Gauntlet™ for IRIX™ Administrator's Guide

Document Number 007-2826-004

CONTRIBUTORS

Written by John Raithel with updates by Pam Sogard
Production by Julie Sheikman
Engineering contributions by Ed Mascarenhas
St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

Gauntlet™ for IRIX™ Administrator's Guide
Document Number 007-2826-004

Contents

List of Figures

About This Guide

Audience

This guide is intended for firewall administrators. It assumes familiarity with UNIX® system administration, networking and basic firewall concepts. System administrators should be familiar with TCP/IP, domain name service, sendmail, and router configuration. Consult your local library, bookstore, network resources, and IRIX® administrator for additional references.

About This Guide

This guide is comprised of three parts and contains the following chapters:

Part I, “Understanding the Gauntlet Internet Firewall,” presents the initial information about the firewall.

- Chapter 1, “Understanding the Gauntlet Firewall,” presents an overview of what firewalls are and why they are important. It presents an overview of how the Gauntlet™ firewall system works.

Part II, “Configuring and Using Proxies,” explains how to configure the various applications and proxies.

- Chapter 2, “Managing SMTP Services,” explains what the SMTP proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for mail applications.
- Chapter 3, “Managing POP3 Services,” explains what the POP3 proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for mail applications.
- Chapter 4, “Managing Terminal Services,” explains the types of terminal service applications that the Gauntlet firewall supports. It explains what the TELNET and Rlogin proxies do and how they work. It presents instructions for configuring the

Gauntlet firewall, as well as required and potential configuration steps for the terminal applications.

- Chapter 5, “Managing FTP Services,” explains what the FTP proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for the FTP application. It also includes notes on running an anonymous FTP server.
- Chapter 6, “Managing Rsh Services,” explains what the Rsh proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for Rsh.
- Chapter 7, “Managing Gopher and WWW Services,” explains the types of information services the Gauntlet firewall supports. It explains what the HTTP proxy does for HTTP, SHTTP, SSL, and Gopher proxies and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for these applications.
- Chapter 8, “Managing RealAudio Services,” describes the RealAudio proxy, which securely handles requests to listen to audio data.
- Chapter 9, “Managing MediaBase Services,” describes the MediaBase proxy, which securely handles requests to play video and multimedia data.
- Chapter 10, “Managing X Window Services,” explains what the X11 proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for the X11 applications.
- Chapter 11, “Managing LP Services,” explains what the lp proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for lp.
- Chapter 12, “Managing Sybase Services,” explains what the Sybase proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for Sybase.

Part III, “Administering General Gauntlet Firewall Services,” presents information on the other administrative tasks for the Gauntlet firewall.

- Chapter 13, “Managing NNTP and General TCP Services,” explains the types of News and network services the Gauntlet firewall supports. It explains what the plug proxy does and how it works. It presents instructions for configuring the Gauntlet firewall, as well as required and potential configuration steps for the News and network applications.

- Chapter 14, “Managing General TCP Services With Authentication,” explains what the user authentication management system does, and how to use it with the supported strong authentication systems.
- Chapter 15, “Managing Information Services on the Firewall,” explains how the system logs activity. It explains the different types of reports, how to configure them, and how to interpret them.
- Chapter 16, “Using the Network Access Control Daemon,” discusses firewall backup and explains how to ensure that the firewall contains the files and data that it should.
- Chapter 17, “The Graphical Management Interface,” explains what the graphical administrative interface does, how to access it from local and remote locations, and how to use it to configure your Gauntlet firewall.
- Chapter 18, “Managing User Authentication,” explains what the user authentication management system does, and how to use it with the supported strong authentication systems.
- Chapter 19, “Using the Login Shell,”explains what the login shell does and how it works. It presents instructions for configuring the Gauntlet firewall for more secure access.
- Chapter 20, “Logging and Reporting,” explains how the system logs activity. It explains the different types of reports, how to configure them, and how to interpret them.
- Chapter 21, “Backups and System Integrity,” explains considerations for incorporating the Gauntlet firewall in an administrator’s general backup schedule. It also presents considerations for restoring the Gauntlet firewall.

“Appendixes” present reference material.

- Appendix A, “Gauntlet System Files,” explains the format and precedence of the trusted and untrusted network tables that the Gauntlet firewall uses.
- Appendix B, “Netperm Table,” explains the format and precedence of the netperm-table, which contains configuration information for the Gauntlet firewall, and the concepts behind policies.
- Appendix C, “Virtual Private Networks,” Virtual Private Networks, explains how you can use your Gauntlet Internet Firewall to exchange encrypted traffic with other Gauntlet Firewalls.

- Appendix D, “Configuring SSL on the Gauntlet Firewall,” explains the Secure Socket Layer protocol and how to configure it to protect remote administration sessions of the Gauntlet firewall.

The Glossary presents definitions of terms used in this document.

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Bold—Literal command-line arguments.

Italics—Backus-Naur Form entries, executable names, filenames, IRIX commands, URLs, manual/book titles, new terms, onscreen button names, tools, utilities, variable command-line arguments, and variables to be supplied by the user in examples, code, and syntax statements.

`Fixed-width type`—Prompts, and onscreen text.

`Bold fixed-width type`—User input, including keyboard keys (printing and nonprinting); literals supplied by the user in examples, code, and syntax statements (see also <>).

ALL CAPS—Environment variables.

“ “ (Double quotation marks)—Onscreen menu items and references in text to document section titles.

() (Parentheses)—Following IRIX commands, parentheses surround the reference page (man page) section number.

[] (Brackets)—Surrounding optional syntax statement arguments.

<> (Angle brackets)—Surrounding nonprinting keyboard keys, for example, <Esc>, <Ctrl-d>, and surrounding required variables in italicized text.

#—IRIX shell prompt for the superuser (root).

%—IRIX shell prompt for users other than superuser.

Installation and System Requirements

Refer to the release notes with your Gauntlet firewall product for information regarding software and hardware requirements as well as installation information.

Additional Resources

This collection of resources is presented as a starting point for your information. It is not an endorsement of any of the products or organizations.

Books

Building Internet Firewalls. Chapman, D. Brent & Zwicky, Elizabeth. O'Reilly & Associates, Inc. ISBN 1-56592-124-0.

Firewalls and Internet Security: Repelling the Wily Hacker. Cheswick, Steven M. & Bellovin, William R. Addison Wesley. ISBN 0-201-63357-4.

Newsgroups

comp.security.firewalls—Discussions of anything regarding network security firewalls.

Mailing Lists

The Firewalls mailing list is for discussions of Internet firewall security systems and related issues. Relevant topics include the design, construction, operation, maintenance, and philosophy of Internet firewall security systems.

To subscribe to the regular mailing list, send the following command in the body of an email message (NOT on the "Subject:" line!) to majordomo@greatcircle.com:

```
subscribe firewalls
```

To subscribe to the digest version of the mailing list, send the following command in the body of an email message (NOT on the "Subject:" line!) to majordomo@greatcircle.com:

```
subscribe firewalls-digest
```

Frequently Asked Questions Lists

The Internet Firewalls Frequently Asked Questions list is maintained by Marcus J. Ranum and located at:

`http://www.v-one.com/pubs/fw-faq/faq.htm`

White Papers

Application Gateways and Filtering Gateway: *A Comparison of Firewall Designs* Avolio, Frederick M. and Sebes, J. Data Security Letter, Number 59.

`http://www.tis.com/Home/NetworkSecurity/Firewalls/FWComp.html`

Firewalls Are Not Enough Avolio, Frederick M. *Data Security Letter*, Number 50.

http://www.tis.com/Home/NetworkSecurity/Firewalls/FirewallsNotEnough.html

A Network Perimeter with Secure External Access Avolio, Frederick M. and Ranum, Marcus J. Internet Society Symposium on Network and Distributed Systems Security, February 1994.

http://www.tis.com/Home/NetworkSecurity/Firewalls/isoc.html

ftp.tis.com */pub/firewalls/isoc94.ps.Z*

Thinking About Firewalls Ranum, Marcus J. Presented at SANSII, 1993.

http://www.tis.com/Home/NetworkSecurity/Firewalls/ThinkingFirewalls.html

ftp.tis.com */pub/firewalls/firewalls.ps.Z*

A Toolkit and Methods for Internet Firewalls Avolio, Frederick M. and Ranum, Marcus J.

http://www.tis.com/Home/NetworkSecurity/Firewalls/Usenix.html

ftp.tis.com */pub/firewalls/usenix-paper.ps.Z*

How to Get Latest Security Patches

The CD-ROM containing the Gauntlet firewall software contains necessary security patches (if any) at the time of product release, so be sure to install those patches. Stay in touch with the WWW site for Silicon Graphics Security Headquarters at http://www.sgi.com/Support/Secur/security.html for new security patches and security advisories. Be sure to install any security patches that replace patches found on your CD-ROM.

PART ONE

Understanding the Gauntlet Internet Firewall

Chapter 1

Understanding the Gauntlet Firewall

The Gauntlet Internet Firewall is a hardware- and software-based firewall system that provides secure access and internetwork communications between private networks and public networks (such as the Internet), and between subnets of private networks. The firewall offers application-level security services for both incoming and outgoing communications based on existing security practices or an organization's security policies.

If the paragraph above does not make any sense, do not despair. This chapter provides an overview of the Gauntlet Firewall and how it works. However, it is not a thorough discussion of firewalls or security practices. Consult "Additional Resources" on page xxiii for a list of other resources that provide excellent introductory and advanced discussions of firewalls.

Understanding Gauntlet Firewall Concepts

Simply put, a firewall is a single point of defense that protects one side from the other. In networking situations, this usually means protecting a company's private network from other networks to which it is connected. Firewalls can be as simple as a router that filters packets or as complex as a multi-machine, multi-router solution that combines packet filtering with application gateways.

Design Philosophy

The Gauntlet Internet Firewall exemplifies a minimalist and reductionist approach. Simple is better than complex. It follows this paradigm:

That which is not expressly permitted is prohibited.

The firewall will only allow activities which are explicitly set, either through system defaults or through your own configurations. New services can't slip through the

firewall unless you allow them through. You must be able to identify and remove any “back doors” that may be surreptitiously put into place.

All of the software is written with the idea that simplicity is an important advantage. The number of lines of code for the various proxies and utilities are smaller than their standard IRIX counterparts. This makes the programs readable, understandable, and less prone to having an error hidden in some complex programming structure. Also, the source code for the Gauntlet firewall is provided so anyone can examine and confirm the programs operation. They are also examinable by any Gauntlet customer, not hidden away in some sort of black box. The security of the Gauntlet Internet firewall does not depend on secret algorithms or source code.

Recognizing that most security breaches occur through a compromised user account, the Gauntlet Internet Firewall generally has no user accounts. While you can setup an administrator account, users do not need to log into the firewall to access information on the other side.

The Gauntlet Internet Firewall is auditable, controllable, and configurable. You can configure many options to match your security policies. The software logs the specified activities and processes fore review, so that if you suspect a security breach you can look back to the log files to see if and when it might have happened.

Security Perimeter

Establishing a network security perimeter involves designating a network of machines that you wish to protect and defining the mechanisms used to protect them. The firewall is the communications gateway for all hosts within the perimeter. To have a successful network security perimeter, *all* communications to hosts inside the perimeter must pass through the firewall.

Trusted and Untrusted Networks

Your firewall must be configured to differentiate between the “good guys” and the “bad guys.” The firewall makes this determination using information you provide about different networks. It understands three types of networks: trusted, untrusted, and unknown.

Trusted Networks

Trusted networks are the networks inside your security perimeter. Trusted networks are usually the ones that you are trying to protect. Often, you or someone in your organization administers the machines on these networks. Your organization controls the security measures for these networks. Usually, they are within the physical security perimeter. They can also be connected by links you control in a Virtual Private Network, as explained in Appendix C.

When you set up the firewall, you explicitly configure the networks your firewall can trust. After initial configuration, the trusted networks usually include the firewall itself and all networks behind the firewall.

Untrusted Networks

Untrusted networks are the networks outside your security perimeter. They are untrusted because they are outside of your control or knowledge. You have no control over the administration or security policies for these sites. They are the ones from which you are trying to protect your network. However, you still need to and want to communicate with these networks, even though they are untrusted.

When you setup the firewall, you explicitly configure the networks from which your firewall can accept requests, but which it does not trust. By default, after initial configuration, the untrusted networks are all networks outside the perimeter.

The firewall applies different policies (sets of rules) for requests from untrusted networks than it does for requests from trusted networks. For some types of requests (including TELNET, FTP, rlogin, rsh, and POP3), the firewall may use additional authentication before processing the request. For others, the firewall may deny the request altogether.

Unknown Networks

Unknown networks are those networks that are neither trusted or untrusted. They are unknown quantities to the firewall because you have not explicitly told the firewall that this network is a trusted or an untrusted network. By default, there are no unknown networks because the default list of untrusted networks covers everything that is not a trusted network.

Consider a company that lists its own networks as the trusted network. The company lists the networks for three clients as the untrusted networks. All other networks on the Internet are now unknown networks and cannot pass requests through the firewall.

Policy

Just as you have a general security policy for your organization, the Gauntlet Internet Firewall uses policies to summarize its rules. The policies are collections of rules about what the firewall can and cannot do in particular situations. They indicate which proxies can run, and whether they require authentication, special logging, or other general settings. The firewall policies, which you create, should be based on your site security policies.

By default, the Gauntlet firewall includes one set of policies for requests from trusted networks and one set of policies for requests from untrusted networks. The firewall determines which policy applies by the source IP address of the request. The default policy for trusted networks does not require users to authenticate; the default policy for untrusted networks does require users to authenticate. When installed, all services are turned off. It is up to you to enable the services which your site needs.

Transparency

Transparency indicates that your firewall is not visible to your users as they work. They can continue to TELNET to client sites without having to explicitly stop at the firewall.

The default Gauntlet firewall configuration implements transparency inside your firewall for your trusted networks. This is accomplished by creating default routes that send all requests to untrusted networks through the firewall and by certain configuration options on the firewall.

In contrast, the firewall does not implement transparency for requests from untrusted networks. In this case, you want users to be aware that they are entering your site through your firewall.

The advantage of transparent access is that you do not need to reconfigure your client systems or learn new procedures in order to use supported services. Non-transparent access is supported, but users must learn procedures to perform their tasks.

Understanding Gauntlet Firewall Components

Hardware and Software

The Gauntlet firewall uses hardware and software to protect your network.

Hardware

The specific hardware components of the Gauntlet Internet Firewall are the network interfaces. Multiple network interface cards can be used to physically separate networks from one another.

Software

The software components of the firewall include a “hardened” operating system, application-level security services, security programs, and other management utilities.

Operating System

The operating system is a version of the standard Silicon Graphic’s IRIX operating system, “hardened” by the Gauntlet software (see “Introductory Management Form” on page 118 for information on minimizing exposure while implementing the Gauntlet software.) All known security holes are patched as of the release of the Gauntlet product (refer to “How to Get Latest Security Patches” on page xxv for information on security patches.) As part of the firewall, the operating system has been tailored to provide support for only the services necessary to run the firewall. For example, source routing is not honored, and ICMP redirects are rejected. These services change the directions that routed packets flow and could direct networks to circumvent the firewall. Services such as NFS®, NIS, and RPC cannot easily be made secure and so should be disabled (refer to “Introductory Management Form” on page 118 for more information on minimizing exposure.)

Unsupported network services do not just report an error to the requesting site. The operating system logs these access attempts, providing information about probes of your system.

Application-Level Security Services (Proxies)

The software on the Gauntlet firewall includes security services on a per-application protocol basis. As noted above, all packets, and therefore all application requests, go to the firewall. On the firewall, proxy software relays information from one side of the firewall to the other. The proxy prevents the applications on outside networks from talking directly with the applications on your inside network, and vice versa. No IP packets pass directly from one side of the firewall to the other. All data is passed at the application level. (The “trusted ports” feature in this implementation is an exception to this generalization.)

Each application generally talks through a different proxy that understands the protocol for that application. Currently, the Gauntlet firewall includes proxies for the following types of services:

- Terminal services (TELNET and rlogin)
- Electronic mail (SMTP and POP3)
- File transfer services (FTP)
- Remote Execution (Rsh)
- Usenet news (NNTP)
- Web services (HTTP, SHTTP, SSL)
- Gopher services (Gopher, Gopher+)
- X Window services (X11)
- Printing services (lp)
- SQL services (Sybase SQL Server)
- Audio service (Real Audio)

In addition, the Gauntlet firewall includes a generic plug-board proxy. This proxy connects TCP traffic from a particular port on one side of the firewall to a particular port on another system on the other side of the firewall. As with the service specific proxies, no IP packets pass directly from one side of the firewall to the other. If you have not installed a proxy for a service, that type of traffic does not pass through the firewall.

Because the proxies use the same protocols to communicate as the applications, you do not need to modify the original client or server applications. For example, when the TELNET application connects to the firewall it and the proxy both communicate using

the standard TELNET protocol in RFCs 764 and 854. You can continue to use the same TELNET application to connect to remote sites.

All of the proxies are configurable. You can accept or reject requests to or from certain sites and networks, or set up other rules that the proxies use when passing requests through the firewall. You can also enable or disable individual proxies and run only the ones that you need. You can easily translate your security policies into configuration rules.

The proxies log all activities to and through the firewall. You can use the logs to gather usage statistics or to look for potential attacks.

In addition, several of the proxies support strong user authentication systems. These one-time passwords or security token systems provide additional security because each time users access the network they use a different password that cannot be reused if "sniffed" by an attacker.

Additional Features

The Gauntlet Firewall provides additional security by using the IRIX IP filter utility *ipfilterd* (see ipfilterd(1M)). This allows Gauntlet to check IP packets based on several criteria (for example, address and protocol) and processes or rejects the packets. It detects spoofed packets claiming to be from one network that are actually from another network. This software also allows Gauntlet to be transparent to your users for most activities.

Management Utilities

In addition, the Gauntlet firewall also contains several programs that ease the job of administering the firewall. These include management tools for configuring the firewall, scripts for reporting activity through the firewall, and performing general administration.

The *gauntlet-admin* administrative tool provides access for most standard configuration activities. You do not need to modify system files or configuration files unless you want to further customize your configuration.

