older than a specified age
- `a.out` files
- `core` files
- `ed.hup` files

Next, set up routine monitoring and clean up of the following types of files (and directories):

Executable files owned by root that have the SUID and/or SGID bit set can pose a serious threat to security. These files execute with root authority regardless of who executes them. Unless the code is rock solid, it is possible to break out of it and run arbitrary commands as root. That is why it is so important to routinely monitor for these types of files. There are, however, AIX system programs that require the SUID and/or SGID bits to be set, and this is normal. Be sure to get a baseline listing of these files on a newly-installed system so that you know which files have a legitimate need for these settings.

To find files owned by root with either the SUID or SGID bit set:

```
# find / -perm -4000 -user 0 -ls
# find / -perm -2000 -user 0 -ls
```

You may also want to remove files with no owner, which are typically caused by removing an account without removing the files belonging to that account. In addition, you may also want to remove `.rhosts` files because they enable unauthenticated system access and `.netrc` files because they expose user IDs and passwords in plain text.

Important

Under some HACMP configurations, `/.rhosts` files are required for the proper functioning of an HACMP cluster on AIX. In those situations, do not remove the `/.rhosts` files, but, instead, ensure that the ownership is `root.system` and that the permissions are set to `600`.

To find files with no user:

```
# find / -nouser -ls
```

To find `.rhosts` files:

```
# find / -name .rhosts -ls
```

To find `.netrc` files:

```
# find / -name .netrc -ls
```

World-writable directories and files are also another potential area of security exposure. Again, you should get a baseline listing on a newly-installed system so that you can know which directories and files are normally required to be this way for the proper functioning of the system.

To find world-writable directories (monitor this list):

```
# find / -perm -0007 -type d -ls
```

To find world-writable files:

```
# find / -perm -2 -type f -ls
```

8.8.4 Monitor cron and at jobs

Both `cron` and `at` are favorite hunting grounds for attackers. These commands enable attackers to plant time bombs on the systems and be long gone well

before they explode. For this reason, you should be familiar with all of the jobs that are scheduled on your systems with `cron` or `at`.

To monitor all `cron` jobs:

```
# cronadm cron -l
```

To monitor all `at` jobs:

```
# cronadm at -l
```

Appendix A. NSA Scan Options

There are a number of scan options available in NSA (as documented in the `/etc/nsa/scannerdefs` file):

```
define default
  tcpports 21,22,23,25,109,110,111,139,143,512-514,6000
  udpports 69,111,137,161,177
  rpcsvcs *
  options no-user-login
end

define baseline
  tcpports 21,22,23,25,53,79,80,109,110,111,139,143,512-514,6000
  udpports 53,69,111,137,161,177
  rpcsvcs *
  options ftp-walk-tree
  options no-user-login
end

define medium
  tcpports 21,22,23,25,53,79,80,109,110,111,139,143,1080,6000
  udpports 53,69,111,137,161,177
  rpcsvcs *
  options tcp-seq-num
  options ftp-walk-tree
  options smtp-require-reject, smtp-mail-from, smtp-mail-rcpt
  options nntp-no-post
end

define standard
  tcpports 1-1023,6000-6063
  udpports 1-1023
  rpcsvcs *
  options tcp-seq-num
  options ftp-walk-tree
  options smtp-no-relay
  options smtp-require-reject, smtp-mail-from, smtp-mail-rcpt
  options nntp-no-post
end

define fulltcp
  tcpports 1-65535
  udpports 53,69,111,137,161,177
  rpcsvcs *
  options tcp-seq-num
  options ftp-walk-tree
  options smtp-no-relay
  options smtp-require-reject, smtp-mail-from, smtp-mail-rcpt
  options nntp-no-post
end
```

```
define complete
  tcpports 1-65535
  udpports 1-65535
  rpcsvcs *
  options tcp-seq-num
  options ftp-walk-tree, ftp-no-wdir, ftp-no-mkdir
  options smtp-mail-from, smtp-mail-rcpt
  options nntp-no-post
  options smtp-no-relay
end

define firewall
  tcpports 1-65535
  udpports 1-65535
  rpcsvcs *
  options ip-source-route, tcp-seq-num, ip-options
  options ftp-walk-tree, ftp-no-wdir, ftp-no-mkdir
  options smtp-require-reject, smtp-mail-from, smtp-mail-rcpt
  options smtp-no-relay
  options nntp-no-post
end
```

Appendix B. Script used to scan a network with fping

This script can be used with the fping tool to scan a whole network with netmask 255.255.255.0 as mentioned in Section 6.1.2, "Using fping" on page 86.

```

#!/bin/ksh
#
# To scan a network with fffffff0 netmask
#
# Declare datafile to store list of hosts to scan
datafile=/tmp/fping.data

tput clear; echo
if [ $# -ne 1 ]; then
    echo "Usage : $0 {Network address}"
    echo "Example : $0 172.16.30.0\n"
    exit 0
fi

ID1=`echo $1 | awk -F . '{print $1}'`
ID2=`echo $1 | awk -F . '{print $2}'`
ID3=`echo $1 | awk -F . '{print $3}'`
ID4=`echo $1 | awk -F . '{print $4}'`

# If last octet is the null string or 0, scan 255 addresses
if [[ $ID4 = "" ]] || [[ $ID4 -eq 0 ]]; then
    tput clear
    echo "Please enter filename to output results"
    read outfile
    echo "Output file is $outfile\n"
    # Remove $datafile if it already exists.
    if [ -f $datafile ]; then
        echo "The default file to store list of hosts to scan is $datafile"
        echo "There is already a file named $datafile"
        echo "Do you want to remove it?[y(es)/n(o)]"
        read removeoutfile
        case "$removeoutfile" in
            y* | Y*)
                echo "Removing file\n"
                rm $datafile;;
            *)
                echo "\nRerun program after renaming/removing that"
                exit 0;;
        esac
    fi
    echo "Obtaining list of hosts to scan and storing the list into $datafile"
    y=0
    while [ "$y" -le 255 ]; do
        echo "$ID1.$ID2.$ID3.$y" >> $datafile
        y=`expr $y + 1`
    done
    echo "\nScanning hosts now. Please wait for prompt to return."
    cat $datafile | /usr/local/bin/fping -sA > $outfile 2>&1
    echo "View output file $outfile for results"
else
    # If last octet is a host IP, then scan the host only
    /usr/local/bin/fping -sA $1
fi
exit 0

```

Appendix C. Script to merge the AIX passwd files

This script merges the `/etc/passwd` and `/etc/security/passwd` files. It was obtained from the Crack package (available at the Bull site). The John the Ripper tool requires a single password file, as mentioned in Section 7.2, "John the Ripper" on page 119.

```
#!/bin/sh
###
# This program was written by and is copyright Alec Muffett 1991,
# 1992, 1993, 1994, 1995, and 1996, and is provided as part of the
# Crack v5.0 Password Cracking package.
#
# The copyright holder disclaims all responsibility or liability with
# respect to its usage or its effect upon hardware or computer
# systems, and maintains copyright as set out in the "LICENCE"
# document which accompanies distributions of Crack v5.0 and upwards.
###

SHADOW=/etc/security/passwd
PASSWD=/etc/passwd

(
  awk '
/^ [a-zA-Z0-9]+: / {
  curruser = $1;
  next;
}
$1 == "password" {
  print "STAG:" curruser $3;
}' < $SHADOW

  sed -e 's/^PTAG:/' < $PASSWD
) |
awk -F: '
BEGIN {
  OFS=":";
}
$1 == "STAG" {
  pw[$2] = $3;
  next;
}
$1 == "PTAG" {
  $3 = pw[$2];
  print $0;
}' |
sed -e 's/^PTAG:/'
```

Appendix D. Special notices

This publication is intended to help an RS/6000 AIX or SP/Cluster Specialist who wants to implement additional security tools other than those provided by IBM AIX 4.3.3 or PSSP 3.2. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM AIX and PSSP software products. See the PUBLICATIONS section of the IBM Programming Announcement for IBM products mentioned in this redbook for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.


Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee

that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

| | |
|---|------------------|
| e (logo)®  | IBM |
| Netfinity | Redbooks |
| Redbooks Logo  | RISC System/6000 |
| RS/6000 | SAA |
| SecureWay | SP |
| SP1 | System/390 |
| TCS | |

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries

licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Lotus Notes is a registered trademark of Lotus Development Corporation

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 221.