The Gauntlet Firewall also includes shell scripts that assist in creating backups and checking integrity.

How a Firewall Works

Consider a company, Yoyodyne, that has a connection to the Internet via an Internet service provider (ISP). They have installed a Gauntlet Internet Firewall to protect their corporate network (yoyodyne.com) from all other hosts on the Internet. They are using the standard configuration shown in Figure 1-1.

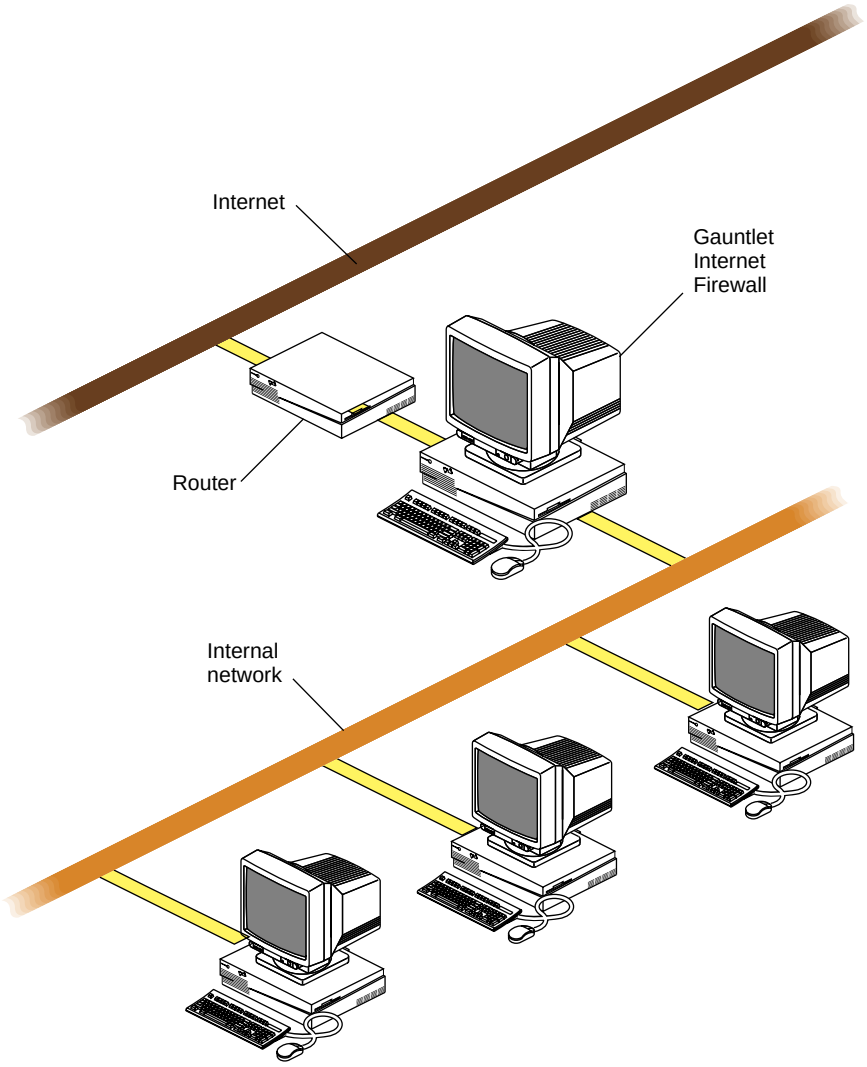

Figure 1-1 Gauntlet Internet Firewall Standard Configuration

The Yoyodyne network is first separated from the rest of the Internet by a router. The router only passes traffic from the Internet to the Gauntlet firewall when that traffic is bound for some part of the Yoyodyne internal network. More sophisticated routers can additionally strengthen a companies security perimeter by implementing certain security functions such as “IP spoofing filters”.

The firewall is helping to establish a security perimeter to protect the internal network. It screens *all* requests that need to pass from one side of the firewall to the other. Using rules Yoyodyne created based on their security policies, the firewall determines whether to accept or pass requests through (at the application level) to the other side.

Dual-Homed Bastion Host

In order to protect the inside network, the firewall must be able to see all of the packets intended for hosts on the inside network. While there are a number of ways to physically and logically accomplish this, the recommended configuration is the firewall machine installed as a dual-homed bastion host.

As a dual-homed bastion host, the firewall machine has two network interface cards, and thus two connections: one to your network and one to the outside, as shown in Figure 1-2.

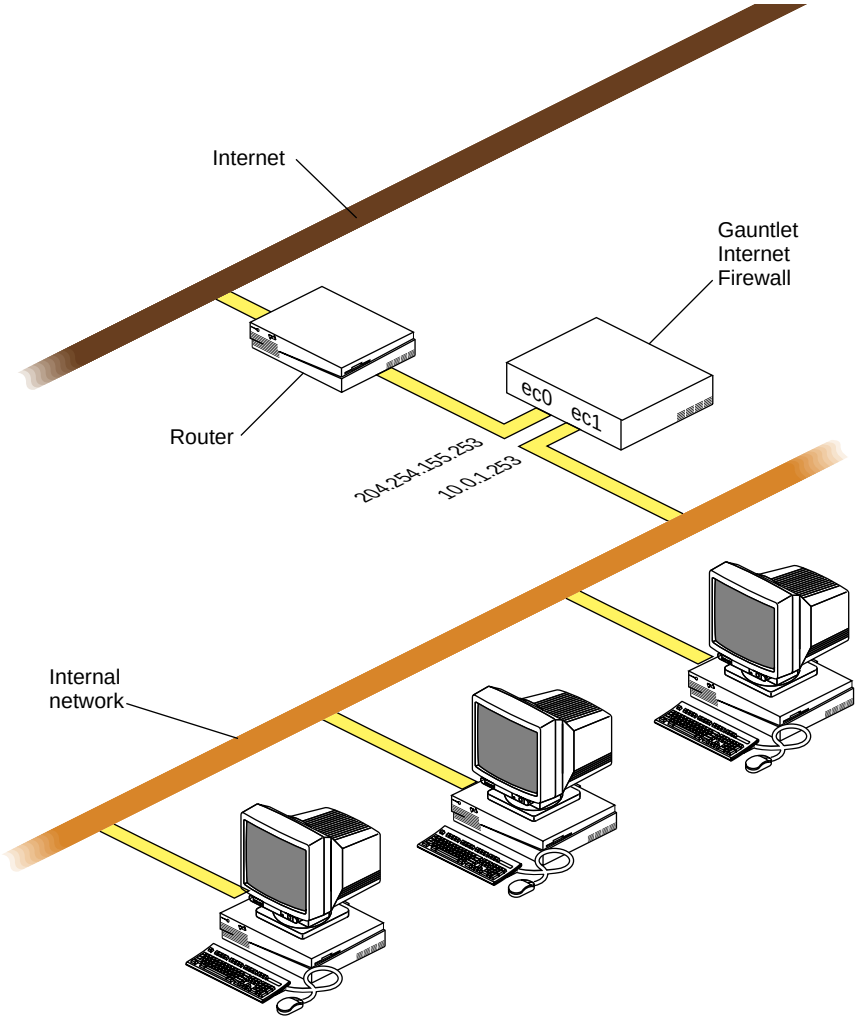

Figure 1-2 Dual-Homed Bastion Host

All outside network traffic enters and exits the firewall through one network interface, such as ec0. Similarly, all inside network traffic enters and exits through a network interface, such as ec1. To accomplish this, each interface has a separate IP address. Yoyodyne was assigned the 204.254.155 network, and chose 204.254.155.253 as the outside IP address and 10.0.1.253 for the inside IP address.

Note: You can also use two firewalls to create a virtual private network (or a virtual network perimeter), exchanging encrypted information across an untrusted network. Because of United States government export regulations, this feature is generally not available outside the United States and Canada. Refer to Appendix C for more information.

Processing Packets and Requests

The firewall follows a standard set of steps for the packets it receives on either interface:

1. Receive packet.
2. Check source and destination.
3. Check request type.
4. Run appropriate program.
5. Process the request.

As we examine each step of the process, consider a Yoyodyne employee working at a client site (outside the perimeter) who needs access to a machine at work via TELNET.

Receive Packet

Routing information on outside hosts and at the ISP directs all requests for the company to the firewall. In addition, the domain name system (DNS) on the firewall and other outside DNS servers advertises the outside IP address of the firewall as the *only* way to connect to anything on the inside network. Hosts on the inside network use routing information to direct all requests for outside networks to the inside address of the firewall.

For example, the client company machines consult their routing information and pass the TELNET request along until it reaches the Yoyodyne firewall.

Check Source and Destination

Once the firewall receives a packet, it must determine what to do. First, the operating system examines the destination of the packet and determines whether it needs to deliver the packet locally. Local delivery includes packets destined for hosts inside the firewall. The firewall grabs these packets and gives them to an appropriate proxy. If there is no

proxy configured to accept a packet, the firewall drops the packet and drops the failed access.

Next, the firewall examines the source address of the packet and the interface on which it received the packet. This process verifies the information against configuration tables, which prevents the firewall from accepting IP spoofed packets. If this check indicates that this request could not possibly have come in through this interface, it rejects the packet and logs it. For example, if the Yoyodyne firewall receives a packet on ec0 (the outside interface) claiming to be from 10.0.1.10 (an inside address), the firewall ignores the packet.

In our TELNET example, the destination of the packet is the firewall. The firewall receives request on ec0, the outside interface. The address does not indicate that it came from an inside network. The firewall accepts the packet for local delivery and processing.

Check Request Type

Now that the firewall is configured to deliver the packet locally, it looks at the contents of the packet. The operating system checks various tables on the firewall to determine if it offers the requested service on the requested port. If it does not, it logs the attempt as a potential security alert and rejects the request.

In our TELNET example, the packet indicates that it is a TELNET request on port 23. The configuration tables indicate that the firewall supports this type of service.

Run Appropriate Program

Now that the firewall is configured to offer the requested service, the operating system uses other configuration information to start the appropriate program. In our TELNET example, the firewall starts the TELNET proxy, which processes the TELNET request.

Process the Request

The proxy or application now processes the request. It first checks its configuration information. The proxy determines how to handle the request based on the source (IP address) of the request. By default, it uses one policy (set of rules) for trusted networks and another policy for untrusted networks.

Once configured, the proxy processes the requests as the standard application would. The proxies follow the same protocols and handshakes as indicated in the RFCs or other

documents. Requesting applications think they are talking to an actual server, not a proxy.

The proxies also check to determine if the request is permitted for the destination. For some services, the proxies can perform the additional step of authenticating the user. This verification provides additional assurance that the user is really who they says they are. The proxy then passes the request to the appropriate program on the other side of the firewall using the standard protocol for that service.

In our TELNET example, the TELNET proxy uses the generic outside policy because the request came from an outside network. The outside policy permits TELNET to internal machines, but requires authentication. The firewall prompts the user to authenticate. Once the user authenticates, the proxy provides a small menu allowing the user to indicate the internal machine to which they wish to connect. The proxy then uses standard TELNET protocol to pass packets back and forth between the host on the outside network and the host on the inside network.

PART TWO

Configuring and Using Proxies

Chapter 2

Managing SMTP Services

For many people, electronic mail is an integral tool for conducting business. Exchanging electronic mail is often the reason that sites decide they need to connect to the Internet. Such connections are not without risks.

The protocol for transferring mail around the Internet is the simple mail transport protocol (SMTP). The transfer requests are handled by a message transfer agent, such as the *sendmail* program used on IRIX systems. The *sendmail* program is large and requires many privileges. Our design philosophy of reductionism frowns upon the direct use of *sendmail* as a critical security component of the Gauntlet Firewall. The Gauntlet Firewall includes a two-part proxy that securely handles the transfer of SMTP mail between the inside and outside networks.

This chapter explains the concepts behind the proxy and how it works, how to configure the proxy for SMTP mail transfer, and how to configure these services to run through the firewall.

Understanding the Proxy

The proxy for SMTP is actually two different processes: a client (*smap*) and daemon (*smapd*). Together, they provide configurable access control and logging mechanisms. The processes, which run on the firewall, transfer mail between internal and external mail servers, based on rules you supply. You can also configure the message transfer agent that the firewall uses to deliver the messages to other hosts.

The proxies also prevent versions of *sendmail* on the inside network from talking with versions of *sendmail* on the outside network. The proxies log all successful and unsuccessful mail connections, and the number of bytes transferred.

How It Works

The firewall runs the client proxy (*smap*) as a daemon listening for requests on the standard SMTP port (25). When the firewall receives requests for SMTP services on this port, the *smap* client collects the mail from the sender, logs the message, and places the mail in a temporary directory. Periodically, based on a configurable value (by default every 60 seconds), the daemon (*smapd*) wakes up and checks to see if there is any new mail. The *smapd* daemon checks the headers of the mail for formatting problems. It then calls the configured message transfer agent (usually *sendmail* in delivery mode) for final delivery.

Both the *smap* client and the *smapd* daemon run using a user ID you specify, such as *uucp*. Rather than running as a root process as *sendmail* often does, the *smap* and *smapd* processes run with as few or as many privileges as you assign. In addition, both programs change their root directory to the transfer directory you specify.

A common policy is to have one mail hub for the inside network. In this scenario, outside networks know (via DNS) that they should send all mail for the domains (yoyodyne.com) on the inside networks to the firewall (firewall.yoyodyne.com) itself for processing. An outside host informs the firewall it has mail by connecting to *smap* on the SMTP port. The *smap* client collects the mail from the outside host and writes it to a directory (*/var/spool/smap*) on the firewall.

At some system administrator-configurable interval, the *smapd* daemon awakens and looks for new mail on the firewall. It parses the mail headers, and calls *sendmail* to deliver the messages. *sendmail* checks its configuration information, which tells it where to deliver mail. For example, its configuration files may tell it to deliver internal mail to an internal mail hub (mail.yoyodyne.com), in which case *sendmail* will transfer the mail to the mail hub using SMTP.

Configuring the Firewall for SMTP

Configuring the Gauntlet firewall involves planning, configuring the firewall, configuring the proxies to enforce your policy, advertising your mail exchanger, and configuring your internal mail hub.

Planning

1. Understand your existing mail configuration: hosts, hubs, and so on.
2. Plan *early* to make your DNS changes for mail records. This may require contacting an outside organization providing DNS service, such as an internet service provider (ISP). We cannot stress enough the importance of this step.

Configuring the Firewall

If you wish to allow SMTP traffic through the firewall, configure the firewall using the *gauntlet-admin* interface. The interface stores this information using *configmail* in conjunction with the auto-configuring version of the *sendmail.cf* configuration file.

To configure the firewall for SMTP, follow these steps:

1. Enter the external hostname of your firewall. For example, in Figure 1-2, the external hostname is the name assigned to the ec0 interface.
2. Enter the domain name of your firewall. For example, *yoyodyne.com*.
3. Enter the hostname or alias for all relay hosts. A relay is a host inside the firewall that determines where to send mail with an unknown address (you might have only one relay).
4. Provide subdomains to be recognized if you want outgoing mail addresses rewritten to keep subdomain information. The *sendmail* program transforms sender addresses from the *user@host.domain* format (penny@dimension.yoyodyne.com) into the *user@domain* format (penny@yoyodyne.com). Recognized subdomains will not be stripped off, so *user@host.corp.domain* is rewritten to *user@corp.domain* if *corp* is a recognized subdomain, or *user@domain* if *corp* is not a recognized subdomain.

Warning: This rewriting affects only certain sender lines (such as From:). It does not hide the names of your internal machines in the Received and other headers.

If you need an internal mail hub or multiple mail hubs, you must further customize the sendmail.cf file on the firewall so that it delivers inbound email to your hub or hubs instead of delivering the mail directly. Refer to the *IRIX Advanced Site and Server Administration Guide* for more information.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support SMTP traffic. This is a standard service, and you can use *gauntlet-admin* to modify the configuration files. If you need to, you can instruct *gauntlet-admin* to not make modifications so you can make the customizations for your site.

Configuring the Proxy Rules

You should not need to modify the proxy rules for SMTP services. If you do decide to modify */usr/gauntlet/config/template.netperm-table*, you may wish to add the *badadmin* attribute for debugging purposes. Information sent to this alias aids greatly in debugging mail delivery problems. See Appendix B, "Netperm Table" for more information on *smap* and *smapd* options, *netperm-table* options, and order of precedence.

Advertising the Firewall as a Mail Exchanger

You need to advertise the firewall as the mail exchange site for your domain. The DNS configuration in *gauntlet-admin* can do this for you. Consult the section on DNS configuration for specific instructions.

Configuring Your Internal Mail Hub

As long as you are using transparency to pass all packets for outside networks to the firewall, you do not need to configure your internal mail hub or mail agents. Because of the transparency, attempts to deliver to outside network hosts will be grabbed by the firewall.

If you are not using transparency, configure your internal mail hub to use the firewall as a mail forwarder, and direct clients to the internal mail hub. If you don't have an internal mail hub, configure the clients to use the firewall directly as a mail forwarder.

Verifying Your Setup

Verify your configuration by sending mail from an inside host to an outside host. Watch the logs on the firewall for error messages. Run *Mail* in verbose mode and send mail to the bouncing service listed below, which automatically generates a reply:

```
dimension-23: Mail -v bouncer@bbnplanet.com
Subject: Test Configuring Mail and the Gauntlet Firewall
This is a test.
^D
```

The verbose mode ensures that you see the details of the delivery. The bouncer service sends you a return message shortly.

If you need to test header rewriting or other custom configurations, consider starting *sendmail* in debug mode.

Using Mail

The firewall and the *smap* and *smapd* proxies for SMTP traffic are transparent to the user once the firewall, and possibly client machines, are configured

Chapter 3

Managing POP3 Services

The number and variety of computer systems at company sites today is expanding rapidly and, for a variety of reasons, it is not convenient to run a full mail transfer system using SMTP on these systems. The Post Office Protocol Version 3 (POP3) is one of the protocols that allow a workstation to access a mail server. The POP3 proxy included with the Gauntlet Firewall allows administrators to selectively allow outside hosts to exchange mail with a POP3 mail server through the firewall. The POP3 server must use APOP for authenticating the user.

This chapter explains the concepts behind the proxy and how it works, how to configure the proxy for POP3 mail transfer, and how to configure POP3 services to run through the firewall.

Understanding the Proxy

The Gauntlet POP3 proxy is an application-level gateway that provides configurable access control, authentication, and logging mechanisms. The POP3 proxy, which runs on the firewall, transfers mail between external workstations and internal mail servers, based on rules you supply:

- source IP address
- source hostname
- destination IP address
- destination hostname
- user name

Using these options, you can configure your firewall to allow specific hosts on outside networks to exchange mail with an internal mail server via POP3. For example, an employee working with a laptop PC running Windows™ needs to read mail while on travel. The employee can use the mail user agent (such as Eudora Pro™) on the laptop to collect their mail from the mail server inside the perimeter. The proxy uses the APOP

command (part of the POP3 protocol) for strong authentication. The proxy logs all successful and unsuccessful mail connections, and the number of bytes transferred.

You can manually configure the POP3 proxy to allow inside workstations to exchange mail with POP3 servers outside the perimeter. However, in most security policies (including the Gauntlet Firewall default), this is not considered a good idea. The POP3 protocol assumes that the SMTP proxy has already checked the formatting in the headers of incoming mail messages. In addition, allowing POP3 clients to communicate with outside mail servers adds another level of complexity. It bypasses the central control center of the inside mail hub, which rewrites addresses and enforces other company policies. Your mail server should be behind the firewall on the inside network. All POP3 clients on the inside network can collect their mail from this mail server.

How the POP3 Proxy Works

The firewall runs the POP3 proxy (*pop3-gw*) as a daemon listening for requests on the standard POP3 port (110). When the firewall receives requests for POP3 services on this port, the proxy checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use POP3 services. If the host does not have permission, the proxy logs the connection attempt and displays an error message.

If the host has permission, the POP3 proxy authenticates the user using APOP and logs the connection. The proxy then passes the message on to the POP3 server on the internal mail hub, and authenticates on behalf of the user using APOP. The proxy remains active until either side terminates the connection or Gauntlet times-out the connection.

The default Gauntlet policy allows users on outside (untrusted) hosts to connect to a specific internal mail server to collect mail. The firewall itself cannot run a POP3 server, because the POP3 proxy is running on the standard POP3 port.

Configuring the Firewall for POP3

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the proxy to enforce your policy, configuring your internal POP3 server, and creating APOP accounts for users who will need to authenticate.

Planning

Determine your policies for

- source and destination addresses
- user access to POP3

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support POP3 traffic.

Configuring the Proxy Rules

If you are using the Gauntlet Firewall default configuration, you need to modify the proxy rules for POP3 services. This involves accessing the *gauntlet-admin* Proxies form, where you can enter the name of the destination POP3 server and modify the timeout value if you desire. See Appendix B for more information on *pop3-gw* options, `netperm-table` options, and order of precedence

Configuring Your Internal POP3 Mail Server

Configure your internal POP3 mail server:

1. Configure your POP3 mail server to accept POP3 requests from the firewall. If you need to specify an IP address, remember to use the internal IP address for the firewall.
2. Ensure that the POP3 mail server is using the POP3 port (110).
3. Configure your POP3 mail server to support APOP.
4. Configure the APOP password for each user.

Setting APOP Passwords on the Firewall

Use the authentication management system to add users to the Gauntlet user authentication database for any users who need to authenticate when using POP3 services. See Chapter 18, “Creating Users” on page 181 for details.

Verifying Your Setup

Verify your setup by retrieving mail (using POP3) from a host outside the perimeter. See “Using POP3 to Exchange Mail” on page 28 for instructions.

Using POP3 to Exchange Mail

Because the POP3 proxy requires authentication, users must follow different procedures to use POP3 services.

To retrieve electronic mail using POP3 with authentication, follow these steps:

Note that the order of these steps may differ for different user agents.

1. Configure the mail user agent and set the name of the POP3 server to the firewall.
2. Retrieve mail, causing the user agent to connect to the firewall.
3. Authenticate to the proxy by supplying your APOP password.
4. Continue as though the firewall were not there.

The example below shows a user named John working on an outside network who needs to retrieve mail from the mail server on the inside network.

First, John configures his mail reader to get his mail via POP3 from the firewall. Figure 3-1 shows the configuration screen for Eudora Pro for Windows, a popular mail application.

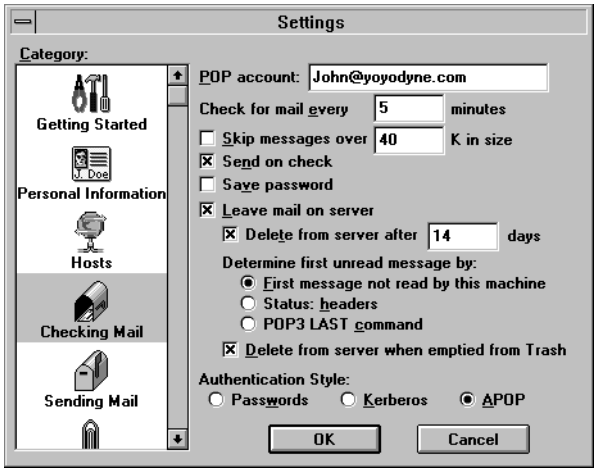

Figure 3-1 Eudora Pro Configuration for APOP

John, working on his laptop (cavalier.yoyodyne.com) at home, configures his mail reader to connect to the firewall (firewall.yoyodyne.com) to get his mail.

Next, John retrieves his mail. As part of the connection, the proxy requests authentication information from the user agent, which prompts him.

After authenticating, the proxy transfers the request to the internal POP3 mail server (mail.yoyodyne.com), authenticates using the user's POP password as stored on the firewall, and retrieves his mail.

Chapter 4

Managing Terminal Services

Terminal service access to other computers can be a vital part of many network activities. The TELNET and rlogin protocols are used for making these terminal connections, and they are not without risk. The Gauntlet Firewall includes proxies for both the TELNET and rlogin protocols, which securely handle terminal services between the inside and outside networks.

This chapter explains the concepts behind the TELNET and rlogin proxies and how they work, how to configure the proxies, and how to use terminal services.

Understanding the Proxies

The Gauntlet TELNET and rlogin proxies are application-level proxies that provide configurable access control, authentication, and logging mechanisms. The TELNET and rlogin proxies, which run on the firewall, pass TELNET and rlogin requests through the firewall, using rules you supply. The TELNET proxy also passes TN3270 requests through the firewall. You can configure the proxies to allow connections based on

- source IP address
- source hostname
- destination IP address
- destination hostname

Using these options, you can configure your firewall to allow specific hosts on outside networks to connect to inside hosts or vice versa. Employees working at customer sites can access their workstations inside the perimeter.

The strong authentication features of the proxies allow administrators to require users to authenticate before connecting. The proxies log all successful and unsuccessful connection attempts, and the amount of data transferred.

Used together, these access controls and log files allow you to have much more control over the connections to and from your system than you have when you use the standard IRIX TELNET and rlogin programs.

Note that you can use the TELNET proxy without the rlogin proxy, or rlogin without TELNET. You can configure different policies for hosts and authentication, as well.

How the Proxies Work

In the default configuration, the IRIX system runs the network access control daemon (*netacl*) as a daemon listening for requests on the standard TELNET port (23). Whenever the firewall receives a TELNET request on this port, the *netacl* daemon checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use TELNET. If the host has permission, the *netacl* daemon starts the standard TELNET program (*telnetd*) or the TELNET proxy (*tn-gw*), depending upon the originating host. If the host does not have permission, the daemon displays an error message. Similarly, the *netacl* daemon running on the standard login (513) starts either the rlogin daemon (*rlogind*) or the rlogin proxy (*rlogin-gw*).

The default policy for this scenario is to allow all inside hosts to initiate TELNET or rlogin sessions without authenticating. The inside host passes TELNET requests to the firewall, which starts the *netacl* daemon. The *netacl* daemon checks its permissions, and determines that the inside host can use TELNET. The *netacl* daemon starts the proxy. The proxy logs the transaction and passes the request to the outside host. The proxy remains active until either side closes the connection.

The default policy for this scenario allows outside hosts to initiate TELNET or rlogin sessions after authenticating. The outside host passes TELNET requests to the firewall, which starts the *netacl* daemon. The *netacl* daemon checks its permissions, and determines that the outside host can use TELNET. The *netacl* daemon starts the proxy. The proxy prompts the user for authentication. If it is successful, the proxy prompts the user for the inside host, logs the transaction, and passes the request to the inside host. The proxy remains active until either side closes the connection.

Note that users are not logging into the firewall directly. While users use the proxy on the firewall for authentication, the proxy simply passes the user's TELNET or rlogin session on to the appropriate host.

If you need to log in remotely to the firewall, you must use *netacl* to start the proxies. In this configuration, administrators on either inside or outside hosts initiate TELNET requests to the firewall, which accesses the netacl daemon. The *netacl* daemon checks its permissions, and determines that the host can use TELNET. The *netacl* daemon starts the proxy. The proxy prompts the user for authentication. If it is successful, the proxy prompts the user for the host and logs the transaction. When the user indicates a wish to connect to the firewall itself (by specifying the destination "localhost"), the *netacl* daemon reviews the destination and starts the actual IRIX TELNET daemon on the firewall, thereby connecting the user to the firewall.

Using the TELNET and Rlogin Proxies Without Network Access Control

In this scenario, the firewall runs the TELNET (*tn-gw*) or Rlogin (*rlogin-gw*) proxies as daemons listening for requests on the standard TELNET port (23) and Rlogin port (513). Common policies allow inside hosts to connect without authentication, and outside hosts to connect with authentication.

This configuration using just the TELNET and Rlogin proxies (without the *netacl* daemon) prohibits running either TELNET or Rlogin on the firewall itself (which would allow you to login to the firewall remotely). Because the proxies are running on the standard TELNET and Rlogin ports on the firewall, all requests start the proxy. There is no way to start the TELNET and Rlogin daemons needed to service these requests on the standard ports.

Configuring the Firewall for Terminal Services

Configuring the Gauntlet firewall involves planning, configuring the firewall, indicating which daemons the system will run, configuring the proxies to enforce your policy, and adding the users who will need to authenticate to the Gauntlet user authentication database.

Planning

1. Determine whether you wish to allow TELNET and TN3270 connections through the firewall.
2. Determine whether you wish to allow rlogin connections through the firewall.

3. Determine whether you wish to allow remote access to the firewall itself. Working from the physical firewall console is more secure than connecting from another host on a network. If you work remotely to administer the firewall, you risk disclosure of the user authentication management database and disclosure of the authentication passwords. However, when circumstances sometimes prohibit physical access to the firewall, the firewall can be configured to allow remote access.
4. Determine your policies for authentication.

Configuring the Firewall

If you wish to allow remote system administrator login to the firewall itself, configure the firewall using the *gauntlet-admin* interface to permit remote logins.

This setting actually changes the settings in the *netperm-table* file so that the TELNET and rlogin proxies will start the actual TELNET and rlogin daemons when you try to connect to the firewall itself using the “localhost” host name.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support TELNET or rlogin traffic.

Configuring the Proxy Rules

If you are using the Gauntlet Firewall default configuration, you do not need to modify the proxy rules for TELNET, rlogin and TN3270services. If you have chosen different welcome or other messages, you must modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B, “Netperm Table” for more information on *tn-gw* and *rlogin-gw* options, *netperm-table* options, and order of precedence.

Note that the settings for the TELNET proxy (*tn-gw*) affect both TELNET and TN3270 access through the firewall.

Creating Authentication User Entries

Use the authentication management system to add users to the Gauntlet user authentication database for any users who need to authenticate when using TELNET and rlogin services. See Chapter 18, “Creating Users” on page 181 for more information.

Verifying Your Setup

Verify your configuration by connecting to an inside host from an outside host. See the section below for instructions.

Using Terminal Services

TELNET, Rlogin, and TN3270 Without Authentication

You can configure the proxies so that they are transparent to your users. Enable transparent proxies using *gauntlet-admin* to configure the proxies so that users working on the trusted networks behind the firewall do not see a change in their daily TELNET, rlogin, and TN3270 activities. For example, a transparent TELNET through firewall.yoyoyne.com might look like this:

```
dimension-26: telnet blaze.clientsite.com
Trying 10.0.2.120 port 23...
Connected  to blaze.clientsite.com
BSDI BSD/OS 2.0.1 (blaze) (ttyp5)
login:
```

If you do not enable transparent proxies for terminal services, or if you require user authentication, users must first access the corresponding firewall proxy with a terminal service and, once established on the firewall, they may then connect to a host outside. The next section describes how to connect through the firewall when user authentication is in force.

TELNET and Rlogin With Authentication

If you have configured any terminal services to require authentication, users must follow different procedures to use TELNET or rlogin.

For example, to TELNET using authentication, follow these steps:

1. TELNET to the firewall itself.
2. Authenticate to the proxy.
3. Connect to the desired host.
4. Continue as before.

The default policy for the TELNET proxy is to authenticate all requests from untrusted networks to or through the firewall. The example below shows a sample TELNET session from an untrusted network to a trusted network, using S/Key authentication at the firewall.

```
blaze.clientsite.com-28: telnet firewall.yoyodyne.com
Trying 204.255.154.100...
Connected to firewall.yoyodyne.com

Escape character is '^]'.
Username: scooter
Skey Challenge: s/key 651 fi19289 SAFE DUB RISK CUE YARD NIL

Login Accepted
firewall.yoyodyne.com telnet proxy (Version 3.1) ready:

tn-gw> c dimension
Trying 10.0.1.120 port 23...
Connected to dimension.yoyodyne.com
BSDI BSD/OS 2.0.1 (dimension) (ttyp5)
login: scooter
Password: #########

Welcome to dimension.yoyodyne.com
3:57PM up 16 days, 5:35, 4 users, load averages: 0.03, 0.01, 0.00
dimension-26:
```

In this example, Scooter, working at a client site (blaze.clientsite.com), needs TELNET access to the dimension.yoyodyne.com system behind the firewall. He first telnets to the firewall for Yoyodyne (firewall.yoyodyne.com). The TELNET proxy on firewall prompts him to authenticate. Scooter provides his authentication user ID (scooter). When the proxy prompts, he enters the response to the authentication challenge. The proxy authenticates scooter.

Scooter now indicates the host he needs to access (dimension). The TELNET proxy connects Scooter to dimension, and the TELNET daemon running on that machine. The TELNET daemon on dimension prompts Scooter for his user name and password on dimension. The TELNET daemon on dimension verifies Scooter's user name and password, and logs him in.

TN3270 With Authentication

If you have configured terminal services to require authentication, users need to follow different procedures to use TN3270. To use TN3270 with authentication:

1. TN3270 to the firewall itself, disabling true TN3270 support for the initial handshake
2. Authenticate to the proxy
3. Connect to the desired 3270 host
4. Continue as before

The corporate policy that requires authentication before using TELNET from untrusted hosts to trusted hosts also applies to using TN3270. Generally, the only difference is in starting the TN3270 client:

```
blaze-55: X3270 -model 2 -efont 3270-12 a: fire-out.yoyodyne.com
```

Chapter 5

Managing FTP Services

Sometimes the easiest way to transfer information from one machine to another is to actually transfer the relevant files. The file transfer protocol (FTP) is one of several protocols that make this possible. The Gauntlet firewall includes a proxy that securely allows the transfer of files between trusted and untrusted networks.

This chapter explains the concepts behind the FTP proxy and how it works, how to configure the proxy, and how to use FTP services. A section also discusses considerations for running anonymous FTP servers.

Understanding the FTP Proxy

The Gauntlet FTP proxy is an application-level proxy that provides configurable access control, authentication, and logging mechanisms.

The FTP proxy, which runs on the firewall, passes FTP requests through the firewall, using rules you supply. You can configure the FTP proxy to allow file transfer activity based on

- source IP address
- source hostname
- destination IP address
- destination hostname
- FTP command (for example, STOR and RETR)

Using these options, you can configure your firewall to allow specific hosts on outside networks to transfer files to and from inside hosts. Employees working at specific customer sites can access files on their workstations. Similarly, you can configure your firewall to permit users on the inside network to copy files (using the FTP daemon RETR command) from hosts on the outside network, but not place files (using the FTP daemon STOR command) on these outside hosts.

The strong authentication feature of the FTP proxy allows administrators to require users to authenticate before transferring files. The FTP proxy logs all successful and unsuccessful file transfer attempts, and the number of bytes transferred.

Used together, these access controls and log files allow you to have much more control over the files entering and leaving your system than using the standard IRIX FTP programs.

How the FTP Proxy Works

In this most common scenario, the firewall runs the network access control daemon (*netacl*) as a daemon listening for requests on the standard FTP port (21). Whenever it receives an FTP request on this port, the *netacl* daemon checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use FTP. If the host has permission, the *netacl* daemon starts the standard FTP server (*ftpd*) or the FTP proxy (*ftp-gw*). If the host does not have permission, the daemon displays an error message.

The default policy for this scenario is to allow all inside hosts to initiate FTP sessions and transfer files without authenticating. The inside host passes FTP requests to the firewall, which starts the *netacl* daemon. The *netacl* daemon checks its permissions, and determines that the inside host can use FTP. The *netacl* daemon starts the *ftp-gw*. The proxy logs the transaction and passes the request to the outside host. The *ftp-gw* remains active until either side terminates the connection. The default policy also allows outside hosts to initiate FTP sessions. However, they must authenticate before accessing inside hosts.

The default policy does not allow either inside or outside hosts to FTP directly to the firewall itself. If you configure your Gauntlet firewall to allow anonymous FTP to the firewall, hosts connect to the firewall with an FTP request. The firewall starts the *netacl* daemon. The *netacl* daemon checks its permissions, and determines that outside hosts can use FTP to the firewall itself. The *netacl* daemon starts the standard FTP daemon (in a chrooted environment).

This configuration using *netacl* allows a fair amount of flexibility in configuring FTP services. Users inside the perimeter can continue to interact with outside hosts, generally without authentication. Users outside the perimeter can interact with inside hosts, generally with authentication.

Configuring the Firewall for FTP Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the FTP proxy to enforce your policy, and creating user accounts for users who will need to authenticate.