- *A Comprehensive Guide to Virtual Private Networks, Volume III*, SG24-5309
- *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855
- *AIX Version 4.3 Differences Guide*, SG24-2014
- *AIX 4.3 Elements of Security Effective and Efficient Implementation*, SG24-5962
- *Exploiting RS/6000 SP Security: Keeping It Safe*, SG24-5521
- *Safe Surfing: How to Build a Secure WWW Connection*, SG24-4564
- *TCP/IP Tutorial and Technical Overview*, GG24-3376

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

| CD-ROM Title | Collection Kit Number |
|--|-----------------------|
| IBM System/390 Redbooks Collection | SK2T-2177 |
| IBM Networking Redbooks Collection | SK2T-6022 |
| IBM Transaction Processing and Data Management Redbooks Collection | SK2T-8038 |
| IBM Lotus Redbooks Collection | SK2T-8039 |
| Tivoli Redbooks Collection | SK2T-8044 |
| IBM AS/400 Redbooks Collection | SK2T-2849 |
| IBM Netfinity Hardware and Software Redbooks Collection | SK2T-8046 |
| IBM RS/6000 Redbooks Collection | SK2T-8043 |
| IBM Application Development Redbooks Collection | SK2T-8037 |
| IBM Enterprise Storage and Systems Management Solutions | SK3T-3694 |

E.3 Other resources

These publications are also relevant as further information sources:

- *Building Internet Firewalls*, ISBN 1-5659-2124-0, by D. Brent Chapman, Elizabeth D. Zwicky, and Deborah Russell
- *Firewalls and Internet Security: Repelling the Wily Hacker*, ISBN 0-2016-3357-4
- *Hacking Exposed: Network Security Secrets and Solutions*, ISBN 0-0721-2748-1
- *Information Warfare and Security*, ISBN 0-2014-3303-6, by Dorothy E. Denning
- *Masters of Deception: The Gang That Ruled Cyberspace*, ISBN 0-0609-2694-5, by Michele Slatalla and Joshua Quittner,
- *Practical Unix and Internet Security*, ISBN 1-5659-2148-8, by Simson Garfinkel and Gene Spafford
- *Sendmail*, ISBN 1-5659-2222-0
- *The Art of War*, ISBN 0-3852-9216-3, by Sun Tzu
- *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*, ISBN 0-6717-2688-9, by Clifford Stoll
- *The X Windows System Administrator's Guide, Volume 8*, ISBN 0-9371-7583-8
- *Www.security How to Build a Secure World Wide Web Connection*, ISBN 0-1361-2409-7

E.4 Referenced Web sites

The following Web sites are also relevant as further information sources:

- <http://www.ssh.org>
- <http://www.bull.com>
- <http://www.postfix.org>
- <http://www.cerias.purdue.edu/coast/satan.html>
- <http://www.ssh.org>
- <http://all.net>
- <http://wuarhive.wustl.edu/packages/security/TAMU/>
- <http://www.faqs.org/faqs/kerberos-faq/general/>

- **IBM Redbooks:**
<http://www.redbooks.ibm.com/>
- **Designing an Authentication System: a Dialogue in Four Scenes:**
<http://web.mit.edu/kerberos/www/dialogue.html/>
- **Trusted Computing System Evaluation Criteria (TCSEC) Rainbow Series Library:**
<http://www.radium.ncsc.mil/tprep/library/rainbow/index.html/>
- **Information Technology Security Evaluation Criteria (ITSEC) Assurance Level Criteria:**
<http://www.itsec.gov.uk/>
- **Wietse Venema's (author of tcp_wrapper) tools and papers:**
<http://sneezy.ice.ntnu.edu.tw/os/unix/security/>
- **The Admin guide to cracking by Dan Farmer and Wietse Venema:**
http://www.cerias.purdue.edu/coast/satan-html/docs/admin_guide_to_cracking.html/
- **The Internet Assigned Numbers Authority:**
<http://www.iana.org/>
- **Large archive of SMIT installable freeware:**
<http://www.bull.de/pub>
- **Secure Shell homepage:**
<http://www.ssh.org>
- **Computer Emergency Response Team:**
<http://www.cert.org>
- **IBM emergency response team:**
<http://www.ers.ibm.com>
- **Postfix homepage, replacement for sendmail:**
<http://www.postfix.org>
- **Fred Cohen's page, programmer of the Deception Toolkit:**
<http://all.net>
- **Chaos Computer Club (partly German):**
<http://www.ccc.de>
- **Security information, large download area with hacker tools and security tools:**
<http://packetstorm.securify.com>
- **This is a *security* site, not a hacker site:**
<http://www.attrition.org>

- **Top 50 security sites:**

- <http://www.cyberarmy.com/t-50/index.shtml>
- <http://www.ers.ibm.com/>
- <http://www.sanstore.org/>
- www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm
- www.checkpoint.com
- www.phoneboy.com/fw1
- www.checkpoint.com/support/technical/documents/index.html
- www.enteract.com/~lspitz/fwtable.txt
- www.fc.net/phrack/files/p48/p48-13.html
- www.enteract.com/~lspitz/intrusion.htm
- www.infoworld.com/supplements/99poy_win/99poy_s.html
- <http://www-3.ibm.com/security/>
- <http://www-4.ibm.com/software/security/firewall/>
- www-4.ibm.com/software/network/dispatcher/
- www.mimesweeper.integralis.com
- www.telemate.net
- <http://www-4.ibm.com/software/security/firewall/about/>
- www.onsight.com/faq/ssh/ssh-faq.html
- <ftp://ftp.ssh.com/pub/ssh/>
- <http://www.ssh.com/ssh/download.html>
- <http://www.openssh.com>
- <http://www-frec.bull.fr>
- <http://www.f-secure.com/products/ssh/>
- <http://www.vandyke.com/products/securecrt/>
- <http://www.zip.com.au/~roca/ttssh.html>
- <ftp://ftp.porcupine.org/pub/security/index.html>
- <http://www.insecure.org/nmap>
- <http://www.wwdsi.com/saint/index.html>
- <http://www.psionic.com/abacus/>
- <http://www.psionic.com/abacus/portsentry/>

- <http://www.psionic.com/abacus/logcheck/>
- <http://www.tripwiresecurity.com/>
- <http://www.tripwiresecurity.com/products/index.cfm>
- <http://www.bull.de/pub/out/>
- <http://www.pgpi.org/doc/faq/pgpi/en/#International>
- <http://www.pgp.com>
- <http://www.pgpi.com>
- <http://www.nai.com>
- <http://www.pgpinational.com>
- www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/topnav/topnav.htm
- <http://www.isc.org/products/BIND/>
- www.dictionary.com
- <http://www.sendmail.org>
- <http://www.sendmail.com>
- <http://www.sendmail.org/faq>
- <http://www.sendmail.org/antispam.html>

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

| | e-mail address |
|----------------------------|---|
| In United States or Canada | pubscan@us.ibm.com |
| Outside North America | Contact information is in the "How to Order" section at this site: http://www.elink.ibm.com/pbl/pbl |

- **Telephone Orders**

| | |
|---------------------------|--|
| United States (toll free) | 1-800-879-2755 |
| Canada (toll free) | 1-800-IBM-4YOU |
| Outside North America | Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibm.com/pbl/pbl |

- **Fax Orders**

| | |
|---------------------------|--|
| United States (toll free) | 1-800-445-9269 |
| Canada | 1-403-267-4455 |
| Outside North America | Fax phone number is in the "How to Order" section at this site: http://www.elink.ibm.com/pbl/pbl |