Planning

1. Determine whether you wish to allow outside hosts to FTP through the firewall to inside hosts or to the firewall itself. This decision will determine whether or you need to use the network access control daemon.
2. Determine your policies for
   - requiring authentication
   - allowing specific FTP commands (for example, RETR and STOR)
   - permitting or denying specific sources and destination

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support FTP traffic.

Configuring the Proxy Rules

If you are using the Gauntlet Firewall default configuration, you do not need to modify the proxy rules for FTP services. Use the *gauntlet-admin* Proxies form if you want to enable FTP or anonymous FTP. If you have chosen a different denial message, you must modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on *ftp-gw* options, *netperm-table* options, and order of precedence.

Creating Authentication User Entries

Use the authentication management system to add users to the Gauntlet user authentication database for any users who need to authenticate when using FTP services. See “Creating Users” on page 181 for more information.

Verifying Your Setup

Verify your configuration by transferring files to an inside host from an outside host. For example, connect to your favorite FTP site and download their *README* file. See the section below for instructions.

Using FTP Services

The idea behind the FTP proxy is that most users working on the trusted networks behind the firewall will not see a change in their daily FTP activities. The default policy allows users on trusted networks to FTP to untrusted networks without authenticating. Users on the trusted networks do not need to change their FTP procedures.

Using Authentication

If you have configured any FTP activities to require authentication, users must follow different procedures to use FTP.

To FTP using authentication, follow these steps:

1. FTP to the firewall itself.
2. Authenticate to the proxy.
3. Connect to the desired FTP server.
4. Continue as before.

A common policy for the FTP proxy is to authenticate all requests from untrusted networks to or through the firewall. The example below shows a sample FTP session from an untrusted network to a trusted network, using S/Key authentication at the firewall.

```
blaze.clientsite.com-27: ftp firewall.yoyodyne.com
Connected to firewall.yoyodyne.com
220-Proxy first requires authentication

220 firewall.yoyodyne.com FTP proxy (Version 3.1) ready.
Name (firewall.yoyodyne.com:clancy): clancy
331 Skey Challenge: s/key 653 fi19289
Password: <password does not display>
230 User authenticated to proxy
```

```
ftp> user clancy@dimension
331- (-----GATEWAY CONNECTED TO dimension----)
331- (220 dimension FTP server ready.)
331 Password required for clancy.
Password: #########
230 User clancy logged in.
ftp>
```

In this example, the user Clancy, working at a client site (blaze.clientsite.com), needs FTP access to a machine behind the firewall (dimension.yoyodyne.com). He first FTPs to the firewall for Yoyodyne (firewall.yoyodyne.com). The FTP proxy on firewall prompts him to authenticate. Clancy provides his authentication user ID (clancy). When the proxy prompts, he enters the response to the authentication challenge, which does not display. The proxy authenticates clancy.

Clancy indicates the host he needs to access and his user name for that host (clancy@dimension). The FTP proxy connects Clancy to dimension and prompts him for his password on dimension. Clancy enters his password for dimension. The FTP server on dimension verifies Clancy's user name and password, and logs him in. Clancy can now transfer files using regular FTP commands.

Using Authentication With Some GUI FTP Tools

The FTP proxy can require you to authenticate twice. Some GUI FTP tools for Microsoft Windows™ and the Macintosh® require you to specify the user name and password in a dialog box. These tools assume that once you supply this information, you are connected.

The FTP proxy displays the challenge and response information for authentication in FTP comments. Some Microsoft Windows and Macintosh operating system FTP tools do not display FTP comments. Unless users see the comment, they will have a really difficult time trying to guess the current challenge. You can still use these FTP tools with S/key authentication, by combining the authentication and FTP host information.

To authenticate using some GUI tools, follow these steps:

1. For the hostname, supply the name of the firewall.
2. For the user name, supply the firewall authentication user name, the FTP host user name, and the name of the FTP host, in the form

   authentication-username@ftp-host-username@ftp-host.

3. For the password, supply the authentication response and the FTP host password in the form

   authentication-response@ftp-host-password

   You may need to TELNET to the firewall to see what the next challenge is.

The example below shows the information a user would enter in their FTP tool when going from an untrusted network to a trusted network, using S/Key authentication for the firewall:

```
host:firewall.yoyodyne.com
username:clancy@clancy@dimension
password:elk elba iris odd skim lee@#########
```

Because you cannot tell what the next challenge will be when using most other challenge-response authentication mechanisms, you may not be able to use these instructions with some GUI FTP tools.

Running an Anonymous FTP Server

By its very nature, an anonymous FTP server requires easy access by the public. If you place the anonymous FTP server behind the firewall, you are allowing an additional type of access within your security perimeter. If you place the FTP server on the firewall itself, you are allowing additional access to your firewall. Evaluate both setups for the possible security risks to your site and how your site security policy addresses this type of access.

Gauntlet for IRIX allows you to run the standard IRIX FTP server (*ftpd*) in an isolated chrooted environment as an anonymous FTP server (but you give up the ability to allow authenticated users from untrusted networks to use *ftp-gw* to access trusted networks).

The best solution is generally to place your anonymous FTP server on a machine outside the perimeter. Follow good host-oriented security practices for this machine:

- Turn off all other services.
- Create the minimum number of user accounts.
- Use strong authentication.
- Patch your operating system and applications with all current security patches.

- Use checksums to watch for file changes.
- Back up frequently.

You can also use the Info Server included with the Gauntlet firewall as an anonymous FTP server on the firewall itself. See “FTP Server” on page 102 for more information.

Chapter 6

Managing Rsh Services

Administration and support activities can be easier when you can just execute a shell on a remote machine. The Rsh service allows users to do this. The Rsh program is not without risks: it runs programs on another machine and requires some privileges to login. The Gauntlet firewall includes a proxy that securely handles the execution of Rsh requests from machines inside the network to machines outside the network.

This chapter explains the concepts behind the *Rsh* proxy and how it works, how to configure the proxy, and how to use Rsh services.

Understanding the Rsh Proxy

The Gauntlet *Rsh* proxy is an application level gateway that provides configurable access control, authentication and logging mechanisms. The *Rsh* proxy, which runs on the firewall, passes *Rsh* requests through the firewall, using rules you supply. You can configure the *Rsh* proxy to allow remote shell activity based on:

- source IP address
- source host name
- destination IP address
- destination host name

Using these options, you can configure your firewall to allow specific hosts on the inside network to start remote shells on outside hosts. Employees working behind the firewall can start remote shells on outside hosts at a customer site. The *Rsh* proxy logs all successful and unsuccessful remote shell attempts, and the number of bytes transferred.

These access controls allow you to have much more control over the Rsh requests entering and leaving your system than using the standard UNIX *Rsh* program. The logging capabilities are also much more extensive.

How It Works

In this default configuration, the firewall runs the *Rsh* proxy (*rsh-gw*) as a daemon listening for requests on the standard Rsh port (514). Whenever the system receives an *Rsh* request on this port, the *Rsh* proxy checks its configuration information (in the *netperm-table*) and determines whether the initiating host has permission to use *Rsh*. If the host has permission, the proxy logs the transaction and passes the request to the outside host. The *rsh-gw* remains active until either side closes the connection.

The default policy for this scenario allows inside hosts to *Rsh* without authenticating. Users on inside hosts can continue to *Rsh* as they did before the firewall was put into place. The default policy does not allow outside hosts to *Rsh* to hosts inside the perimeter.

The default policy and configuration using just the *Rsh* proxy prohibit running an *Rsh* server on the firewall itself. Because the *Rsh* proxy is running on the standard *Rsh* port on the firewall all *Rsh* requests start the proxy. There is no way to start the *Rsh* daemon needed to service *Rsh* requests.

Configuring the Firewall for Rsh Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the *Rsh* proxy to enforce your policy, turning on the proxy, and rebooting your firewall.

Planning

Determine whether you wish to allow inside hosts to *Rsh* through the firewall to outside hosts.

Configuring Network Services

You do not need to modify the UNIX configuration files on the firewall to support *Rsh* traffic.

Configuring the Proxy Rules

Configure the *Rsh* proxy to enforce your security policies. This involves modifying */usr/local/etc/netperm-table*. See Appendix B for more information on *rsh-gw* options, *netperm-table* options and order of precedence. To configure the *netperm-table:*

1. Add the Rsh proxy to your trusted policies, as appropriate.

   ```
   policy-trusted:permit-proxy rsh-gw
   ```

2. Configure other Rsh proxy options, as appropriate for your setup. These could include the default directory and timeout values.

   ```
   #Rsh proxy rules
   rsh-gw: timeout 300
   ```

Verifying Your Setup

Verify your configuration by accessing a machine outside the perimeter from a machine inside the perimeter.

Using Rsh Services

Following some initial configuration, the firewall and the *rsh-gw* proxy are transparent to the user. Users can continue to use *rsh* to outside hosts as they did before.

Configuring the Remote Machine

Before using Rsh, users must add their user name and the name of the firewall to their *.rhosts* file on the remote machine:

user@firewall

where:

1. *user* is their user name within the domain from which the request comes. The user does not actually need to have an account on the firewall itself. The Rsh request simply appears to be coming from the firewall.
2. *firewall* is the name (including domain if necessary) of the firewall. This name should be the name of the interface on firewall closest to the remote machine.

For example, Penny, who works at Yoyodyne, needs to execute something remotely using her account at Big University. She adds a line to the *.rhosts* file in her account at Big University:

```
penny@fire-out.yoyodyne.com
```

Chapter 7

Managing Gopher and WWW Services

What can we say about the World Wide Web? Your users probably argue that they really need it to do their jobs. There is a vast wealth of information stored on machines connected the Internet. The graphical interfaces of browsers and web pages make it much easier to access and digest this information. Along with this ease can come problems. World Wide Web (WWW) services allow for the transfer of a wide variety of file types and for running a number of different programs. This complexity means a greater potential for problems, especially in terms of security. These services are generic file transfer mechanisms and require logging and access control consistent with FTP and terminal services.

The HTTP proxy and authenticating HTTP proxy included with the Gauntlet Firewall securely handles requests for information via hypertext, Gopher, and file transfer. The proxy supports hypertext transfer via the HTTP, SHTTP, and SSL protocols; Gopher transfer via Gopher and Gopher+ protocols; and file transfer via FTP.

This chapter explains the concepts behind the HTTP proxy and how it works; how to configure the proxy for web services, Gopher services, and file transfer services; and how to configure these services to run through the firewall. In addition, it includes information on running HTTP and Gopher servers.

Understanding the Proxy

The Gauntlet HTTP proxy is an application-level proxy that provides configurable access control and logging mechanisms. The HTTP proxy, which runs on the firewall, passes HTTP, SHTTP, SSL, and Gopher requests, and FTP URLs and selectors through the firewall, using rules you supply. You can configure the proxy to allow connections based on

- source IP address
- source hostname

- destination IP address
- destination hostname

Using these options, you can configure your firewall to allow clients on the inside network to access Gopher sites on the outside network. You can also limit the web sites your employees can access from machines on the inside network. The proxies log all successful and unsuccessful connection attempts, and the amount of data transferred.

The Gauntlet authenticating HTTP proxy works in conjunction with the HTTP proxy to authenticate users. Using the authenticating HTTP proxy, you can configure the proxy to allow connections based on username. You can require all users to user strong or weak authentication before accessing information on the outside network.

You can configure the HTTP proxy to allow outside hosts to access web and Gopher servers behind your firewall on inside networks. However, in most security policies (including the Gauntlet Firewall default), this is not considered secure. By design, these services require easy access by people all over the Internet, and having a separate host outside the firewall is best. See the section on "*Configuring the Firewall for WWW and Gopher Services*" at the end of this chapter.

How It Works

The IRIX system runs the HTTP proxy as a daemon listening for requests on the HTTP port (8080) and / or the gopher port. When the firewall receives requests for services (via HTTP, SHTTP, SSL, Gopher, or Gopher+), the proxy looks at the request and places it in one of several categories. The proxy then checks the appropriate configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use the desired service to the desired destination. If the host does not have permission, the proxy logs the connection and displays an error message.

If the host has permission, the *http-gw* proxy passes the request to the desired host using the standard port (or the port specified in the request). As the outside host returns data to the requesting client, the firewall translates the data into the form the client expects and returns the data to the client. The proxy remains active until either side terminates the connection.

The default configuration for HTTP requests allows all inside hosts to access any WWW sites. In this scenario, the web browser on the inside host passes a request with a URL for a particular web page to the firewall on port 80. The request is received by the *http-gw*

proxy. The proxy examines the request and determines that it is a basic request for HTTP service. The proxy checks the source and destination ports in the *netperm-table* file. It then sends the request to the web server specified in the URL. When it receives the requested data, it passes the data back to the requesting web browser.

If the request is for Gopher or FTP services (from a Web or Gopher client), it is still the *http-gw* proxy which receives the request, and it still uses the *http-gw* rules.

If the request is for some sort of secure HTTP transaction using either the SHTTP or SSL protocols, the proxy performs the appropriate hand-off with the secure server at the other end of the connection.

If you have not configured or can not configure the web browser to know about the HTTP proxy, the firewall still calls the HTTP proxy for requests on port 80. However, it does not handle requests for services on other ports (for example, 8080). You can, however, run a second or even a third HTTP proxy on popular alternate ports.

Authenticated HTTP

If you want to authenticate users before allowing them to access information, the firewall runs the authenticating HTTP proxy (*ahttp-gw*) as a daemon listening for requests on the HTTP port (8080). When the firewall receives requests for service on this port, it performs the normal configuration checks to ensure that the initiating host has permission to use the desired service to the desired destination.

If the host has permission, *ahttp-gw* prompts the user to authenticate. It verifies the information with Gauntlet authentication database. If the user provided proper authentication, *ahttp-gw* passes processing over to the HTTP proxy.

The proxy remains active as long as a persistent connection between the source and destination remains. Each time the connection breaks (due to inactivity, pressing the stop button, or selecting a link before the initial page finishes loading, or any other reason), the *ahttp-gw* proxy reauthenticates you. If you are using reusable passwords, your browser remembers this information and reauthenticates on your behalf. If you are using strong authentication, you must reauthenticate each time the connection breaks.

Gopher and FTP Services

If the request is for Gopher services (from a Web or Gopher client), the firewall calls a second copy of the *http-gw* proxy, running as *http-gw* on port 70. It still uses the *http-gw* rules in the *netperm-table*.

If the request is for FTP services (from a Web client), the firewall still calls the *http-gw* proxy and uses the *http-gw* rules in the *netperm-table* if you have your FTP proxy set to the HTTP proxy. If you have not set an FTP proxy in your Web browser, the FTP proxy (*ftp-gw*) handles requests for FTP service.

SHTTP and SSL Services

If the request is for some sort of secure HTTP transaction using either the SHTTP protocol (on port 8080) or SSL protocol (on port 443), the proxy performs the appropriate hand-off with the secure server at the other end of the connection.

If you have not configured or can not configure the web browser to know about the HTTP proxy as the security proxy, the firewall calls the SSL plug proxy for all requests on port 443.

Configuring the Firewall for WWW and Gopher Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, and configuring the proxies to enforce your policy.

Planning

1. Determine which services you will allow.
2. Determine your policies for source and destination sites.
3. Determine whether you wish to require authentication.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support HTTP, SHTTP, SSL, Gopher, or FTP.

Configuring the Proxy Rules

If you are using the Gauntlet Firewall default configuration, you do not need to modify the proxy rules for HTTP and Gopher services. If you have chosen other options, you must modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on *http-gw* options, *netperm-table* options, and order of precedence. Remember that the HTTP proxy uses its own rules for FTP transfers. If you have denied a particular site for the FTP proxy, you will want to deny it for the HTTP proxy as well.

Creating User Authentication Entries

Use the authentication management system to create authentication user entries for any users who authenticate when using the authenticating HTTP proxy. See Chapter 17 for more information. Consider using multiple authentication servers (as explained on page 6) if you wish to require strong authentication for other inbound services and weak authentication for outbound HTTP requests.

Verifying Your Setup

Verify your setup by connecting to some of your favorite WWW, Gopher, and FTP sites. Connect to secure Web sites as well. See the section below for specific configuration instructions.

Using Web Services

Once you have configured a proxy-aware Web browser, the HTTP proxy is generally transparent to the user. When using a browser that does not support proxies, users need to modify their activities.

If you are using the authenticating HTTP proxy, users must use a proxy-aware browser. It must support persistent connections if you wish to use strong authentication. Once you have configured their web browser, they are aware of the proxy because they must authenticate to access outside sites.

Using Proxy-Aware Browsers

Many Web browsers, such as Netscape and Mosaic are aware of application proxies for different types of Web services. Once you configure these browsers, the browser sends the request to the appropriate proxy.

If you are using the authenticating HTTP proxy, ensure that the browser supports proxy authentication and persistent connections.

Configuring Web Browsers

The steps vary depending upon the browser, operating system, and version. Some browsers allow you to indicate the information using a dialog box from a preferences menu, while others require you to edit a configuration file, and others use environment variables.

If you are using the authenticating HTTP proxy, ensure that the browser supports proxy authentication and persistent connections.

To configure the browser, follow these steps:

1. Specify that you can only have one network connection at a time if you are using the authenticating HTTP proxy with strong authentication.
2. Specify the name of the firewall for the HTTP proxy and port 8080 as the HTTP port.
3. Specify the name of the firewall for the Gopher proxy and port 8080 as the Gopher port.
4. Specify the name of the firewall for the FTP proxy and port 8080 as the FTP port. Note that this is not the standard FTP port 23. When the firewall receives an FTP request on port 8080, the *http-gw* proxy does the actual FTP processing, not the *ftp-gw* proxy. This is because Web browsers use the HTTP protocol to communicate with the firewall proxy, not the FTP protocol.
5. Specify the name of the firewall for the security proxy and port 8080 as the security port.

6. Specify the names of hosts for which you do not want to access the HTTP proxy in the No Proxy section. These are generally hosts on your trusted networks. These include:
   - inside IP address of your firewall (if you plan to use the graphical user interface to configure your firewall)
   - host names of any internal or corporate HTTP servers
   - localhost (127.0.0.1)

Note that if you use the IP address instead of the hostname, you must use the internal IP address of the firewall.

Figure 7-1 shows the configuration screen for version 2.0 of Netscape Navigator™ for Microsoft Windows.