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

Index

Symbols

!tty0 191
145
\$FWDIR/conf/fwauthd.conf 43
\$FWDIR/conf/fwopsec.conf 43
\$FWDIR/log 29
\$HOME/.deadletter 178
\$HOME/.pgp/pubbring.prk 129
\$HOME/.pgp/secring.skr 129
\$HOME/.rhosts 70
\$HOME/.shosts 70
\$HOME/.ssh/authorized_keys 70, 71
\$HOME/.ssh/authorized_keys2 74
\$HOME/.ssh/config 70
\$HOME/.ssh/id_dsa 71, 74
\$HOME/.ssh/id_dsa.pub 74
\$HOME/.ssh/identity 70
\$HOME/.ssh/known_hosts 68
\$HOME/.ssh/known_hosts2 72
\$JOHNDIR/john.ini 122
\$JOHNDIR/john.pot 123
(^X^R) 191
.deadletter 178
.forward 177
.forward files 179
.netrc 202
.rhosts 60, 202
.shosts 68
./rhosts 203
/etc/exports 168, 175
/etc/hosts 153, 168
/etc/hosts.allow 79
/etc/hosts.deny 79, 199
/etc/hosts.equiv 60, 70
/etc/inetd.conf 43, 148
/etc/inittab 28, 69, 142, 174
/etc/krb.conf 181
/etc/krb.realms 181
/etc/named.boot (BIND4) 158, 160
/etc/named.conf (BIND8) 158, 159, 161
/etc/nsa/license.nsa 50
/etc/nsa/scannerdefs 50
/etc/openssh/ssh_host_key 69
/etc/openssh/ssh_host_key.pub 72
/etc/openssh/sshd_config 68, 69, 72
/etc/passwd 122, 168
/etc/profile 68, 194
/etc/rc.dt 144
/etc/rc.dt, protecting with Tripwire 144
/etc/rc.local 28
/etc/rc.net 28
/etc/rc.nfs 174, 175
/etc/rc.openssh 69
/etc/rc.tcpip 28, 68, 144, 157
/etc/SDR_dest_info 181
/etc/security/audit/events 201
/etc/security/failedlogin 201
/etc/security/login.cfg 191, 193
/etc/security/sysck.cfg 114
/etc/security/user 190, 193, 194
/etc/sendmail.cf 175, 177, 178
/etc/sendmail.st 178
/etc/shosts.equiv 68, 70
/etc/snmpd.conf 180
/etc/ssh/known_hosts 68
/etc/tcp.clean 145
/etc/tftpaccess.ctl 181
/etc/utmp 201
/spdata 174
/spdata/sys1/install/pssplpp 174
/tftpboot 181
/tmp 202
/usr/adm/tcheck/databases/ 118
/usr/adm/tcheck/databases/tw.db_ 112
/usr/adm/tcheck/tw.config 112
/usr/local/bin 68
/usr/local/bin/john 121
/usr/local/bin/john-1.6/password.lst 121
/usr/local/lib/john-1.6/words.list.sort 122
/usr/local/lib/tripwire-1.2/README.CheckConfig 115
/usr/local/man 68
/usr/lpp/FireWall-1/log 29
/usr/lpp/nsauditor/doc/ 50
/usr/lpp/spp 181
/usr/samples/tcpip/named-booconf.pl 158
/usr/sbin/named 157
/usr/sbin/named4 157
/usr/sbin/named8 157
/usr/sbin/sendmail 178
/usr/security/bin/tripwire 112
/usr/share/dict/words 121
/usr/shared/dict/words 195

/usr/vac/bin/cc 170
/var/adm/sulog 191, 192, 201
/var/adm/wtmp 201
/var/spool/mqueue 178

Numerics

022 192
077 192
3DES 64, 72
3-way TCP/IP handshake 187
80/20 rule 3

A

a.out 202
access control, TFTP 181
access control, X11 182
account expiration time, defining 192
account login times, defining 192
account policies, strong 189
ACK packet 187
active ftp 185
additional security tools, installing 197
ADSM
 See TSM
ADSTAR Distributed Storage Manager (ADSM)
 See TSM
advertising name servers 164
advisory locking in UNIX 174
AIX
 /etc/inittab 142
 network services 141
 securing 139
AIX 3.2.x 157
AIX 4.1.x 157
AIX 4.2.x 157
AIX 4.3.3 (with bos.net.tcp.server 4.3.3.13) 157
AIX 4.3.x 157
AIX account attributes
 expires 192
 loginretries 190
 logintimes 192
 rlogin 190, 191
 tpath 190, 191
 ttys 190, 191
 umask 190, 191
AIX account attributes
 registry 190, 191
AIX account roles 192

AIX herald, disabling 193
AIX password settings
 dictionlist 195
 histexpire 195
 histsize 195, 196
 maxage 195, 196
 maxexpired 195, 196
 maxrepeats 195, 197
 minage 195, 196
 minalpha 195, 196
 mindiff 195, 197
 minlen 195, 197
 minother 195, 196
 pwdwarntime 195, 197
AIX versions of sendmail 175
AIX, DNS versions 157
all-known-IP list 188
AllowGroups 75
allow-query substatement 163
allow-query, example 163
allow-transfer substatement 159
allow-transfer, master example 160
allow-transfer, options example 160
allow-transfer, slave example 160
allow-update substatement 161
allow-update, example 161
AllowUsers 75
alternatives to disabling DNS recursion 162
alternatives to SSH 77
always 191
anonymous FTP site for h2n 158
APAR IX86922 188
application layer (proxy) firewalls 15, 16
architecture of security framework 6
assessment, threat 4
at 201, 203
attack, SYN 37, 38
attacks
 buffer overflow 153
attacks, buffer overflow 152
attacks, denial of service 152
attacks, smurf 187
attributes, other 193
audit logs 200
AUTH_DES 173
AUTH_SYS 168
AUTH_UNIX 168, 173
authentication 62, 71
authentication methods, RPC 168

authoritative 163
authwarnings 176
autoconf6 147
automatic socksification support in AIX 4.3.3 186
automounted home directories 173

B

banner replacement, ftp 184
banner, ftp 184
banner, sendmail 176
Basic Networking Utilities
 See BNU

bastion hosts 177
bcastping 187
Berkeley Internet Name Daemon
 See BIND

biff 151

BIND 147, 153

 4.8.3 157
 4.9 159
 4.9 xfrnets directive 159
 4.9.3, ipv4 157
 8.1.2, ipv6 157
 8.2 155
 8.2.1 156
 8.2.2 (patch 5), ipv6 157
 homepage 153
 symbolic link 157

BIND4 153, 157

 BIND8 conversion script 158
 configuration file 158
 converting to BIND8 158
 no longer supported 153
 options no-fetch-glue 163
 options no-recursion 162
 secure_zone record 163

BIND8 153, 157

 allow-transfer substatement 159
 allow-update substatement 161
 configuration file 158
 converting to from /etc/hosts 158
 dynamic update capability 161
 options fetch-glue no 163
 options recursion no 162

biod 174

biod, defined 174

blackholing, defined 156

Blowfish 64

BNU 151
bogus DNS update 155
bootp 146
bootp protocol, DHCP extension 146
bootpd 151
bootps 149, 151, 181
bos.data 195
bos.net.ipsec 180, 183
bos.net.ipsec.rte 15
bos.net.nfs.client 174
bos.net.tcp.client 157, 165
bos.net.tcp.server 4.3.3.13 157
bos.txt.spell 195
boundary routers 2
broadcast protection 187
buffer overflow attacks 152, 153, 174
building a security framework 2

C

cache update, restricting 163
cache, DNS 154
CAST 125
CDE 144, 153
CDE and Tripwire 144
CDE desktop 183
CDE startup, disabling 144
CDE, uninstalling 144
chargen 152
Check Point FireWall-1 14, 24, 25, 26, 37, 38, 49, 197
chmod 0000 /usr/sbin/dpid2 148
chmod 4511 /usr/sbin/sendmail 178
chrooted 179
circuit level firewalls 15
clean_partial_conns 187
cleanup rule 34
clear text 149
clock source, reliable 147
closing vulnerabilities during system startup 28
cluster, HACMP 203
cmsd 153
Command
 lspp 68
 opensshd 72
Commands
 at 201, 203
 biff 151
 bootps 149

chmod 0000 /usr/sbin/dpid2 148
 chmod 4511 /usr/sbin/sendmail 178
 cron 201, 203
 cronadm at -l 204
 cronadm cron -l 204
 delset 1 185
 dspcat -g 185
 egrep 122
 exportfs -au 175
 exportfs -u 175
 ftp 148
 fw 36
 gencat 185
 grpck 190
 kloginl 149
 kshell 149
 last 201
 lsof 144
 make 170
 netstat 44, 152
 netstat -af inet 144
 no 28, 186
 nsa 50
 nslookup 159, 165
 on 152
 opensshd 69
 pgpe 132
 pgpk 127
 ps 152
 pwdck 190
 rcp 151
 refresh -s inetd 149
 rftp 185, 186
 rlogin 148, 151
 rpcinfo 168
 rsh 148, 151
 rtelnet 186
 rtn 186
 scp 148
 sendmail -v -bi 179
 sftp 148
 showmount 170
 smitty failed_logins 191
 spray 152
 ssh 148
 ssh-keygen 71, 73, 76
 su 201
 talk 152
 tcp.clean 145
 tcpdchk 80
 tcpdmatch 81
 telnet 148
 tftp 149
 usrck 190
 who 201
 who /etc/security/failedlogin | more 201
 write 144
 xauth 182
 xhost 182
 xhost + 182, 183
 xlock 183
 xterm 183
 commands to find root SUID/SGID files 202
 commands to remove default users and groups 189
 comment out vs. delete 145
 commercial version of sendmail 178
 Commnads
 xhost - 183
 Common Desktop Environment
 See CDE
 communications daemons, disabling 144
 community names, SNMP 180
 community names, unique SNMP 148
 compartmentalized approach 23
 compartmentalized firewall environment design 21
 compartmentalized firewalls 35
 Compartments 21
 complementary software for FireWall-1 26
 complimentary software for IBM Secureway Firewall 48
 comstat 151
 configuration file for BIND4 158
 configuration file for BIND8 158
 configuring and using SSH1 69
 configuring and using SSH2 72
 configuring and using Tripwire 112
 configuring DNS 153
 configuring John the Ripper 121
 configuring remaining services 153
 configuring TCP Wrapper 79
 configuring Tripwire 112, 114
 connections, viewing 36
 converting from /etc/hosts to BIND8 158
 converting from BIND4 to BIND8 158
 COPS 119
 core 202
 counterfeit update requests 155

covering tracks 5
crack 197
crack utility 197
cracking tools 194
creating a useful rulebase 34
cron 201, 203
cronadm at -l 204
cronadm cron -l 204
cross-verify 200
cryptographic authentication in DNS 156