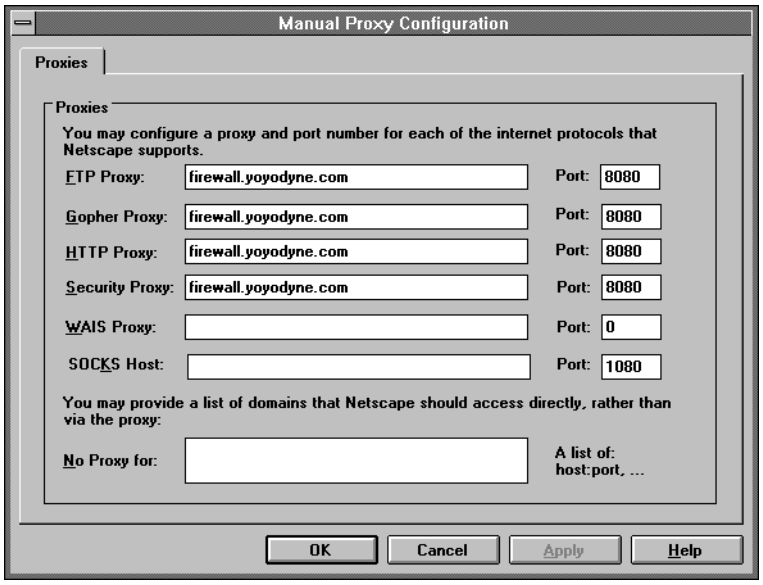

Figure 7-1 Proxy Configuration for Netscape Navigator 2.0 for Windows

Accessing Web Services Without Authentication

Once configured, the proxy is transparent to the user. Users can continue to access the Web as they did before.

If you have configured the proxies to block certain types of services (for example, no Gopher services) or to block certain destinations (for example, no educational [.edu] sites) users do see your denial messages.

Accessing Web Services with Authentication

Once configured, users are aware of the proxy. In a particular session, the proxy prompts for authentication the first time you attempt to access a site on the outside network.

To use web service using weak authentication:

1. Open a URL
2. Authenticate to the proxy
3. Continue as before

Weak Authentication

If you are using weak authentication, enter your username and password when your browser prompts you to. The proxy remembers this information and reauthenticates you if the connection breaks.

Strong Authentication

If you are using strong authentication, enter your username when your browser prompts you to. The proxy uses your user name to determine the type of authentication you are using. It prompts you a second time with the appropriate challenge. Enter your username and your response. Be prepared to reauthenticate each time your connection breaks.

Using Non-Proxy-Aware Browsers

Some older Web browsers are not aware of proxies. Using these browsers, you must explicitly send your requests through the firewall.

Configuring Web Browsers

The steps vary depending upon the browser, operating system, and version.

To configure the browser, set up the default home page as the name of the firewall, using the internal firewall name, for example:

```
http://firewall.yoyodyne.com:8080
```

Accessing Web Services

For regular use of a web browser, if you cannot create a default home page, prefix each URL you enter with the internal name of the firewall and the proxy port. For example:

```
http://www.clientsite.com
```

becomes

```
http://firewall:8080/http://www.clientsite.com
```

where firewall is the hostname of the firewall (firewall.yoyodyne.com). You must also prepend all saved URLs in bookmarks and hotlists.

Using Gopher Services

The firewall configuration for the *http-gw* proxy for Gopher services is transparent to the user if transparent proxies have been enabled using *gauntlet-admin*. Users can continue to point their Gopher clients to Gopher servers as they did before.

If you have disabled transparent proxies, then users must rewrite each Gopher address. If a user has a set of bookmarks for Gopher servers that was created before you installed the firewall, the user must modify the bookmark information to include the name of the firewall. For example:

```
name: Big University Gopher Server
host: gopher.bigu.edu
port: 70
path:
```

becomes

```
name: Big University Gopher Server
host: firewall.yoyodyne.com
port: 8080
path: gopher://gopher.bigu.edu:70
```

Running a WWW Server

By its very nature, a WWW server requires easy access by the public. If you place the WWW server behind the firewall, you are allowing an additional type of access within your security perimeter. If you place the WWW server on the firewall itself, you are allowing additional access to your firewall. Furthermore, *most* WWW servers are large and complicated pieces of software, and running such software on the firewall increases the likelihood that someone may be able to exploit bugs in the WWW server to break into your firewall

The best solution is generally to place your WWW server on a separate machine outside the perimeter. Follow good host-oriented security practices for this machine:

- Turn off all other services.
- Create the minimum number of user accounts.
- Use strong authentication.
- Patch your operating system and applications with the latest patches, especially security patches.
- Use checksums to watch for file changes.
- Back up frequently.

You can also use the Info Server included with the Gauntlet firewall as a WWW server on the firewall itself. See Chapter 15, “Managing Information Services on the Firewall,” for more information.

Chapter 8

Managing RealAudio Services

More and more people wish to listen to audio files found at sites on the Internet. As with other protocols, access to these files is not without risk, so they require logging and access control, as with other services.

The RealAudio protocol allows people to play and listen to audio material. The Gauntlet Firewall includes a RealAudio proxy that securely handles requests to listen to audio data.

This chapter explains the concepts behind the RealAudio proxy, how it works, and how to configure and use it.

Understanding the RealAudio Proxy

The Gauntlet RealAudio proxy is an application level proxy that provides configurable access control. The proxy, which runs on the firewall, passes RealAudio client requests through the firewall, using rules you supply. You can configure the RealAudio proxy to allow connections based on:

- source host name
- source IP address
- destination host name
- destination IP address

Using these options, you can configure the firewall to allow RealAudio clients on the inside network to access RealAudio servers on the outside network. You can also limit the RealAudio sites your users can access from machines on the inside network. The RealAudio proxy also logs all successful and unsuccessful connection attempts, as well as the amount of data transferred.

Note: You cannot configure the RealAudio proxy to allow access to RealAudio servers that you have placed on the inside network.

Used together, these access controls and log files give you much more control over the RealAudio connections to and from your system than you would have without the firewall.

How It Works

The firewall runs the RealAudio proxy (*rap-gw*) as a daemon listening for requests on the default RealAudio port (7070). When the firewall receives requests for services on this port, the RealAudio proxy checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use RealAudio. If the host has permission, the proxy logs the transaction and passes the request to the outside host. The *rap-gw* daemon remains active until either side closes the connection. This proxy running on the RealAudio port 7070 only works if you have enabled transparent proxies. It does not require RealAudio players which can be configured to explicitly use a proxy.

The firewall also runs the proxy on the default RealAudio proxy port (1080). The proxy works as described above. However, you must configure your RealAudio player to use the RealAudio proxy that is running on port 1080. Only recent RealAudio players can be configured explicitly to use the RealAudio proxy on port 1080. The transparent proxy feature does not need to be enabled in this case.

The default policy allows inside hosts to use RealAudio. Users on inside hosts can continue to access RealAudio servers on outside hosts to download audio files and listen to live broadcasts.

The default policy does not allow outside hosts to connect to RealAudio servers running on the inside network.

This configuration prohibits running a RealAudio server on the firewall itself. Because the RealAudio proxy is running on the RealAudio default player port on the firewall, all RealAudio requests access the proxy. There is no way to start the RealAudio server daemon needed to service RealAudio requests.

Configuring the Firewall to Use the RealAudio Proxy

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, and configuring the RealAudio proxy to enforce your policy.

Planning

Determine which internal users and hosts can use these services, and determine whether you want to run the RealAudio proxy transparently on port 7070, or on port 1080.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support the RealAudio server. This is a standard service, included in the default versions of these configuration files on the Gauntlet firewall.

Configuring the Proxy Rules

If you are using the Gauntlet firewall default configuration, you do not need to modify the proxy rules for the RealAudio server. To enable the RealAudio server, use the *gauntlet-admin* Proxies form to enable the server.

Alternatively, you may modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on *rap-gw* options, *netperm-table* options, and order of precedence.

Verifying Your Setup

Verify your installation by using your RealAudio player to listen to audio files or live broadcasts from hosts on the outside network. See the section below for instructions.

Using the RealAudio Proxy

Most users and most sites do not need to change the way they access RealAudio files after installing the RealAudio proxy.

If you are using transparency on your firewall and you have installed the RealAudio proxy on the RealAudio default player port (7070), you do not need to change the way you access RealAudio files.

If you are not using transparency or you have installed the firewall on the RealAudio default proxy port (1080), you need to configure your RealAudio player to know about the proxy and the other port.

To configure the RealAudio player:

1. Select View.
2. Select Preferences.
3. Select Proxy.
4. Check the Use Proxy box.
5. Enter as the host the name for the inside interface of your firewall.

Now, when you point your web browser or RealAudio player at a RealAudio file, they use the proxy.

Chapter 9

Managing MediaBase Services

MediaBase is a collection of multimedia and hypertext that allows users to select and play videos using their Web browser. The Gauntlet Firewall includes a MediaBase proxy that securely handles outside user requests to view video data on a MediaBase server inside the firewall. This proxy also allows users inside the firewall to access MediaBase servers on outside networks.

This chapter explains the concepts behind the MediaBase proxy and how it works.

Note: For additional information on setting up the Gauntlet firewall for MediaBase, see "Configuring a MediaBase Proxy for the Gauntlet Internet Firewall" in the *WebFORCE MediaBase Administrator's Guide*.

Understanding the MediaBase Proxy

The Gauntlet MediaBase proxy is an application level proxy that provides configurable access control. The proxy, which runs on the firewall, passes MediaBase client and server requests through the firewall, using rules that you supply. You can configure the MediaBase proxy to allow connections based on:

- source host name
- source IP address
- destination host name
- destination IP address

Using these options, you can configure the firewall to allow MediaBase clients on the inside network to access MediaBase servers on the outside network. You can also limit the MediaBase sites your users can access from machines on the inside network.

By default, verbose logging is not enabled on the MediaBase proxy. If you enable verbose logging, information on all connections is logged.

Used together, these access controls and log files give you much more control over the MediaBase connections to and from your system than you would have without the firewall.

How It Works

The firewall runs the MediaBase proxy (*mbase-gw*) as a daemon listening for requests on a series of ports: ports 6301, 6309, 6310, 6312, and 6313 handle control information; ports 6320 through 6323 and 6340 handle data information. When the firewall receives requests for those ports, the MediaBase proxy checks its configuration information (in the *netperm-table* file) and determines whether the initiating client has permission to use MediaBase. If the client has permission, the proxy logs the transaction and passes the request to the appropriate host.

The *mbase-gw* daemon is always active. This daemon requires that MediaBase players also be configured to use a proxy.

The default policy allows clients inside the network to connect to MediaBase servers; it does not allow outside clients such access, however. Because the firewall runs the MediaBase proxy on all MediaBase ports, all requests from outside clients access the MediaBase proxy rather than the server. This configuration prohibits running a MediaBase server on the firewall itself—there is no way to start a MediaBase server to accept such requests.

Configuring the Firewall to Use the MediaBase Proxy

Configuring the Gauntlet firewall involves planning, indicating which servers may be accessed, and configuring the MediaBase proxy to enforce your policy.

Planning

Determine which internal users and hosts can use MediaBase, and determine whether you want to run the MediaBase proxy.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support the MediaBase server. This is a standard service, included in the default versions of these configuration files on the Gauntlet firewall.

Note: On IRIX 6.2 systems, system patches 1485 and 1418 are required.

Configuring the Proxy Rules

If you are using the Gauntlet firewall default configuration, you do not need to modify the proxy rules for the MediaBase server. To enable the MediaBase server, use the *gauntlet-admin* Proxies form to enable the server.

Alternatively, you may modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on *mbase-gw* options, *netperm-table* options, and order of precedence.

Verifying Your Setup

Verify your installation by using your MediaBase player to connect to MediaBase servers on the outside network. See the section below for instructions.

Using the MediaBase Proxy

Users must set up the client-side configuration files to enable the MediaBase client to communicate with a MediaBase firewall proxy.

Chapter 10

Managing X Window Services

The X Window System provides many features and functions that allow machines to share input and output devices. A user running the X Window System on one machine can display the results of a graphical program on another machine running an X Window client. This flexibility is also the source of a number of well-known security problems. When you allow access to your display, you are essentially allowing access to your screen, mouse and keyboard. Most sites do not want to provide this sort of free access to their machines, but administrators recognize that these services can be useful. The X11 proxy included with the Gauntlet Firewall allows administrators to selectively allow X11 services through their firewall.

This chapter explains the concepts behind the X11 proxy and how it works, how to configure the proxy, and how to use X11 services through the firewall.

Understanding the X11 Proxy

The Gauntlet X11 proxy is an application-level proxy that provides configurable access control. The proxy, which runs on the firewall, passes X11 display requests through the firewall, using rules you supply. You can configure the proxy to allow display requests based on

- display name
- user

Using these rules, you can configure your firewall to allow only certain machines on the inside network to display information from machines on an outside network. An employee working on the inside network can configure his or her machine to display information from a program on a client's machine on the outside network. Similarly, you can configure your firewall to permit only certain users to use the X11 proxy.

The X11 proxy also requires the user to confirm each new request for a connection to their display. Because of the lack of strong authentication systems for X11, this reconfirmation provides an additional opportunity to confirm that you really want to accept the connection. You can watch for other people trying to hijack your display.

Because the X11 proxy works in conjunction with the TELNET and Rlogin proxies, you can still configure access based on the source or destination hostname or IP address. The strong authentication feature is also available. The TELNET and Rlogin proxies also log X requests and connections.

How the X11 Proxy Works

Unlike some of the other Gauntlet proxies, the firewall does not start the X11 proxy when it receives display requests. Instead, users must explicitly start the X11 proxy from either the TELNET or Rlogin proxy. The firewall denies all requests for services on the standard X port (6000).

A user TELNETs to the firewall, which runs the TELNET proxy. After checking permissions and authenticating users (as described in chapter 1), the TELNET proxy (*tn-gw*) displays a prompt for the user. At the prompt, the user indicates a wish to allow X displays across the firewall. The TELNET proxy starts the X11 proxy (*x-gw*) on port 6010 (corresponding to X display ":10") or higher. The X11 proxy checks its configuration information (in the *netperm-table* file) and determines whether the initiating user has permission to use X11 services related to the desired display.

If the user has permission, the proxy creates a "virtual display" on the firewall for the requesting client. When the outside X client requests access to the user's display, the virtual display server passes a query display to the X server on the display machine. This X server displays the query window on the real display, prompting the user to confirm the request. After the user confirms the request, the real X server receives the display information from the virtual X server. The proxy remains active until either end closes the connection.

The default policy is to allow both inside and outside hosts to start the X11 proxy.

Configuring the Firewall for X11 Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, and configuring the proxies to enforce your policy.

Planning

1. Determine whether you wish to allow X11 display connections through the firewall.
2. Determine which users and which displays can issue and receive display requests.
3. Ensure that your policies for X11 services and TELNET and Rlogin are compatible.

Configuring Network Services

You do not need to modify your network files on the firewall to use the X11 proxy. The TELNET and Rlogin proxies are the only programs that can start the X proxy, and they read their configuration information from the *netperm-table* file.

Configuring the Proxy Rules

To enable the X11 proxy for TELNET and Rlogin users, use the *gauntlet-admin* Proxies form.

Alternatively, you may modify */usr/gauntlet/config/template.netperm-table* to configure the X11 proxy to enforce more specific security policies. See Appendix B for more information on *x-gw* options, *netperm-table* options, and order of precedence.

Verifying Your Setup

TELNET to a machine outside the perimeter and display an X11 client on your machine inside the perimeter. See the section below for instructions.

Using X11 Services

Users need to follow slightly different procedures to use X11 services through a firewall. The minimal time needed for these additional steps outweighs the time and money you would spend to recover after someone hijacks your display and thereby penetrates security.

To use X11 services, follow these steps:

1. Allow the firewall to access your display (remember, it is the firewall you permit to access your display, not the client).
2. TELNET (or Rlogin) to the firewall.
3. Authenticate to the proxy, if necessary.
4. Start the X proxy.
5. TELNET (or Rlogin) to the desired host.
6. Inform the client of the host and display information that the proxy provides.
7. Start the X client application.
8. Confirm the display request on the real display.

The example below shows a user working on the inside network who needs to display information from a program running on a machine on an outside network.

Clancy Rawhide, working at his machine (dimension) on the inside network, needs to run an X program on a client machine (blaze.clientsite.com) on an outside network, and display the results on his display. He first gives the firewall access to his system's display. He then TELNETs to the firewall for Yoyodyne (firewall.yoyodyne.com). The policy for his site does not require authentication for inside requests, so the firewall connects him to the TELNET proxy.

First, Clancy starts the X11 proxy and establishes a TELNET connection with the outside host:

```
dimension-27: xhost +firewall
dimension-28: telnet firewall
Trying 204.255.154.100...
Connected to firewall.yoyodyne.com
Escape character is '^]'.
firewall.yoyodyne.com telnet proxy (Version 3.1) ready:
tn-gw> x
```

Clancy indicates he wants to start an X proxy. The firewall displays an X status window on Clancy's display, showing the port (see Figure 10-1).

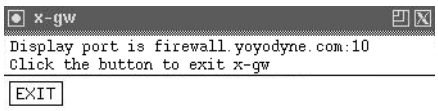

Figure 10-1 Example X Window Port Information

He then TELNETs to the client machine (blaze.clientsite.com).

```
tn-gw> c blaze.clientsite.com
Connecting to blaze.clientsite.com .... connected
HP-UX blaze A.09.01 E 9000/710 (ttys1)
```

The TELNET daemon on blaze prompts Clancy for his user name (crawhide) and password on blaze. The TELNET daemon on blaze verifies Clancy's user name and password, and logs him in.

```
login: crawhide
Password: #########

Please wait...checking for disk quotas
You have mail.
blaze.clientsite.com-1:
```

Next, Clancy provides the X display information to the client machine (blaze) and starts the client application. He uses the display information that the X proxy provided when he started the X proxy:

```
blaze.clientsite_1: setenv DISPLAY firewall.yoyodyne.com:10.0
blaze.clientsite_2: xclock &
blaze.clientsite_3:
```

Clancy uses the information the proxy provided to tell X where to display information. Clancy then starts the program, and confirms the display request on his machine (see Figure 10-2).

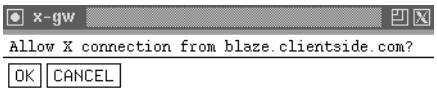

Figure 10-2 Example X Window Confirmation

Finally, Clancy views the results on his screen inside the firewall.

Chapter 11

Managing LP Services

Printing continues to be a widely used feature of most computer networks. In some circumstances, users need to print information using printers connected to other machines on other networks. Users behind a firewall might want to print to printers on systems on the outside, or behind other firewalls. Others might want to be able to print from a remote system, for example a mobile PC, to a printer behind a firewall. The Gauntlet Firewall includes an *lp* proxy that securely handles the transfer of print requests.

This chapter explains the concepts behind the *lp* proxy and how it works, how to configure the proxy, and how to use lp services.

Understanding the lp Proxy

The Gauntlet *lp* proxy is an application-level gateway that provides configurable access control and logging mechanisms. The *lp* proxy, which runs on the firewall, passes *lp* requests through the firewall, using rules you supply. You can configure the *lp* proxy to allow file transfer activity based on

- source IP address
- source hostname
- destination IP address
- destination hostname
- *lp* commands (for example, number and priority)
- printer queue

Using these options, you can configure your firewall, for example, to allow specific hosts on the inside network to print files on outside hosts. Employees working behind the firewall can then send print jobs to printers at customer sites. Similarly, traveling employees can send print jobs to printers at corporate headquarters inside the defense

perimenter. Or, you could deny access to *lp* commands, allowing users to print, but not allowing them to restart or remove print jobs.

The *lp* proxy logs all successful and unsuccessful file transfer attempts, and the number of bytes transferred. Used together, these access controls and log files allow you to have much more control over the files entering and leaving your system than you have when you use the standard IRIX *lp* program.

How the lp Proxy Works

The IRIX system runs the lp proxy (*lp-gw*) as a daemon listening for requests on the standard printer port (515). When the firewall receives requests for services on this port, the *lp* proxy checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use *lp*. If the host has permission, the proxy logs the transaction and passes the request to the outside host. The *lp-gw* remains active until either side closes the connection or the proxy times-out the connection.

The default policy allows inside hosts to use *lp*. Users on inside hosts can continue to print to outside hosts as they did before the firewall was put into place. The default policy does not allow outside hosts to connect to inside hosts for printing.

The default policy and this configuration prohibit running an *lp* server on the firewall itself. Because the *lp* proxy is running on the standard *lp* port on the firewall, all *lp* requests start the proxy. There is no way to start the *lp* daemon needed to service *lp* requests. The default policy does not allow any hosts to print to the firewall.

Configuring the Firewall for lp Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, and configuring the *lp* proxy to enforce your policy.

Planning

1. Determine which internal users and hosts can use these services.
2. Determine which external users and hosts can use these services.

Configuring Network Services

To configure network services with the *gauntlet-admin*, enable *lp* in the Proxies form, and modify the idle timeout if desired.

You can use the gauntlet-admin Proxies form to create virtual queues on the firewall, which will be translated to the real servers and queues you specify. This is implemented using the *printer* directive in the *netperm-table* file.

Configuring the Proxy Rules

You may further configure the *lp* proxy to enforce your security policies. This involves modifying */usr/gauntlet/config/template.netperm-table*. See Appendix B for more information on *lp-gw* options, *netperm-table* options, and order of precedence.

To configure the *netperm-table* file, follow these steps:

1. Add the *lp* proxy to your inside and outside policies, as appropriate.
2. Create an *lp* proxy section, specifying the inside hosts, outside server and printer queue:

   ```
   lp-gw: printer * -host blaze.clientsite.com -printer lp-main
   ```

3. Configure other *lp* proxy options, as appropriate for your setup. These could include logging or denying specific commands.
4. Comment your additions.

Configuring the Sending Machine

Configure the print queue information on the sending machine. Define the print queue so that the firewall is the print queue destination.

Configuring the Receiving Machine

Configure the print queue information on the sending machine. Define the print queue to accept requests from the firewall.

Verifying Your Setup

Verify your configuration by printing a file from a host inside your firewall to a host outside your firewall. If you are configured to do so, print a file from a host outside your firewall to a host on the inside of your firewall.

Using lp Services

The firewall and the *lp-gw* proxy are transparent to the user. Users can continue to use *lp* to permitted servers and printers as they did before.

Chapter 12

Managing Sybase Services

Database services are essential in most organizations. As with other services you offer, you want to securely configure database access. Sybase is a relational database management system in use in many organizations. The Gauntlet firewall includes a proxy that securely allows connections between Sybase clients on the inside network and servers on the outside network.

This chapter explains the concepts behind the Sybase proxy and how it works, how to configure the proxy, and how to use Sybase services.

Understanding the Sybase Proxy

The Gauntlet Sybase proxy is an application level proxy that provides configurable access control, authentication and logging mechanisms. The Sybase proxy, which runs on the firewall, passes Sybase requests through the firewall (at the application level), using rules you supply. You can configure instances of the Sybase proxy to service:

- Sybase client to server communications
- Sybase server to server communications

For each version of the Sybase proxy, you can configure the proxy to allow connections based on:

- source IP address
- source host name
- source port
- destination IP address
- destination host name
- destination port

Using these options, you can configure your firewall to allow Sybase clients on certain trusted hosts to access a Sybase server on an untrusted host. Employees working behind the firewall can access Sybase databases at customer sites. You can also configure your firewall to allow Sybase servers on opposite sides of the firewall to communicate. A Sybase replication server can communicate with another Sybase replication server on the other side of an Intranet firewall.

You can configure the Sybase proxy to allow Sybase clients on untrusted hosts to access Sybase servers on your trusted networks. According to most security policies, including the Gauntlet firewall default, it's not a good idea. If you must allow this sort of service, consider using client-side password encryption. Consider limiting the databases and data to which users have access as all of the data is transferred unencrypted.

The proxies log all successful and unsuccessful connection attempts, and the amount of data transferred.

These access controls allow you to have much more control over the connections to and from your system than without a firewall. The logging capabilities are also much more extensive.

How It Works

The firewall runs different instances of the Sybase proxy (*syb-gw*) as daemons on different ports for different Sybase applications, based on the information in the */etc/services* and */usr/gauntlet/bin/gauntlet* files. These files indicate which Sybase services the firewall should run on which ports.

Whenever the firewall receives a Sybase request on one of these ports, the Sybase proxy checks its configuration information (in the *netperm-table*) and determines whether the initiating host has permission to initiate this type of request. If the host does not have permission, the Sybase daemon logs the connection attempt and displays an error message.

If the host has permission, the proxy logs the transaction and passes the request to the destination host. The Sybase proxy remains active until either side closes the connection.

The default policy does not allow either inside or outside hosts to use the Sybase proxy. The recommended configuration allows trusted hosts to access Sybase servers on untrusted networks. The recommended configuration does not allow untrusted hosts to

access Sybase servers on trusted networks. While the Sybase proxy does perform checks to ensure that the packets appear to be Sybase packets, someone could spoof this protocol. The Sybase proxy does not perform any user authentication. You are relying on the authentication mechanisms of the Sybase server to control access to your Sybase server and its data.

Configuring the Firewall for Sybase Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the proxies to enforce your policy, and configuring Sybase clients.

Planning

1. Determine which Sybase servers users need to access. Determine whether you want to limit access to particular a server or not. Obtain host name or IP address information for each server.
2. For each server, determine the port(s) on which the server accepts connections.
3. Determine which external hosts can use these services.
4. Determine which internal hosts can use these services.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support NNTP. This is a standard service, included in the default versions of these files on the Gauntlet Firewall.

Configuring the Proxy Rules

In most cases, you do not need to modify the proxy rules for NNTP. This is a standard service.

Configuring Sybase Clients

Add or modify the interfaces file on the client (using a tool like *sybinit* or *SQLEdit*) to provide information about the Sybase server:

1. Specify the port number you selected for the Sybase proxy.
2. If you are using transparency (the default configuration), specify the host name as the host name of the actual machine running the Sybase server. If you are not using transparency, specify the host name as the IP address of the firewall.

If you are using server-to-server communications, configure your servers as clients. Consult your Sybase administration documentation for further information on configuring clients for accessing servers.

Verifying Your Setup

Use your Sybase client on a trusted host to run a simple query against the Sybase server on the untrusted host. Watch the logs on the firewall for error messages.

PART THREE

Administering General Gauntlet Firewall Services

Managing NNTP and General TCP Services

Usenet news continues to be one of the most widely used features on the Internet. Many sites rely on Usenet news for information on the latest technology. Although the Network News Transfer Protocol (NNTP) does little in comparison to other network protocols, you must configure it carefully to protect internal news groups that may contain sensitive proprietary information.

The plug proxy included with the Gauntlet firewall allows administrators to tunnel NNTP-based news feeds through their firewall. The NNTP connections come from known sites (as opposed to the multitude of sites that may connect via SMTP to deliver mail). NNTP is also a very straightforward protocol. For these reasons, it can be proxied using the generic plug proxy.

Other common programs, such as *whois* and *webster*, run over TCP. You can also tunnel these services through the firewall with the plug proxy.

Many sites also rely on applications such as America Online, CompuServe and Lotus Notes. Each of these services uses a proprietary protocol, which could require a multitude of application-specific proxies. Instead, administrators can use the plug proxy to tunnel these through the firewall.

Warning: The consequences of allowing proprietary protocols through your firewall are not well known. Because the protocols are proprietary, the firewall and the proxy have no idea what sorts of data or requests the applications are sending. Nor can there be any idea how safe the actual application is. Do not use the plug proxy for proprietary protocols without first performing a risk assessment.

The plug proxy does not support UDP-based services. UDP is not a connection oriented protocol. Because there is no connection, there are no sequence numbers. This makes it much easier for someone to create a UDP packet that appears valid but contains fabricated source and destination information.

This chapter explains the concepts behind the *plug* proxy and how it works, how to configure the proxy for NNTP news and other services, and how to configure these services to run through the firewall.

Understanding the Proxy

The Gauntlet plug proxy is a TCP gateway that provides configurable access control and logging mechanisms. The plug proxy, which runs on the firewall, passes NNTP or other application requests through the firewall, using rules you supply. It essentially tunnels information from a port on the firewall to a specific port on another machine.

You can configure instances of the plug proxy to service

- NNTP news feeds
- webster
- whois

This is not an exhaustive list. The plug proxy is protocol neutral, so you can tunnel a variety of other applications. Weigh the risks carefully for each application.

For each version of the plug proxy, you can configure the proxy to allow connections based on

- source IP address
- source hostname
- source port
- destination IP address
- destination hostname
- destination port

Using these options for the plug proxy, you could configure your firewall to allow your service provider's host on the outside to connect to the firewall and pass news via NNTP to your news machine on the inside network. You could also route all internal requests for whois lookups to a specific whois server on the outside network.

The proxies log all successful and unsuccessful connection attempts, and the amount of data transferred.

Used together, these access controls and log files allow you to have much more control over the connections to and from your system than without a firewall. However, you may be allowing proprietary protocols into your network, which can be dangerous.

How It Works

The firewall runs different instances of the plug proxy (*plug-gw*) as daemons on different ports for different applications. These files indicate which services the firewall should run on which ports. For example, the firewall runs an instance of the plug proxy on port 119 to handle NNTP requests if you have enabled NNTP from *gauntlet-admin*.

When the plug proxy receives a request on its port, it checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to initiate this type of request. If the host has permission, the plug proxy passes the connection to the specified port on the specified machine. This instance of the plug proxy remains active until either side terminates the connection.

The Gauntlet firewall includes configuration information for NNTP transfer. The default policy is to allow requests to and from one internal news server and one external news server. The firewall itself cannot run an NNTP news server, because the plug proxy is using the standard port for these services.

Hosts on both the inside and outside think the firewall is servicing requests. The external news server thinks it is feeding news to the firewall, and the internal news server thinks that it is receiving news from the firewall. The firewall is simply acting as the tunnel, via the plug proxy.

Configuring the Firewall for NNTP

Configuring the Gauntlet firewall involves planning, configuring the firewall, indicating which daemons the system will run, configuring the proxies to enforce your policy, informing your news feed, and configuring your internal news server.

Note: If you receive news feeds from multiple external sources, see the section on “Configuring Multiple Newsfeeds” on page 91. If you are not receiving a news feed and want to configure the NNTP proxy for reading news, see the subsection on Configuring Your NNTP Proxy for Reading News on page 90.

Planning

1. Do not use the firewall as a news server.
2. Allow external NNTP connections from known servers only.

Configuring the Firewall

If you wish to allow NNTP traffic through the firewall, configure the firewall using the *gauntlet-admin* interface's Proxies form.

To configure the firewall, follow these steps:

1. Enter the IP address for your internal NNTP news server. Use the IP address rather than the hostname.
2. Enter the IP address for your external NNTP news server. Use the IP address rather than the hostname.
3. Enable the NNTP checkbox.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support NNTP. This is a standard service, included in the default versions of these files on the Gauntlet Firewall.

Configuring the Proxy Rules

In most cases you do not need to modify the proxy rules for NNTP. This is a standard service.

Informing Your News Feed

Inform your external news feed (often your Internet service provider) that it should now send all NNTP news to your firewall, rather than your internal news server.

Configuring Your News Server

Configure your internal news server software to transfer and receive articles from the firewall, rather than your external news server.

Verifying Your Setup

Run your news server as you did before. Watch the logs for errors.

Using NNTP

The firewall and the plug proxy for NNTP traffic are transparent to the user. Users should continue to point their news readers (*rn*, *trn*) and other news-aware tools (Netscape) towards the internal news server. It's that easy.

Configuring the Firewall for Other Protocols

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the proxies to enforce your policy, and configuring your service.

Note: If you have simple plug gateway needs, you can add custom plug gateways using the *gauntlet-admin* proxies page. If you use that method, you may still need to modify */etc/services*, but do *not* need to modify */etc/init.d/network.local* or */usr/gauntlet/config/template.netperm-table*.

This section uses the Quote of the Day (*qotd*) service as an example. Of course, you must carefully determine if the benefits of something like a Quote of the Day service outweigh the risks of allowing that type of service within your security perimeter.

Planning

1. Determine which protocols and which applications you wish to proxy through your firewall.

Warning: Again, the consequences of allowing proprietary protocols through your firewall are not known. Because the protocols are proprietary, the firewall and the proxy have no idea what sorts of data or requests the applications are sending. Nor can it be determined how safe the actual application is. Do not use the plug proxy for proprietary protocols without first performing a risk assessment.

2. Verify that the protocol is stream based.

3. Determine what port these services use. Verify that the service uses the same port for sending and receiving.
4. Determine which external hosts can use these services.
5. Determine which internal hosts can use these services.

Configuring Network Services

Configuring network services involves modifying IRIX configuration files so the system knows which programs to start.

To configure network services, follow these steps:

1. Add information to */etc/services* so that the system knows what service it is offering on that port:

   ```
   qotd          17/tcp  quote          # Quote of the Day
   ```

   The protocol you indicate is the protocol that the plug proxy uses (TCP).

2. Add information about the plug proxy to */etc/init.d/network.local* so that the system knows what daemon to start to handle Quote of the Day requests:

   ```
   echo "    qotd"
   /usr/etc/plug-gw -as qotd-gw -daemon qotd qotd &
   ```

   See the comments in */etc/init.d/network* on how to ensure that */etc/init.d/network.local* will be executed at boot time.

   Use the same name for the service that you specified in */etc/services*.

Configuring the Proxy Rules

Configure the plug proxy to enforce your security policies. This involves modifying */usr/gauntlet/config/template.netperm-table*. You may use the *gauntlet-admin* Proxies form. In the section called “Plug Gateways,” enter the source host, the firewall port, the destination host, and the destination port for each plug gateway.

Alternatively, you may modify system files directly. Appendix B provides more information on *plug-gw* options, *netperm-table* options, and order of precedence.

To configure the *netperm-table* file directly, follow these steps:

1. Create a plug proxy section for your service, specifying the inside host(s) that can use this service and destination servers and ports:

   ```
   qotd-gw: port qotd 10.0.1.* -desthost qotd.bigu.edu -destport qotd
   ```

   which indicates that any inside hosts can send Quote of the Day requests to the server at Big University.

2. Specify the outside hosts that can use this service and the inside servers and ports. Because you are not running a Quote of the Day server inside the perimeter, you do not need to add this line for our *qotd* example.

3. Comment your additions.

Configuring Your Service

You may need to configure your service and application to connect to the firewall instead of directly to the server. Consult the documentation included with your plugged service for information on possible configurations.

Verifying Your Setup

Try accessing the service in the way it is meant to be used. Conversely, access the service in inappropriate ways. Watch the logs on the firewall for error messages.

Configuring Multiple Newsfeeds

If you wish to exchange news with multiple news servers outside the perimeter, you must add additional configuration rules for the plug proxy that handles NNTP transfers. This is done without the GUI.

To configure additional newsfeeds, follow these steps:

1. Create plug proxy lines that handle multiple internal servers, multiple external servers, or both.

2. Use the same plug proxy line that you use for other news information.

3. Add permit lines to the inside and outside policies for your new plug proxies.

For example, you have configured news.myisp.net as your primary news feed through the gauntlet-admin interface. To add support for a secondary news feed from news.bigu.edu (192.168.1.202) to your internal news machine news.yoyodyne.com (10.0.1.3), use the following lines in your *netperm-table* file:

```
#adds support for additional feed from Big University
nntp-gw: port nntp 10.0.1.3 desthost 192.168.1.202 destport nntp
nntp-gw: port nntp 192.168.1.202 desthost 10.0.1.3 destport nntp
```

Configuring Your NNTP Proxy for Reading News

You may not wish to carry a full news feed for your organization. However, you still want to allow your users to read news. For example, you need to allow users to directly access news servers on untrusted networks.

To configure for reading news from servers on untrusted networks:

1. Use the gauntlet administration tools to disable NNTP configuration for your firewall. This configuration handles a single internal NNTP server connecting to a single external NNTP server. Set both the internal and external NNTP servers to none.
2. Modify the netperm-table to ensure that the inside policy permits the NNTP proxy:

   ```
   policy-trusted:permit-proxy nntp-gw
   ```
3. Add a section to the NNTP section of your netperm-table that indicates the hosts that are allowed to connect to the NNTP port:

   ```
   #allow any inside host to connect to the nntp proxy
   nntp-gw: port nntp 10.0.1.* -port nntp
   ```
4. Configure internal news readers to read and post news to the appropriate news server on the untrusted network.