D

daemons 144
 autoconf6 147
 biod 174
 bootpd 151
 communication 144
 comstat 151
 DHCP 146
 dhcpcd 146
 dhcprd 146
 dhcpsd 146
 DNS 147
 dpid2 144, 148
 email 153
 fingerd 151
 ftpd 148, 184
 gated 147
 hats 188
 httpdlite 143
 imapd 153
 imnss 143
 imqss 143
 inetd 144, 148, 184, 198
 info search 143
 IPv6 147
 krlogind 151
 krshd 151
 lpd 147
 mrouted 148
 multicasting 148
 named 147, 157
 ndnpd-router 147
 ndpd-host 147
 network management 148
 nfsd 174
 pcnfsd 152
 piobe 143
 pop3d 153
 portmap 144, 168
 print 143, 147
 qdaemon 143
 rexcd 152
 rexcfd 151
 rlogind 148, 151
 routed 147
 routing 147
 rpc.lockd 174
 rpc.mountd 174
 rpc.statd 174
 rshd 148, 151
 rstatd 151
 ruserd 151
 rwalld 151
 rwhod 148
 sendmail 144, 175, 176
 snmpd 144, 148
 sprayd 152
 syslog 144
 talkd 152
 telnetd 148
 tftpd 151
 time 147
 time synchronization 147
 timed 147
 uprintfd 144
 uucpd 151
 writesrv 144
 xntpd 147
daemons started by default in /etc/rc.tcpip
 dpid2 144
 inetd 144
 portmap 144
 sendmail 144
 snmpd 144
 syslog 144
data integrity 65
database managers started in /etc/inittab 142
database, Tripwire 200
daytime 152
DCE 147
DCE user management 173
DCE/DFS 147
default accounts 189
default accounts, removing unnecessary ones 189
default daemons started by TCP/IP 144
default groups

- imnadm 189
- printq 189
- uucp 189
- default stanza 190, 193, 194
- default users
 - guest 189
 - imnadm 189
 - lpd 189
 - nuucp 189
 - uucp 189
- default webserver 143
- DefaultUser 177
- defining account expiration time 192
- defining account login times 192
- delete vs. comment out 145
- delset 1 185
- demilitarized zone
 - See* DMZ
- denial of service attack 37
- denial of service attacks 152, 187
- DenyUsers 75
- DES 64
- DES-based, public-key, private-key encryption 173
- description of entries in /etc/inetd.conf 150
- DFS 147
- DHCP 146, 161
- DHCP client daemon, dhcpcd 147
- DHCP daemons, disabling 146
- DHCP relay daemon, dhcprd 147
- DHCP server daemon, dhcpsd 147
- DHCP, extension of bootp protocol 146
- dhcpcd 146
- dhcprd 146
- dhcpsd 146
- dictionary, standard UNIX 194, 195
- dictionlist 195
- difference between SSH1 and SSH2 62
- digital signatures 156
- directed_broadcast 188
- directories, world-writable 203
- disabling CDE startup 144
- disabling communications daemons 144
- disabling DHCP daemons 146
- disabling DNS daemons 147
- disabling info search daemons 143
- disabling IPv6 daemons 147
- disabling multicasting daemons 148
- disabling network management daemons 148
- disabling print daemons 143, 147
- disabling routing daemons 147
- disabling rpc.lockd & rpc.statd 175
- disabling services
 - dt 144
 - httpd-lite 143
 - imnss 143
 - imqss 143
 - piobe 143
 - qdaemon 143
 - uprintfd 144
 - writesrv 144
- disabling TCP/IP daemons
 - autoconf6 147
 - dhcpcd 146
 - dhcprd 146
 - dhcpsd 146
 - dpid2 148
 - gated 147
 - lpd 147
 - mrouted 148
 - named 147
 - ndnpd-router 147
 - ndpd-host 147
 - routed 147
 - rwhod 148
 - snmpd 148
 - timed 147
 - xntpd 147
- disabling the AIX herald 193
- disabling time daemons 147
- disabling VRFY & EXPN 175
- discard 153
- Distributed Computing Environment
 - See* DCE
- Distributed File System
 - See* DFS
- DMZ 3, 7, 19, 20, 35, 168, 177, 189
 - public servers in
- DMZ, private network in 2
- DNS 147, 153
 - advertising name servers 164
 - allow-query substatement 163
 - alternatives to disabling recursion 162
 - cache 154
 - cache update, restricting 163
 - checking version 157
 - configuring 153
 - cryptographically authenticated name servers 156

- daemons, disabling 147
- debugging with nslookup 168
- determining version 157
- DNS spoofing compared to IP spoofing 156
- double lookup 156
- dynamic updates, restricting 161
- fetch-glue no option 163
- finding version 157
- glue fetching, defined 163
- internal resolution name servers 164
- IP address resolution steps 154
- IP spoofing compared to DNS spoofing 156
- nslookup 165
- principles 154
- protecting against spoofing exploit 156
- queries, restricting 162
- recursion disabled 162
- recursion, alternatives to disabling 162
- recursion, defined 161
- recursion, restricting 161
- referrals 162
- resolvers 156
- restrict cache update 163
- restrict dynamic updates 161
- restrict recursion 161
- restrict zone transfers 159
- restricting queries 162
- securing on AIX 159
- Security Extensions (DNSSEC), RFC 2065 156
- split configuration 164
- spoofing 154, 155
- spoofing exploit, explained 155
- spoofing exploit, protecting against 156
- spoofing, new and improved 155
- spoofing, old and tired 154
- Start Of Authority (SOA) record 166
- TXT (txt) records 163
- update, bogus 155
- versions on AIX 157
- with recursion disabled 162
- zone transfers, restricting 159

- docsearch engine 143
- Domain Name System
 - See DNS
- dpid2 144, 148
- dpid2 security exposure 148
- DSA 62, 64
- dspcat -g 185
- dt 144
- dt entry in /etc/inittab 144
- dtspc 153
- dynamic DNS, protocol extension 161
- Dynamic Host Control Protocol
 - See DHCP
- dynamic IP address assignment 146
- dynamic packet filter firewalls 15, 16
- dynamic routes 188
- dynamic routing 147
- dynamic routing daemons 188
- dynamic update capability in BIND8 161
- dynamic updates, restricting 161
- dynamic vs. static routes 147, 188

E

- echo 152
- echo packets, ICMP 187
- ed.hup 202
- egrep 122
- elements of security monitoring 200
- email daemons 153
- emergency response services (ERS) 10
- enable auditing 200, 201
- enabling other defense mechanisms 37
- encryption 64
- engine, docsearch 143
- entries in /etc/inetd.conf 150
- enumeration 5
- environment variable TIMEOUT 194
- environment variable TMOUOT 194
- etc/security/passwd 122
- exec 151
- expires 192
- EXPN 175
- EXPN, defined 176
- exportfs -au 175
- exportfs -u 175
- external firewalls 2
- external network 19

F

- Fast Mode TCP 42
- favorite haunting ground 203
- fetch-glue no (BIND8) option 163
- fetch-glue no option in DNS 163
- File Transfer Protocol
 - See FTP
- Files