Chapter 14

Managing General TCP Services With Authentication

Many sites rely on group ware or other interconnected applications, such as accounting packages and database applications. Each of these services uses a proprietary protocol, which would require its own application-specific proxy. The plug proxy might be an ideal candidate for many of these applications. However, administrators also want to control who can access the service (by username), which the plug proxy cannot do. Instead, administrators can use the circuit proxy to allow certain users to tunnel these proprietary applications through the firewall.

Caution: Allowing proprietary protocols through your firewall is a really big unknown. Because the protocols are proprietary, the firewall and the proxy have no idea what sorts of data or requests the applications are sending. Nor do we have any idea how safe the actual application is. Do not use the circuit proxy for proprietary protocols without first performing a risk assessment.

This chapter explains the concepts behind the circuit proxy, how it works, and how to configure and use the circuit proxy.

Understanding the Circuit Proxy

The Gauntlet circuit proxy is an authenticated TCP gateway that provides configurable access control and logging mechanisms. The proxy, which runs on the firewall, authenticates users and passes TCP-based application requests through the firewall, using rules you supply. It essentially tunnels information from a port on the firewall to a specific port on another machine, after authenticating the user.

You can configure the circuit proxy to service:

- database applications
- financial applications
- groupware

This is not an exhaustive list. The circuit proxy is protocol neutral, so you can tunnel a variety of other stream-based applications. Weigh the risks carefully for each application.

You can configure the circuit proxy to allow connections based on:

- username
- source host name
- source IP address
- source port
- destination host name
- destination IP address
- destination port

Using these options, you can configure your firewall to allow certain users to use a database server on a machine outside the defense perimeter. Employees working outside the perimeter can access important services inside the perimeter.

The strong authentication features of the circuit proxy require users to authenticate before connecting, if required. The circuit proxy also logs all successful and unsuccessful connection attempts, and the amount of data transferred.

These access controls allow you to have much more control over the connections to and from your system than without a firewall The logging capabilities are also much more extensive.

How It Works

The firewall runs the circuit proxy (*ck-gw*) as a daemon on a user specified port (generally on a port above 1024). The user initiates the connection by TELNETing to the port where the circuit proxy is listening (which is a different port than the port on which the service runs). When the proxy receives a request on this port, it checks its configuration information (in the *netperm-table*) and determines whether the initiating host has permission to initiate this type of connection. If the host has permission, the circuit proxy authenticates the user with the authentication server specified in the configuration information.

If the authentication is successful, the proxy uses its configuration information to create a menu listing the available services for this user. The user selects from the menu the service they want to start. The proxy then waits at the port specified for this service for a connection from the user's machine.

The user then starts the client application, which connects to the firewall on the service's port. The proxy accepts the connection and displays a confirmation request in the user's original TELNET window. After the user confirms the request for the connection, the circuit proxy starts a child process to handle the service request. The child process creates a connection from the client to the application server on the other side of the defense perimeter. The proxy then passes requests back and forth between the application client and server. The child process of the circuit proxy remains active until either side terminates the connection. The original TELNET window also remains active until either side terminates the connection.

Configuring the Firewall for Authenticated TCP Services

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the proxies to enforce your policy, starting your proxy, rebooting your firewall, and configuring your service.

Planning

Caution: Allowing proprietary protocols through your firewall is a really big unknown. Because the protocols are proprietary, the firewall and the proxy have no idea what sorts of data or requests the applications are sending. Nor do we have any idea how safe the actual application is. Do not use the circuit proxy for proprietary protocols without first performing a risk assessment.

1. Verify that the protocol uses TCP.
2. Verify that the protocol is stream-based by consulting your protocol documentation and trying it with the plug proxy.
3. Verify that the protocol uses one port for all server connections.
4. Verify that the protocol opens only one connection for communicating between the client application and server.

5. Determine which external hosts can use these services.
6. Determine which internal hosts can use these services.

Configuring Network Services

Configuring network services involves modifying UNIX configuration files so the firewall knows which programs to start.

To configure network services:

1. Add information to */etc/services* so that the firewall knows about the circuit proxy, indicating the port it uses for TELNET requests, and that it use TCP:

   ```
   ck-gw          2000/tcp          #Circuit Proxy TELNET port
   ```

   - Be sure to use a port above 1025.

2. Add information to */etc/services* so that the firewall knows about the service you are porxying, indicating name of the service, the port and that it uses TCP:

   ```
   oracle         1176/tcp          #Oracle
   ```

3. Modify the default circuit proxy startup script in */usr/local/etc/mgmt/rc* so that the system knows what daemon to start:

   - Rename the default circuit proxy script (*D270ck-gw*) to a file name that starts with S and a number. The firewall starts the daemons in numeric order, so name your file accordingly.

     ```
     fire-in# rename D270ck-gw S270ck-gw
     ```

   - Edit the new script and modify the following items to add information about the circuit proxy:

     PORT — name of the service or port on which the proxy runs. Use the same service name or port number you specified in */etc/services*.

     ```
     PORT=oracle
     ```

     VARIABLE — name of the variable used to identify the service in the Gauntlet administration menus to turn the proxy on and off.

     ```
     VARIABLE=proxy_custom2
     ```

ARGS — command line arguments for the proxy, including the *-as* option so it runs under a different name:

```
ARGS="-as qotd-gw"
```

Do **NOT** use the -daemon option with the circuit proxy, as you would with other Gauntlet proxies. The circuit proxy automatically starts as a daemon.

- Note that you do NOT need to add information about the service to the startup script. The circuit proxy starts the process on the appropriate port only after the user confirms the request for service.

Configuring the Proxy Rules

Configure the circuit proxy to enforce your security policies. This involves modifying */usr/local/etc/netperm-table*. See Appendix for more information on *ck-gw* options, *netperm-table* options and order of precedence.

To configure the netperm-table:

1. Add the circuit proxy as a permitted proxy for your policies as appropriate:

   ```
   #allow trusted hosts to use the circuit proxy
   policy-trusted: permit-proxy ck-gw
   ```

2. Create a circuit proxy section for your service, indicating the services offered and the ports used:

   `ck-gw: server` *service* `-port` *remote-port* `-host` *remote-host*

   - where:
     - server *service* indicates the name of the available service. Used by the proxy to create the menu of available services.
     - port *remote-port* indicates the port on the remote host to which the circuit proxy connects. Specify by service name or port number.
     - host *remote-host* indicates the name of the remote host to which the circuit proxy connects. Specify an individual machine. Use IP addresses or host names. This option is required if you are not using transparency.
   - For example:

     ```
     #create a circuit for oracle to clientsite
     ck-gw: server oracle -port oracle -host db.clientsite.com
     ```

3. Indicate the authentication server the circuit proxy uses:

   `ck-gw: authtype` *hosts* `[-authhost` *host*`] [-authport` *port*`]`

   - where:
     - authtype *hosts* indicates the hosts for which the circuit proxy authenticates. Specify individual machines, entire networks, or subnets. Use IP addresses or host names. The * wildcard is valid.
     - authhost *host* indicates the host running the authentication server. Specify by IP address or host name.
     - authport *port* indicates the port on the host that the circuit proxy uses for communicating with the authentication server.
   - For example:

     ```
     #use the auth server on the firewall
     ck-gw: authtype * -authhost 127.0.0.1 -authport 7777
     ```

   - You can use the *authserver* attribute instead of *authtype*. If you specify an *authtype* attribute, the circuit proxy uses the *authtype* attribute instead of the *authserver* attribute.
4. Comment your additions.

Verifying Your Setup

Verify your installation by using your application through the circuit proxy. See the section below for instructions. Watch the logs on the firewall for error messages.

Using the Circuit Proxy

Users need to follow slightly different procedures to use their application through the circuit proxy.

1. TELNET to the circuit proxy.
2. Authenticate to the circuit proxy, if required.
3. Select the desired service.
4. Start your client application.

5. Confirm the client application connection.
6. Use your application.

The example below shows a user working on the trusted network inside the defense perimeter. The company has a policy to authenticate the use of some outside services. He is accessing an Oracle database on a machine outside the perimeter at a client site.

First, the user TELNETs to the port on the firewall where the circuit proxy is running. He authenticates using S/Key password:

```
dimension-59: telnet fire-in ck-gw
Trying 10.0.1.100...
Connected to fire-in.yoyodyne.com
Escape character is '^]'.
Username: hikita
Skey challenge: s/key 502 fi34762 SILK SCAR DES DON JOEY RUNT
Login Accepted
fire-in.yoyodyne.com ck-gw proxy (Version 3.1) ready:
ck-gw->
```

In this example, Robert Hikita, working at a machine (**dimension**) inside the perimeter needs to access an Oracle database on a machine outside the perimeter. He first TELNETs to the port (*ck-gw*) on the firewall for Yoyodyne (**fire-in.yoyodyne.com**) on which the circuit proxy is running. The circuit proxy prompts Robert for his authentication userid, which he provides (**hikita**). When the proxy responds with a challenge, he enters his S/Key response. The proxy authenticates him using the appropriate authentication server and provides him with a circuit proxy prompt.

Next, he selects the service he wants to use:

```
ck-gw->services
Valid services are:
oracle
finance
reservations
ck-gw->c oracle
waiting for oracle client to be started (type 'q<return>' to abort)...
```

Robert uses the services command to view a menu of available services. He indicates he wants to connect to the Oracle service (**c oracle**).

The circuit proxy waits as he starts his client application.

Then he starts his client application. Because Yoyodyne is using transparency (the default configuration), he indicates that the database server is on the remote host (**db.clientsite.com**). If Yoyodyne was not using transparency, Robert would tell the client that the database server was the inside address of the firewall (**fire-in.yoyodyne.com**), allowing the firewall to connect to the database server on his behalf.

He confirms the connection in his original TELNET authentication session:

```
waiting for oracle client to be started (type 'q<return>' to abort)...
oracle client started
okay to proceed (answer yes only if you started a oracle client)? y
ck-gw->
```

Robert returns to the original TELNET window in which he connected to the circuit proxy. He notes that the circuit proxy has received a request for service. He confirms the request (**y**). He leaves this TELNET window active while he works so that the circuit proxy remains active.

Finally, the proxy connects to the remote application server, and begins passing information between the client and server.

When he is done using the application, he close the application and the original TELNET window.

Chapter 15

Managing Information Services on the Firewall

Sometimes it is not feasible to run a separate WWW or Gopher server outside your firewall. Because of hardware or other constraints, you cannot devote a separate machine to be your WWW server. Or, you do not expect enough traffic to justify another machine, but still want to offer WWW services to your customers. Instead, you want to run the WWW server securely on the firewall itself.

Warning: Most WWW servers are large and evolving programs, making it harder to ensure that they do not have any security holes. Inclusion of WWW software on a firewall is inherently contrary to the prudent security practice of running only the minimum trusted software necessary. It is a good idea to perform a careful risk assessment before placing WWW software on a firewall.

The Info Server included with the Gauntlet Internet Firewall services requests for HTTP, Gopher, and FTP services.

This chapter explains how the Info server and Info Proxy work, how to configure the server and the proxy for the various protocols, and how to use the server and the proxy.

Understanding the Info Server

The Gauntlet Info Server is a minimal information server. The server, which runs on the firewall, works with a set of management tools to service HTTP, Gopher, and FTP requests. You can configure the server to allow connections based on:

- source IP address
- source hostname

You would use the Gauntlet Info Server in place of another HTTP server (such as the CERN or Netscape HTTP servers), Gopher server (such as the University of Minnesota Gopher server), or the FTP server included with your operating system.

The Gauntlet Info Server implements a minimalist design, in which the server handles only the file requests. A variety of management tools (on a per-service basis) actually provide the data. These smaller programs are easier to analyze and verify that there are no holes. Simpler code is easier to verify.

How It Works

The following sections describe how the InfoServer works.

HTTP and Gopher Server

When serving as an HTTP or Gopher server, the Info Server (*info-gw*) runs on the firewall as a daemon listening for TCP-based requests on port 8000. When the firewall receives a request, it forks a child copy of the Info Server, leaving the parent Info Server to continue listening for requests.

The child Info Server process looks at the request and places it in one of several categories (such as Gopher or HTTP).

It checks the appropriate configuration information (in the *netperm-table*) and determines whether the requesting host has permission to use the desired service. If not, the Info Server logs the connection and displays an error message.

If the host has permission to use the service, the Info Server uses its internal database (by default in */usr/gauntlet/infodb*) to find the requested file or to go to the requested directory. The client thinks it is talking to a regular HTTP or Gopher server, even though it is not.

FTP Server

When serving as an anonymous FTP server, the Info Server runs in conjunction with the network access control (*netacl*) daemon. In this scenario, the IRIX system runs the network access control daemon (*netacl*) as a daemon listening for requests on the standard FTP port (21). Whenever the firewall receives a FTP request on this port, the *netacl* daemon checks its configuration information (in the *netperm-table* file) and determines whether the initiating host has permission to use FTP. If the host has permission, the *netacl* daemon starts the standard FTP proxy (*ftp-gw*) or the Info Server

(*info-gw*) depending upon the originating host. If the host does not have permission, the daemon displays an error message.

This allows outside users to FTP to the firewall and access the Info Server as an anonymous FTP server. Inside users can access the FTP proxy if they need to FTP files from one side of the perimeter to the other. The network access control daemon determines the appropriate program to start, based upon the host sending the request.

Once the connection is made to the Info Server, It checks the appropriate configuration information (in the *netperm-table*) and determines whether the requesting host has permission to use the desired service. If not, the Info Server logs the connection and displays an error message.

If the host has permission to use the service, the Info Server uses its internal database (by default in */usr/gauntlet/infodb*) to find the requested file or to go to the requested directory. The client thinks it is talking to a regular FTP server, even though it is not.

How the Database Works

When the Info Server processes a request, it does not use standard directory commands to traverse the file hierarchy on the firewall. Instead, the Info Server uses a database manager, which translates the FTP, HTTP or Gopher request into the internal database structure. The database manager then tells the Info Server the actual name of the file, which the server displays or returns to the client. The database uses */usr/gauntlet/infodb* as the root directory for the database.

The database structure restricts the number of characters that can exist in a filename and translates others. It uses particular letters to designate particular types of files and directories. The database uses the first letter of file names and directory names to indicate the type of file or directory type.

Directories

The database structure only recognizes directories that start with the letter D. When the Info Server receives an HTTP request for a file in the *images* directory, the database manager translates the request and looks in the *Dimages* directory.

The database structure also translates other characters in directory names. It translates the dot (.) character in filenames to the zero (0) character. When the Info Server receives a request to go the directory */../../etc*, the database manager translates the request and

looks for the directory *D/D00/D00/Detc*. Because the root directory of the database is actually */usr/gauntlet/infodb*, the Info Server is actually looking for */usr/gauntlet/infodb/D/D00/D00/Detc*.

The Info Server always looks for files within its own directory tree. It does not and *cannot* move back out of its directory tree to other areas of the systems, as some HTTP, Gopher, or FTP servers might.

Data Files

The database structure only recognizes data files that start with the letter A. When the Info Server receives a request for the file *readme*, the database manager translates the request and looks for the file *Areadme*. HTTP, Gopher and FTP requests all return these files.

The database structure looks for HTTP header files (for HTTP version 1.0) in files that start with H.

The database structure also limits the characters that can exist in filenames. It translates the dot (.) character in filenames to the zero (0) character. When the Info Server receives a request for the file *latest.gz*, the database manager translates the request and looks for the file *Alatest0gz*.

In many cases, the files that start with A and H are actually symbolic links to the real text or binary file. For example, the file *Alatest0gz* would actually be a symbolic link to *latest.gz*. For text files, the A file is generally a copy of the actual data file with every line terminated with a carriage return / line-feed pair. You don't need to create files specifically for use with the Info Server. You merely need to create symbolic links or copies of the files that the database understands. This process is described below in the section "Creating Files"

Queries and Executable Programs

The database structure only recognizes query programs and executables that start with the letter Q. When the Info Server receives an HTTP request that contains a query, such as *animals?dogs*, the database manager will translate the request and try to run the program *Qanimals*. The database manager passes all of the information after the query marker (? for HTTP requests and a <Tab> for Gopher requests) to the query program.

The Q files are generally symbolic links to the executable program. For example, the program *Qimagemap* could actually be a symbolic link to *imagemap*.

FTP Directory Lists

The database structure limits the clients view of what is in the database and what is available on that server. When the Info Server receives a request to list the contents of a directory, it instead returns specific files that contain the directory listing that you wish to display. For example, when the Info Server receives an FTP LIST (*ls*) request, the database manager translates the request and returns the file *L* in the current directory. The client requesting the directory listing sees a list of files that looks like the list of files you might see on any other FTP server. Similarly, the database manager translates an FTP NLIST (*nlist*) request and returns the file *N* in the current directory.

The *L* and *N* files are actual files that contain directory listings. You create the files, listing only those files that you wish to display. For example, the *L* file could contain only the list of files that you want anyone to view, even though you have other files in the directory.

Gopher Menu Files

When the Info Server receives a request to display a Gopher menu, it instead returns a specific file that contains the list of files that you wish to display for that directory. For example, when the Info Server receives a Gopher request for the menu in a directory, the database manager translates this request and returns a file beginning with G in the current directory. The client displays a menu of files that looks like the list of files you might see with any other Gopher server.

The *G* files are actual files that contain Gopher menus. You create the files, listing only those files that you wish to appear in the menu. For example, the *Gmenu* file could contain only the list of files that you want anyone to view, even though you have other files in the directory.

Configuring the Firewall

Configuring the Gauntlet firewall to run an Info Server involves planning, indicating which daemons the system will run, configuring the Info Server to enforce your policy, and verifying your setup.

Planning

1. Determine which services (HTTP, Gopher, FTP) you wish to offer.
2. Determine whether you wish to allow FTP access to sites inside your security perimeter as well as to the firewall. If you wish to allow both, you must use the netacl daemon to start either the FTP proxy or the info server.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support the Info Server. This is a standard service, included in the default versions of these configuration files on the Gauntlet firewall.