\$FWDIR/conf/fwauthd.conf 43
 \$FWDIR/conf/fwopsec.conf 43
 \$HOME/.deadletter 178
 \$HOME/.pgp/pubring.prk 129
 \$HOME/.pgp/secring.skr 129
 \$HOME/.rhosts 70
 \$HOME/.shosts 70
 \$HOME/.ssh/authorized_keys 70, 71
 \$HOME/.ssh/authorized_keys2 74
 \$HOME/.ssh/config 70
 \$HOME/.ssh/id_dsa 71, 74
 \$HOME/.ssh/id_dsa.pub 74
 \$HOME/.ssh/identity 70
 \$HOME/.ssh/known_hosts 68
 \$HOME/.ssh/known_hosts2 72
 \$JOHNDIR/john.ini 122
 \$JOHNDIR/john.pot 123
 .deadletter 178
 .forward 177, 179
 .netrc 202
 .rhosts 60, 202
 .shosts 68
 /.rhosts 203
 /etc/exports 168, 175
 /etc/hosts 153, 168
 /etc/hosts.allow 79
 /etc/hosts.deny 79, 199
 /etc/hosts.equiv 60, 70
 /etc/inetd.conf 43, 148
 /etc/inittab 28, 69, 142, 174
 /etc/named.boot (BIND4) 158, 160
 /etc/named.conf (BIND8) 158, 159, 161
 /etc/nsa/license.nsa 50
 /etc/nsa/scannerdefs 50
 /etc/openssh/ssh_host_key 69
 /etc/openssh/ssh_host_key.pub 72
 /etc/openssh/sshd_config 68, 69, 72
 /etc/passwd 122, 168
 /etc/profile 194
 /etc/rc.dt 144
 /etc/rc.local 28
 /etc/rc.net 28
 /etc/rc.nfs 174, 175
 /etc/rc.openssh 69
 /etc/rc.tcpip 28, 68, 144, 157
 /etc/security/audit/events 201
 /etc/security/failedlogin 201
 /etc/security/login.cfg 191, 193
 /etc/security/sysck.cfg 114
 /etc/security/user 190, 193, 194
 /etc/sendmail.cf 175, 177, 178
 /etc/sendmail.st 178
 /etc/shosts.equiv 68, 70
 /etc/snmpd.conf 180
 /etc/ssh/known_hosts 68
 /etc/tcp.clean 145
 /etc/tftpaccess.ctl 181
 /etc/utmp 201
 /usr/adm/tcheck/databases/* 118
 /usr/adm/tcheck/databases/tw.db_ 112
 /usr/adm/tcheck/tw.config 112
 /usr/local/bin/john-1.6/password.lst 121
 /usr/local/lib/john-1.6/words.list.sort 122
 /usr/local/lib/tripwire-1.2/README.CheckCon-
 fig 115
 /usr/lpp/FireWall-1/log/* 29
 /usr/lpp/nsauditor/doc/* 50
 /usr/samples/tcpip/named-booconf.pl 158
 /usr/sbin/named 157
 /usr/sbin/named4 157
 /usr/sbin/named8 157
 /usr/sbin/sendmail 178
 /usr/security/bin/tripwire 112
 /usr/share/dict/words 121
 /usr/shared/dict/words 195
 /usr/vac/bin/cc 170
 /var/adm/sulog 191, 192, 201
 /var/adm/wtmp 201
 /var/spool/mqueue 178
 a.out 202
 cc 170
 core 202
 ed.hup 202
 etc/security/passwd 122
 events 201
 exports 168, 175
 failedlogin 201
 h2n (BIND8) 158
 hosts 153, 168
 hosts.deny 199
 include 177
 inetd.conf 148
 inittab 142, 174
 login.cfg 191, 193
 mqueue 178
 named 157
 named.boot (BIND4) 158, 160
 named.conf (BIND8) 158, 159, 161

- named4 157
- named8 157
- named-booconf.pl 158
- named-xfer 158
- nslookup 159, 165
- nsupdate 158
- passwd 168
- profile 194
- rc.dt 144
- rc.nfs 174, 175
- rc.tcpip 157
- sendmail 178
- sendmail.cf 175, 177, 178
- sendmail.st 178
- snmpd.conf 180
- sulog 191, 192, 201
- tcp.clean 145
- tftpaccess.ctl 181
- tripwire.list.sort 114
- tw.config 115
- user 190, 193, 194
- utmp 201
- words 195
- wtmp 201
- files, world-writable 203
- filesets
 - bos.data 195
 - bos.net.ipsec 180, 183
 - bos.net.ipsec.rte 15
 - bos.net.nfs.client 174
 - bos.net.tcp.client 157, 165
 - bos.net.tcp.server 4.3.3.13 157
 - bos.txt.spell 195
 - freeware.egd.rte 65
 - freeware.openssl.rte 65
 - freeware.zlib.rte 65
 - IMNSearch.rte.httplite 143
 - nsaauditor.base 49
 - X11.apps.clients 182
- finding .netrc files 203
- finding .rhosts files 203
- finding files with no user 203
- finding root SUID/SGID files 202
- finding world-writable directories 203
- finding world-writable files 203
- finger 151
- fingerd 151
- firewall
 - design 17
 - firewall design 17
 - firewall hardening 49
 - Firewall Toolkit 179
 - FireWall-1 197
 - FireWall-1 features 25
 - FireWall-1, managing logs 29
 - FireWall-1, securing default configurations 29
 - firewalls 13, 198
 - application layer (proxy) 15, 16
 - circuit level 15
 - comparison between different types 16
 - compartmentalized 35
 - compartmentalized environment design 21
 - default configurations 14
 - dynamic packet filter 15, 16
 - misconceptions 13
 - on AIX 24
 - securing 24
 - static packet filter 15
 - types 15
 - firewalls on AIX 24
 - firewalls, external 2
 - firewalls, internal 2
 - firewalls, misconceptions 13
 - firewalls, securing 24
 - firewalls, compartmentalized 35
 - footprinting 4
 - fping 198, 199
 - framework, security 1
 - freeware.egd.rte 65
 - freeware.openssl.rte 65
 - freeware.zlib.rte 65
 - F-Secure SSH 62
 - FTP 184
 - ftp 148, 198
 - //ftp.porcupine.org/pub/security/index.html 78
 - //ftp.ssh.com/pub/ssh/ 61
 - ftp banner 184
 - ftp banner replacement 184
 - ftp client, socksified 185
 - ftp protocol 184
 - ftp, active 185
 - ftp, passive 185
 - ftpd 148, 184
 - fw 36

G

- gated 147

gencat 185
gentle satirical imitation 154
glue fetching 164
glue fetching, defined 163
glue fetching, turning it off 163
goaway 176
grpck 190
guest 189
gzip 170

H

h2n (BIND8) 158
h2n, anonymous FTP site for 158
HACMP 142, 203
HACMP and /.rhosts 203
handshake, TCP 3-way 187
hardening, firewalls 49
hats topology daemon 188
herald 193
High Availability Cluster MultiProcessing
 See HACMP
histexpire 195
histsize 195, 196
hoax 154
homepages
 BIND 153
 sendmail 175
host name resolution 153
host table, local 153
http
 //www.bull.de/pub/out/ 111
 //www.ers.ibm.com 10
 //www.f-secure.com/products/ssh/ 62
 //www.nai.com 126
 //www.openssh.com 61
 //www.openwall.com/john/ 120
 //www.pgp.com 126
 //www.pgpi.com 126
 //www.pgpi.org/doc/faq/pgpi/en/#International
 124
 //www.pgpiinternational.com 126
 //www.sanstore.org 12
 //www.ssh.com 76
 //www.ssh.com/ssh 61
 //www.ssh.com/ssh/download.html 61
 //www.tripwiresecurity.com/ 111
 //www.tripwiresecurity.com/products/in-
 dex.cfml 111

 //www.vandyke.com/products/securecrf/ 62
 //www-3.ibm.com/security/ 47
 //www-4.ibm.com/software/security/firewall/ 47
 //www-4.ibm.com/software/security/fire-
 wall/about/ 49
 //www-frec.bull 65
 //www-frec.bull.fr 12, 61, 126
htpdlite 143

I

IBM Secure Mailer
 See Postfix
IBM Secureway Firewall 47, 49, 198, 199
IBM Secureway Firewall features 47
IBM Secureway Network Despatcher 48
ICMP 29, 186
ICMP echo packets 187
ICMP echo requests 199
icmpaddressmask 188
IDEA 64
imap2 153
imapd 153
IMN search engine 189
imnadm 189
IMNSearch.rte.htpdlite fileset 143
imnss 143
implementation of security framework 7
implications of having SSH 77
implicit drop rule 34
imgss 143
include 177
incoming mail 151
inetd 144
inetd daemon 148, 184, 198
inetd entry
 bootpd 151
 bootps 151
 chargen 152
 cmsd 153
 comstat 151
 daytime 152
 discard 153
 dtspc 153
 echo 152
 exec 151
 finger 151
 fingerd 151
 imap2 153

- imapd 153
- klogin 151
- krlogind 151
- krshd 151
- kshell 151
- login 151
- netstat 152
- ntalk 152
- pcnfsd 152
- pop3 153
- pop3d 153
- ps 152
- rex 152
- rexecd 151
- rlogind 151
- rshd 151
- rstatd 151
- ruserd 151
- rwalld 151
- shell 151
- sparyd 152
- systat 152
- talk 152
- talkd 152
- tftp 151
- tftpd 151
- time 152
- ttdbserver 153
- uucp 151
- uucpd 151
- inetd.conf 148
- infinite loop 152
- info search daemons, disabling 143
- InfoExplorer 143
- initial access 5
- inittab 142
- install additional security tools 197
- installing John the Ripper 120
- installing NSA 49
- installing OpenSSH on AIX 65
- installing TCP Wrapper 78
- installing Tripwire 111
- integrity, system and data 109
- internal firewalls 2
- internal network 19
- internal network, private 2
- internal resolution name servers 164
- internal servers, private 2
- Internet Boot Protocol server 151
- Internet downloads, verifying 200
- Internet Explorer 186
- Internet Protocol (IP) version 6
 - See IPv6
- interoperability between OpenSSH and SSH.Com 76
- interpreting NSA output 51
- intrusion detection on Check Point FireWall-1 43
- IP address resolution steps in DNS 154
- IP address, dynamic assignment 146
- IP forwarding 28
- IP routing options 188
- IP spoofing compared to DNS spoofing 156
- IP spoofing protection 39
- ip6srcrouteforward 188
- ipforwarding 28, 188
- ipforwarding on SP control workstation 188
- ipignoreredirects 188
- IPSec 7, 8, 15, 26, 180, 183
- ipsendredirects 188
- ipsrcrouteforward 188
- ipsrcrouterecv 188
- ipsrcrouteseend 188
- ipv4 BIND 4.9.3 157
- IPv6 147
- ipv6 BIND 8.1.2 157
- ipv6 BIND 8.2.2 (patch 5) 157
- IPv6 daemons, disabling 147

J

- John 109
- John the Ripper 109, 110, 119, 194, 197, 198, 200
- John the Ripper, configuring 121
- John the Ripper, installing 120
- John the Ripper, obtaining 120
- John the Ripper, using 122

K

- Kerberos 151
- kernel messages 144
- kernel services, NLuprintf 144
- kernel services, uprintf 144
- key concepts of SSH 62
- klogin 149, 151
- krlogind 151
- krshd 151
- kshell 149, 151

L

- last 201
- legacy SNMP daemon, dpid2 148
- licensed products 198
- light parody 154
- list of known vulnerabilities 199
- List of ports that Check Point FireWall-1 uses 43
- ListenAddress 75
- local host table 153
- login 151
- loginretries 190
- logintimes 192
- LogLevel 76
- loop, infinite 152
- loopback alias 199
- lpd 147, 189
- ls subcommand in nslookup 159, 167
- lspp 68
- lsnf 144

M

- mail, incoming 151
- mailnull 177, 178
- make 170
- man pages 143
- managing FireWall-1 logs 29
- MANPATH 68, 127
- maxage 195, 196
- maxexpired 195, 196
- maxrepeats 195, 197
- MD5 109, 110, 198, 200
- messages, kernel 144
- MIB 180
- MIMESweeper 48
- minage 195, 196
- minalpha 195, 196
- mindiff 195, 197
- minlen 195, 197
- minother 195, 196
- modifying startup files 142
- modifying user password attributes 194
- monitor cron and at jobs 201, 203
- monitor files and directories 201, 202
- monitor logs, audit trails, and system behavior 200
- monitor system logs 200, 201
- monitoring all at jobs 204
- monitoring all cron jobs 204
- monitoring security framework 8

- monitoring, elements of 200
- mouted 148
- multicasting daemons, disabling 148

N

- name resolution 153
- name resolution services 147
- named 147, 157
- named daemon in bos.net.tcp.client 157, 165
- named, relinking 158
- named, restarting 158
- named, symbolic link 157
- named-xfer, relinking 158
- NAT 41
- national language code points 197
- ndnpd-router 147
- ndpd-host 147
- needexnhelo 176
- needmailhelo 176
- needvrfyhelo 176
- netmonAdapterHealth 188
- Netscape 186
- netstat 44, 152
- netstat -af inet 144
- network address translation (NAT) 40
- Network Information System
 - See NIS
- network management daemons, disabling 148
- network options 186
- network scanners 199
- Network Security Auditor
 - See NSA
- Network Security Auditor (NSA) 47, 49
- network services 141
- Network Time Protocol
 - See NTP
- network topology verification 199
- network&netmask 161
- network, external 19
- network, internal 19
- network-booting an SP node 151
- NFS 153
 - biod, defined 174
 - configuring 168
 - daemons in bos.net.nfs.client fileset 174
 - daemons, biod 174
 - daemons, nfsd 174
 - daemons, rpc.lockd 174

- daemons, rpc.mountd 174
- daemons, rpc.statd 174
- disabling rpc.lockd 175
- disabling rpc.statd 175
- most common security mistake 173
- nfsd, defined 174
- PC clients 152
- protecting 173
- rpc.lockd, defined 174
- rpc.mountd, defined 174
- rpc.statd, defined 174
- NFS (port 2049) 170
- nfs client tool 170
- nfsd 174
- nfsd, defined 174
- NIS 153, 191
 - configuring 168
- NIS+ 168, 173
- NLuprintf 144
- nmap 198, 199
- no 28, 186
- no options 186
 - bcastping 187
 - clean_partial_conns 187
 - directed_broadcast 188
 - icmpaddressmask 188
 - ip6srcrouteforward 188
 - ipforwarding 188
 - ipignoreredirects 188
 - ipsendredirects 188
 - ipsrcrouteforward 188
 - ipsrcrouterecv 188
 - ipsrcroutesev 188
 - nolocsrcroute 188
 - tcp_pmtu_discover 188
 - udp_pmtu_discover 188
- nobody (-2) 172
- node customization 181
- node installation 181
- noexpn 176
- no-fetch-glue (BIND4) option 163
- nolocsrcroute 188
- non-privileged ports (above 1023) 174
- nonsense 154
- no-recursion (BIND4) option 162
- nosak 191
- notsh 191
- novrfy 176
- NSA 141, 176, 182, 199
- nsa 50
- NSA output, interpreting 51
- NSA, installing 49
- NSA, interpreting its output 51
- NSA, using 50
- nsaauditor.base 49
- nslookup 159, 165
 - DNS debugging 168
 - listing all servers in a domain 167
 - ls subcommand 167
 - querying for name servers in a domain 166
 - querying specific types of records 166
 - querying Start Of Authority (SOA) record 166
 - server subcommand 167
 - zone transfers with ls subcommand 159, 167
- nsupdate, relinking 158
- ntalk 152
- NTP 147
- nuucp 189

O

- obtaining and installing John the Ripper 120
- obtaining and installing PGP 126
- obtaining and installing TCP Wrapper 78
- obtaining and installing Tripwire 111
- obtaining John the Ripper 120
- obtaining PGP 126
- obtaining SSH 61
- obtaining TCP Wrapper 78
- obtaining Tripwire 111
- octal mask 192
- on 152, 191
- Open Platform for Secure Enterprise Connectivity (OPSEC) 25
- OpenSSH 61, 68
- OpenSSH using SSH1 68
- OpenSSH using SSH2 71
- OpenSSH, installing on AIX 65
- OpenSSH, interoperability with SSH.Com 76
- opensshd 69, 72
- options fetch-glue no (BIND8) 163
- options no-fetch-glue (BIND4) 163
- options no-recursion (BIND4) 162
- options recursion no (BIND8) 162
- Oracle 142
- other attributes 193

P

- packet, ACK 187
- packet, SYN 187
- packet, SYN/ACK 187
- packets, ICMP echo 187
- passive ftp 185
- password cracker utility 197
- password minimum length determination 197
- password policy, strong 194
- PATH 68, 127
- PC clients, NFS 152
- PC, SSH clients for 76
- pcnfsd 152
- PERL script to convert /etc/hosts to BIND8 158
- PERL script to convert BIND4 to BIND8 158
- PermitRootLogin 75
- PGP 198, 200
- PGP basics 124
- PGP, installing 126
- PGP, obtaining 126
- PGP, using 127
- pgpe 132
- pgpk 127
- Phil Zimmermann 126
- ping sweep 199
- piobe 143
- planning security framework 3
- planting back door 6
- pop3 153
- pop3d 153
- port allocation via rpc (portmap) 151, 152, 153
- port scanners 199
- portmap 144
- portmap daemon 168
- portmapper 168
- Ports 75
- ports
 - port 11 (TCP), systat 152
 - port 110 (TCP), pop3d 153
 - port 111 (TCP & UDP), portmapper 168, 174
 - port 13 (TCP & UDP), daytime 152
 - port 143 (TCP & UDP), imapd (imap2) 153
 - port 15 (TCP), netstat 152
 - port 161 (TCP & UDP), snmpd 180
 - port 19 (TCP & UDP), chargen 152
 - port 25 (TCP), SMTP (sendmail) 176, 177
 - port 37 (TCP & UDP), time 152
 - port 512 (TCP), rexecd 151
 - port 512 (UDP), comstat 151
 - port 513 (TCP), rlogind 151
 - port 514 (TCP), rshd 151
 - port 518 (TCP), talkd 152
 - port 540 (TCP), uucpd 151
 - port 543 (TCP), krlogind 151
 - port 544 (TCP), krshd 151
 - port 6000 (TCP), X 183
 - port 67 (UDP), bootpd 151
 - port 69 (UD), tftpd 181
 - port 69 (UDP), tftpd 151
 - port 7 (TCP & UDP), echo 152
 - port 79 (TCP), fingerd 152
 - port 9 (TCP & UDP), discard 153
- PortSentry 198, 199
- PortSentry and /etc/hosts.deny 199
- PortSentry and loopback aliases 199
- POSTFIX 78
- Postfix 180
- postman 177
- Pretty Good Privacy (PGP) 109, 110, 124
- principles, DNS 154
- print daemons, disabling 143, 147
- printq 189
- PrivacyOptions 176
 - authwarnings 176
 - goaway 176
 - needexphelo 176
 - needmailhelo 176
 - needvrfyhelo 176
 - noexpn 176
 - novrfy 176
 - restrictmailq 176
 - restrictqrun 176
- private internal network 2
- private internal servers 2
- private key, protecting 134
- private network in DMZ 2
- private-key 63
- privilege escalation 5
- proper network options 186
- protecting /etc/rc.dt with Tripwire 144
- protecting NFS 173
- protecting sendmail with TCP Wrapper 180
- protecting TCP services using SOCKS 186
- protecting your private key 134
- protection, broadcast 187
- protection, IP spoofing 39
- protocol, ftp 184
- proxy servers, SOCKS 186

ps 152
PSSP 173
public key 71
public servers in DMZ
public-key 63
public-key, private-key authentication scheme 184
pwdck 190
pwdwarntime 195, 196, 197

Q

qdaemon 143
Quaintance, Jeff 176
queries, restricting 162

R

RADIUS 25
rc.dt 144
rc.tcpip 144, 157
rcp 151, 198
recommendations for fixing vulnerabilities 199
recursion no (BIND8) option 162
recursion, restricting 161
recursive queries 164
referrals, DNS 162
refresh -s inetd 149
refresh the sendmail daemon 177
registry 190, 191
reliable clock source 147
remote access, secure 59
Remote Procedure Call
 See RPC
removing entries from /etc/inetd.conf 148
removing entries from /etc/inittab 142
removing entries from /etc/rc.tcpip 144
removing sendmail banner 177
removing unnecessary default accounts 189
removing unnecessary services 141
resolvers, DNS 156
restarting named 158
restrict cache update 163
restrict dynamic updates 161
restrict queries 162
restrict recursion 161
restrict zone transfers 159
restrictmailq 176
restrictqrun 176
rexed 152
rexecd 151

RFC 2065 156
rftp 185, 186
rlogin 148, 151, 190, 191
rlogind 148, 151
roles, AIX account 192
root SUID on sendmail 178
root, securing 192
root, steps for securing 192
routed 147
routers, boundary 2
routes, dynamic vs. static 147
routes, static vs. dynamic 147
routing daemons, disabling 147
routing, dynamic 147
RPC 168
rpc (portmap) 151, 152, 153
RPC authentication methods 168
rpc.lockd 174
rpc.lockd, defined 174
rpc.mountd 174
rpc.mountd, defined 174
rpc.statd 174
rpc.statd, defined 174
rpcinfo 168
rpcinfo, example 169
RS/6000 SP 147, 149, 151, 173, 174, 181, 188
RS/6000 SP and ipforwarding 188
RS/6000 SP umask 022 192
RSA 62, 64
RSA authentication 69, 71
rsh 148, 151
rshd 148, 151
rstatd 151
rtelnet 186
rtn 186
rule, cleanup 34
rule, implicit drop 34
rule, stealth 35
RunAsUser 177, 178
ruserd 151
rwalld 151
rwhod 148
rwhod security exposure 148

S

SafeFileEnvironment 178, 179
SAINT 141, 198, 199
SAINT, enhanced SATAN 199

- SAK key (^X^R) 191
- sak_enabled attribute 191
- SATAN 78, 180, 199
- scanning 5
- scp 148, 184, 198
- search engine, IMN 189
- Secure Attention Key (SAK) 191
- Secure NFS 173
- secure remote access 59, 198
- Secure RPC 173
- Secure Shell 59
- secure shell (ssh) 59
- secure_zone record, BIND4 163
- secure_zone, example 163
- SecureCRT 62
- SecureID 25
- Secureway Firewall 47, 49, 198
- Secureway Firewall features 47
- Secureway Network Despatche 48
- securing AIX 139
- securing DNS on AIX 159
- securing FireWall-1 27
- securing FireWall-1 default configurations 29
- securing firewalls 24
- securing root 192
- securing X11 182
- security exposure with /ftpboot directory 182
- security exposure, dpid2 148
- security exposure, rwhod 148
- security framework 1
 - architecture 6
 - implementation 7
 - incident response 10
 - monitoring 8
 - planning 3
- security framework, architecture 6
- security framework, building 2
- security framework, implementation 7
- security framework, incident response 10
- security framework, monitoring 8
- security framework, planning 3
- security tool suite 199
- security tools, third-party 197
- sendmail 144, 175, 198
- sendmail banner 176
- sendmail banner, removing 177
- sendmail built-in commands 175
- sendmail configuration file 175
- sendmail daemon 176
- sendmail homepage 175
- sendmail in daemon mode 177
- sendmail not running as root 177
- sendmail replacement 180
- sendmail restricted shell
 - See smrsh
- Sendmail Switch 178
- sendmail -v -bi 179
- sendmail version 8.8 177
- sendmail, .deadletter 178
- sendmail, .forward files 177
- sendmail, AIX versions 175
- sendmail, DefaultUser 177
- sendmail, include files 177
- sendmail, PrivacyOptions 176
- sendmail, root SUID 178
- sendmail, RunAsUser 177, 178
- sendmail, SafeFileEnvironment 178, 179
- services
 - network 141
- services configurations, tightening 153
- services, name resolution 147
- services, removing unnecessary 141
- set up strong password policy 194
- setting user attributes 190
- sftp 148, 184, 198
- SGID 202
- shadmgr.aix script 122
- shell 151
- showmount 170
- signatures, digital 156
- Simple Mail Transfer Protocol
 - See SMTP
- Simple Network Management Protocol
 - See SNMP
- smitty failed_logins 191
- smrsh 179
- SMTP 153, 175
- SMTP TCP port 25 180
- SMTPGreetingMessage 177
- smurf attacks 187
- SNMP 148, 153, 180
- SNMP community names 180
- SNMP community names, unique 148
- snmpd 144, 148
- SNMPv1 180
- SOCKS 185, 186
- SOCKS support in AIX 4.3.3 186
- SOCKS to protect TCP services 186

- SOCKS v5 186
- socksified ftp client 185
- socksified TCP stack 186
- source routing control 188
- SP 147, 149, 151, 173, 174, 181, 188
- SP administrative Ethernet 173
- SP and ipforwarding 188
- SP nightly mksysb backups 173
- SP node customization 192
- SP node installation 192
- SP required services
 - bootps 149
 - klogin 149
 - kshell 149
 - tftp 149
- SP switch 173
- SP umask 022 192
- split configuration, DNS 164
- spoofed ICMP traffic 187
- spoofing 184
- spoofing, comparison between DNS and IP 156
- spoofing, defined 154
- spoofing, DNS 154, 155
- spray 152
- sprayd 152
- SSH 59, 148, 184, 198
- ssh 148
- SSH clients for the PC 76
- SSH daemon configuration options 75
- SSH scp 184
- SSH sftp 184
- SSH with X 184
- SSH, alternatives to 77
- SSH, F-Secure 62
- SSH, implications of having 77
- SSH, key concepts 62
- SSH, obtaining 61
- SSH.Com, interoperability with OpenSSH 76
- SSH1 61
- SSH1, configuring and using 69
- SSH1, difference with SSH2 62
- SSH2 61, 71
- SSH2, configuring and using 72
- SSH2, difference with SSH1 62
- SSH2, used by OpenSSH 71
- ssh-keygen 71, 73, 76
- standard UNIX dictionary 194, 195
- stanza format 190
- Start Of Authority (SOA) record 166
- startup
 - AIX 142
 - HACMP 142
 - Oracle 142
 - system 142
 - UNIX 142
- starup
 - TCP/IP daemons 144
- static packet filter firewalls 15
- static routes 188
- static vs. dynamic routes 147, 188
- stealth rule 35
- stealth scan 199
- stealth scans 199
- steps for securing root 192
- stopping TCP/IP 145
- StrictHostKeyChecking 75
- strong account policies 189
- strong password policy 194
- su 201
- SUID 202
- suite, security tool 199
- Sun's Remote Procedure Call
 - See RPC
- symbolic link, BIND 157
- symbolic link, named 157
- symmetric key cryptography 63
- SYN attack 37, 38
- SYN attack defined 187
- SYN attack protection 187
- SYN packet 187
- SYN/ACK packet 187
- SYNDefender 37
- syslog 144
- SyslogFacility 76
- systat 152
- system and data integrity 109
- system behavior 200
- system logs 200
- system logs to monitor
 - /etc/security/failedlogin 201
 - /etc/utmp 201
 - /var/adm/sulog 201
 - /var/adm/wtmp 201
- system startup 142

T

- talk 152

talkd 152
 Tamu's Tiger 119
 target acquisition and information gathering 4
 TCP 186

- port 11, systat 152
- port 110, pop3d 153
- port 15, netstat 152
- port 25, SMTP (sendmail) 176, 177
- port 512, rexecd 151
- port 513, rlogind 151
- port 514, rshd 151
- port 518, talkd 152
- port 540, uucpd 151
- port 543, klogind 151
- port 544, krshd 151
- port 6000, X 183
- port 79, fingerd 152

 TCP & UDP

- port 111, portmapper 168, 174
- port 13, daytime 152
- port 143, imapd (imap2) 153
- port 19, chargen 152
- port 37, time 152
- port 7, echo 152
- port 9, discard 153

 TCP & UDP

- port 161, snmpd 180

 TCP stack, socksified 186
 TCP Wrapper 7, 8, 77, 148, 180, 198
 TCP Wrapper /etc/hosts.deny 199
 TCP Wrapper access control lists 198
 TCP Wrapper security features 82
 TCP Wrapper, configuring 79
 TCP Wrapper, installing 78
 TCP Wrapper, obtaining 78
 TCP Wrapper, protecting ftp with 184
 TCP Wrapper, protecting sendmail with 180
 TCP/IP

- stopping 145

 TCP/IP daemon startup 144
 TCP/IP daemons started by default 144
 tcp_pmtu_discover 188
 tcpdchk 80
 tcpdmatch 81
 telephone system analogy 156
 telnet 148
 telnet to port 25 on localhost (0) 176
 telnetd 148
 terminal emulators 183
 TFTP 181
 tftp 149, 151, 181
 TFTP access control 181
 tftpd 151
 The Bat Book 179
 The X Windows System Administrator's Guide, Volume 8 183, 216
 third-party security tools 197
 threat assessment 4
 three-digit octal mask 192
 tighten remaining services configurations 153
 tighten up user accounts 189
 time 152
 time bombs 203
 time daemons, disabling 147
 time synchronization daemons 147
 timed 147
 TIMEOUT 194
 Tivoli Storage Manager

- See TSM

 TMOUT 194
 Tools

- gzip 170
- lsof 144
- nfs 170
- NSA 141
- nslookup 159, 165
- rftp 185, 186
- rtelnet 186
- rtn 186
- SAINT 141
- scp 184
- sftp 184
- xwd 182
- xwud 182

 Topology Services daemon 188
 tpath 190, 191
 tpath always 191
 tpath nosak 191
 tpath notsh 191
 tpath on 191
 Tripwire 109, 110, 192, 198, 200
 Tripwire and CDE 144
 Tripwire CheckConfig script 115
 Tripwire secure database 200
 Tripwire, alternatives to 119
 Tripwire, comments on configuration 118
 Tripwire, configuring 112, 114
 Tripwire, installing 111

Tripwire, obtaining 111
Tripwire, using 112
Tripwire, when should be run 118
tripwire.list.sort 114
Trivial File Transfer Protocol
 See TFTP
Trusted Computing Base (TCB) 119
try before you buy 198
TSM 142
ttbserver 153
tty0 191
tty0 (console) 191
ttys 190, 191
turning off glue fetching 163
tw.config 115
TXT (txt) records, DNS 163
types of firewalls 15

U

UDP 186
 port 512, comstat 151
 port 67, bootpd 151
 port 69 181
 port 69, tftpd 151
udp_pmtu_discover 188
umask 190, 191
umask 022 192
umask 077 192
uninstalling CDE 144
unique SNMP community names 148
UNIX advisory locking 174
untrapped-meta-character vulnerabilities 152
update requests, counterfeit 155
uprintf 144
uprintfd 144
user account cleanup 189
user attributes, setting 190
user password attributes, modifying 194
using John the Ripper 122
using NSA 50
using PGP 127
using Tripwire 112
usrck 190
uucp 151, 189
uucpd 151

V

Venema, Wietse 180

verifying Internet downloads 200
viewing connections 36
VRFY 175
VRFY, defined 176
vulnerabilities, untrapped-meta-character 152

W

webserver, default 143
who 201
who /etc/security/failedlogin | more 201
Wietse Venema 180
world-writable directories 203
world-writable files 203
Wrapper, TCP 148
write 144
writesrv 144
www.checkpoint.com 25
www.checkpoint.com/support/technical/docu-
ments/index.html 27
www.cisco.com/univercd/cc/td/doc/product/iaa-
bu/centri4/user/scf4ch3.htm 17
www.enteract.com/~lspitz/fwtable.txt 36
www.enteract.com/~lspitz/intrusion.htm 43
www.fc.net/phrack/files/p48/p48-13.html 38
www.icsa.net/html/communities/firewalls/certifica-
tion/vendors/index.shtml 24
www.infoworld.com/supple-
ments/99poy_win/99poy_s.html 47
www.mimesweeper.integralis.com 48
www.onsight.com/faq/ssh/ssh-faq.html 60
www.phoneboy.com/fw1 25
www.phoneboy.com/fw1/ 44
www.telemate.net 49
www-4.ibm.com/software/network/dispatcher/ 48

X

X client 184
X libraries 182
X over SSH 184
X server 183
X window system 144
X xterm 183
X11 access control 182
X11 xauth 182
X11 xhost 182
X11 xhost - 183
X11 xhost + 182, 183
X11 xlock 183

X11 xwd 182
X11 xwud 182
X11, securing 182
X11.apps.clients filesset 182
X11Forwarding 76
xauth 182
xfrnets directive 159
xfrnets, example 160
xfrnets, more restrictive example 161
xhost 182
xhost - 183
xhost + 182, 183
xlock 183
xntpd 147
xterm 183
xwd 182
xwud 182

Y

Yellow Pages (YP)
 See NIS

Z

zone substatements 160
zone transfers with nslookup 159, 167
zone transfers, restricting 159

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at ibm.com/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

| | |
|---|--|
| Document Number | SG24-5971-00 |
| Redbook Title | Additional AIX Security Tools on IBM @server pSeries, IBM RS/6000, and SP/Cluster |
| Review | |
| What other subjects would you like to see IBM Redbooks address? | |
| Please rate your overall satisfaction: | <input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor |
| Please identify yourself as belonging to one of the following groups: | <input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above |
| Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities. | <input type="checkbox"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction. |
| Questions about IBM's privacy policy? | The following link explains how we protect your personal information. ibm.com/privacy/yourprivacy/ |



Redbooks

Additional AIX Security Tools on IBM @server pSeries, IBM RS/6000, and SP/Cluster



Redbooks

Additional AIX Security Tools on

IBM pSeries, IBM RS/6000, and SP/Cluster

Customize the security of your pSeries systems

Explore IBM, non-IBM, and freeware security tools

Learn new approaches to security

From firewalls to operating system hardening, this redbook illustrates additional tools and techniques that you can use to enhance the security environment of your pSeries, RS/6000 workstation, SP, or Cluster. The approach taken is from outside to inside and from top to bottom. We move from the servers on the far reaches of your network that are visible to the outside world to those in the innermost recesses of your intranet containing your most confidential data. As we move through these servers, we work from the application layer at the top to the network layer at the bottom. Along the way, we cover third-party software that is readily available, modifications to the standard software that comes with AIX and PSSP, and assorted techniques that can all be used to provide enhanced security in your environment.

Subjects covered in this redbook include:

- Firewalls
- Secure Remote Access
- Network Mapping and Port Scanning
- System Integrity
- Securing AIX

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5971-00

ISBN 0738418382