Configuring the Proxy Rules

If you are using the Gauntlet firewall default configuration, you do not need to modify the proxy rules for the info server. To enable the info server, use the *gauntlet-admin* Proxies form to enable the info server, select an idle timeout period, and specify an information directory. Enable anonymous FTP if desired.

Alternatively, you may modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on *info-gw* options, *netperm-table* options, and order of precedence.

Verifying Your Setup

Access your Info Server as you would any other HTTP, Gopher, or FTP server. Watch the log messages.

Using the Info Server

Managing the Info Server involves planning, placing files on the firewall and adding them to the database, creating list files, and advertising your server.

Planning

Determine who will put the files onto the firewall. Remember that if you want your WWW, Gopher, or FTP administrator to have access, you need to provide an account on the firewall, which is not recommended. Instead, make arrangements with your WWW, Gopher, or FTP administrator to periodically transfer files for them.

Creating Files

Create your text and executable files as you would for use with any HTTP, Gopher, or FTP server. You do NOT need to modify references to directories or to executables within your documents.

Placing Files on the Firewall

To set up your files for use with the Info Server on the firewall, follow these steps:

1. Create your directory structure under */usr/gauntlet/infodb/D*. Prefix each directory with the letter D when you create the directory. For example, if you want to keep all of your pictures in the images directory:

   ```
   firewall-32# cd /usr/gauntlet/infodb/D
   firewall-33# mkdir Dimages
   ```

2. Copy all of your files (HTML, text files, executables, and pictures) to the appropriate directory.

Adding Files to the Database

This process creates the A and H files for HTML files, the Q files for queries, and so forth. The process differs slightly for text and binary files.

Text Files

Adding text files to the database creates the necessary A and H files in the database. Use the *addtext* program (*/usr/gauntlet/infodb/tools/addtext*).

To add text files to the database, create the *A* and *H* files:

addtext *file* [ct*filetype*]

where *file* is the name of the text file, and ct*filetype* is one of the default header file types used to create an HTTP version 1.0 header file:

- chtml—HTML text header (default)
- cttext—Text header

Consult */usr/gauntlet/infodb/tools* for a list of currently available sample headers. Use these files as templates to create your own header files, if necessary.

Repeat this process for every file you wish to have accessible via the Info Server.

Binary Files

Adding binary files to the database creates the necessary A and H files for images. Use the *addfile* program (*/usr/gauntlet/infodb/tools/addfile*).

To add binary files to the database, create the A and H files:

`addfile` *file* `[ct`*filetype*`]`

where

- *file* is the name of the binary file
- ct*filetype* is one the default header file types used to create an HTTP version 1.0 header file:

  ctavi—AVI movie header

  ctgif—GIF image header

  cthtml—HTML text header

  ctjpg—JPEG image header

  ctps—PostScript header

  ctqt—QuickTime movie header

  cttext—Text header

  ctzip—ZIP header

Consult */usr/gauntlet/infodb/tools* for a list of currently available sample headers. Use these files as templates to create your own header files, if necessary.

Repeat this process for every binary file you wish to have accessible via the Info Server.

Query Files

Adding query files to the database creates the necessary symbolic links for the query file.

To add query files, create the symbolic link

`ln -s` *file* **`Q`***filename*

where

- *file* is the path and file of the actual query executable
- Q*filename* is the name of the executable prepended with a Q and any periods converted to the zero (0) character.

Repeat this process for every binary file you wish to have accessible via the Info Server.

Creating FTP List Files

Creating list files actually creates the L and N text files that the Info Server displays when it receives FTP *ls* and *nlist* requests. Use the *makedirlist* script (*/usr/gauntlet/infodb/tools/makedirlist*).

To create list files, run the *makedirlist* script in the appropriate directory.

Repeat this process in each directory in which you wish to have directory listings.

Creating Gopher Menu Files

Creating Gopher menu files actually creates the text file that the Info Server displays when it receives a request for a Gopher menu.

To create Gopher menus, follow these steps:

1. Execute the list command and redirect it to a file that starts with G. You may wish to restrict the files that the command displays, so that it looks like a normal Gopher menu. See the *makedirlist* script for examples of redirecting list files to text files for the Info Server.
2. Modify the resulting file and add the other standard Gopher menu fields.

Advertising Your Server

Advertise your HTTP, Gopher, or FTP Server to your customers or the world. Be sure to

- advertise the outside IP address of the firewall
- specify that connections should use port 8000 for HTTP and Gopher requests

Chapter 16

Using the Network Access Control Daemon

The proxies included with the Gauntlet Internet Firewall allow you to determine whether or not you wish to allow certain hosts to access certain services. If you permit a particular host to use the TELNET proxy, they can TELNET from trusted networks to untrusted networks. If they do not have permission, the proxy displays an error message.

Sometimes, you wish to provide additional alternatives. You want to allow some hosts to use one service, while other hosts use a different service or proxy on the same port. The network access control daemon allows you to do just this.

This section explains the concepts behind the network access control daemon, how it works and how to configure it.

Understanding the Network Access Control Daemon

The network access control daemon is a TCP wrapper program that provides configurable access control and logging mechanisms. The network access control daemon, which runs on the firewall, starts different applications based on the source address of the request.

Using the network access control daemon, you can allow certain hosts to access a standard UNIX program, such as the TELNET daemon, while requests from all other hosts access the TELNET proxy.

The network access control daemon allows you to configure which hosts have access to which TCP-based services. Note that it does not allow you to start UDP-based services. The access control daemon logs all successful and unsuccessful connections.

How It Works

The firewall runs different instances of the network access control daemon (*netacl*) on different ports for different applications, based on the information in the */usr/gauntlet/bin/gauntlet* file. The */usr/gauntlet/bin/gauntlet* file indicates which services should run on which ports. For example, the firewall runs an instance of the network access control daemon on port 23 to handle TELNET requests.

When the network access control daemon receives a request on a port on which it is listening, the daemon checks its configuration information (in the *netperm-table*) and determines whether the initiating host has permission to initiate this type of request. The network access control daemon then verifies that it has permission to run. If the host does not have permission or the network access control daemon is not permitted to run, the firewall displays an error message.

If the host has permission and the network access control daemon is permitted to run, the network access control daemon then starts the program specified in the *netperm-table*. For example, the network access control daemon might start the TELNET proxy (*tn-gw*) for some initiating hosts and the actual TELNET daemon (*telnetd*) for other initiating hosts.

The default configuration of the Gauntlet Internet Firewall uses the network access control daemon to control access to several different proxies and daemons. For example, the default configuration of the Gauntlet Internet Firewall uses the network access control daemon to control access to finger services. The network access control daemon allows hosts on the inside network to use the UNIX finger daemon (*fingerd*) to gather information from hosts outside the perimeter. However, for requests from the outside networks for finger service, the network access control daemon calls *cat* to display a message stating that the firewall does not accept finger requests.

Configuring the Network Access Control Daemon

Configuring the Gauntlet firewall involves planning, indicating which daemons the system will run, configuring the proxy to enforce your policy, turning on your proxy, and rebooting your firewall.

Consult the existing UNIX and Gauntlet firewall configuration files for examples of the network access control daemon in use. This section describes using the network access

control daemon to accept mail from a restricted list of sites. This involves restricting access to the SMTP client proxy (*smap*).

Planning

Determine your policies for which hosts will use which service.

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support NNTP. This is a standard service, included in the default versions of these files on the Gauntlet Firewall.

Configuring the Proxy Rules

In most cases you do not need to modify the proxy rules for NNTP. This is a standard service.

Configuring Your Service

Ensure that the other program you wish to run exists, has appropriate file permissions, etc. For example:

1. Create a file (*/usr/etc/smtp-deny.txt*), using SMTP syntax, that the network access control daemon can display for SMTP requests from the offending hosts.

   ```
   521-Mail from your system is not permitted through the firewall
   521 For more information contact XXXXX at XXX-XXX-XXXX.
   ```

2. Ensure that the file has appropriate permissions:

   ```
   chmod 444 /usr/etc/smtp-deny.txt
   ```

Verifying Your Setup

Verify your setup by accessing your firewall from each set of hosts. Watch the logs on the firewall for error messages.

Chapter 17

The Graphical Management Interface

The Gauntlet system uses a Web browser interface (“forms-based”) to make it easy for you to quickly configure and run the firewall system. The Gauntlet management interface supports all common Gauntlet administrative functions and is organized (like this chapter) into the following browser forms:

Note: You can modify directly some of the files that this interface configures. However, doing so could mean that you may no longer use the GUI as your management interface. Refer to Appendix A for more information.

Caution: If you browse the Internet from the firewall, you should use a browser with Java and Javascript turned off.

First Time User Tips

Each section in this chapter describes a Gauntlet management form. The forms-based interface is designed to be self-sufficient, and it may present enough information for you to make all appropriate configuration decisions. While this chapter provides additional background information, it also duplicates much of the information that is available on the forms.

For initial configuration, you might want to step through the administration forms in sequential order. To step through the forms in this manner, click the *Continue* button at the bottom of each form as you finish with it. You can return to the previous form by clicking *Back*. As you become familiar with the interface and your own configuration, you might prefer to go directly to a particular form in a random order. You can do this by clicking the name of the form in the menu bars that appear at the top and bottom of every form in the graphical management interface.

Help Links

To view additional information on many subjects, select any highlighted (linked) word or phrase on the form.

Caution: If you create or change an entry and use a help link before saving your entry, your changes will be discarded.

Hide and Unhide Buttons

You can “unclutter” forms by hiding sections that you are already familiar with or that do not concern you. To hide a section of a form, click the *Hide* button, shown in Figure 17-1.

Figure 17-1 Hide Button

The selected area is hidden from view and is represented by an *Unhide* button, shown in Figure 17-2. Click the *Unhide* button to display more detailed configuration information on the corresponding section.

Figure 17-2 Unhide Button

Caution: Clicking *Hide* or *Unhide* causes any unsaved changes on a page to be discarded.

Gauntlet Default Settings

Many (but not all) forms provide defaults that are likely to suit your firewall configuration. Defaults are conservatively chosen so that network services are disabled until you specifically enable them.

When to Use Configure All

Choose *Configure All* only after you are sure that all forms are set up as you want them. When you are sure of your setup, choose this option to put your firewall configuration in effect.

Caution: Running “Configure All” interrupts all current connections, including the telnet session if you are using one to manage Gauntlet remotely.

Using the Gauntlet Management Interface

To configure the Gauntlet firewall, you can start the management interface locally from the firewall itself or from a remote host (including a remote X display). You can also perform secure remote management. However, the first time that you use the management interface, you cannot use it remotely—you must perform the configuration by working directly on the firewall. Furthermore, you should always perform remote configurations on a host or X display on a trusted network.

The procedure that follows explains how to configure Gauntlet locally.

Note: If you have already configured Gauntlet and you wish to modify your configuration from a remote host instead of directly on the firewall, use the instructions “Configuring Gauntlet for Remote Administration” on page 168 or “Configuring Gauntlet for Secure Remote Administration” on page 170.

Configuring Gauntlet Locally

To start the management interface and configure Gauntlet locally, use this procedure on the firewall host:

1. Log in to the firewall as root.
2. Enter the *gauntlet-admin* command.

   The *gauntlet-admin* command starts the management interface. After your entry, a dialog box appears requesting the Gauntlet administrative user name and password.
3. Enter the *gauntlet-admin* user name and password.

   By default, the user name for the *gauntlet-admin* management tool is “gauntlet” and the default password is ”admin.” Enter the default user name and password to start the Gauntlet management interface.

   Note: We strongly recommend that you assign a user name and password other than the default, use this command:

   ```
   # gauntlet-admin -p
   ```
4. Follow the instructions in “Introductory Management Form” on page 118 to continue.

Introductory Management Form

Figure 17-3 and Figure 17-4 illustrate the Gauntlet introductory management form. This form is the entry point and the exit point of the forms-based management interface. From this form, you can go directly to any other management form or begin a sequential configuration sequence. When you have configured all forms as desired, you must return to this form and select *Configure All* to put your configuration entries in effect.

Caution: Do not select *Configure All* until you configure all forms appropriately—running “Configure All” interrupts all current connections!

The introductory management form describes how to use the forms-based interface, and contains a list of form names. From this list, you can access any other form, go to the next form, or configure your system.

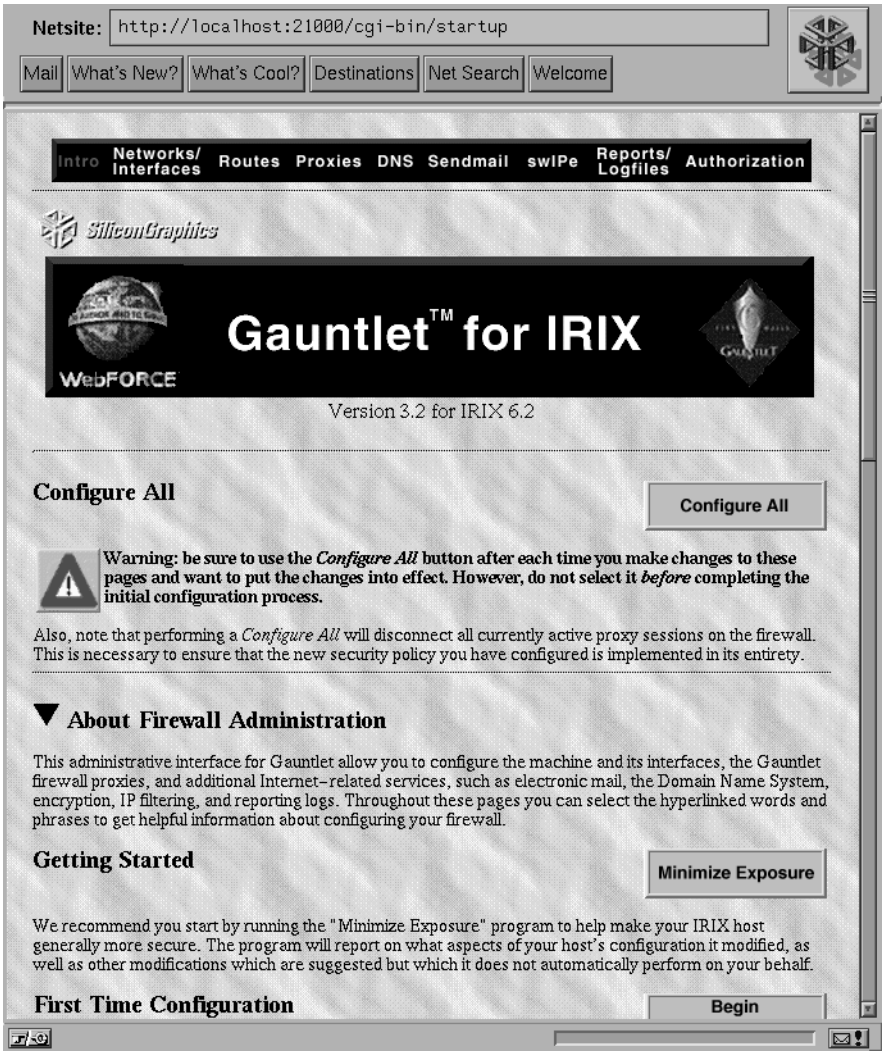

Figure 17-3 Gauntlet Introductory Management Form (1 of 3)

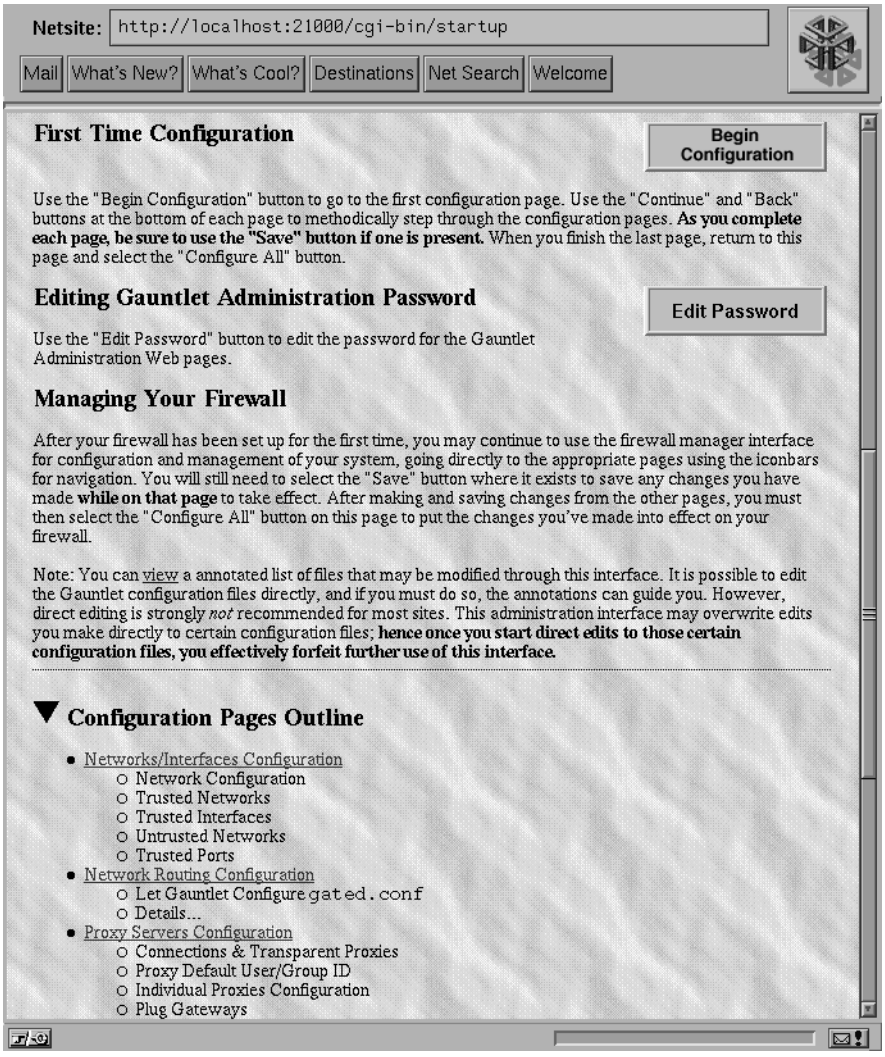

Figure 17-4 Gauntlet Introductory Management Form (2 of 3)

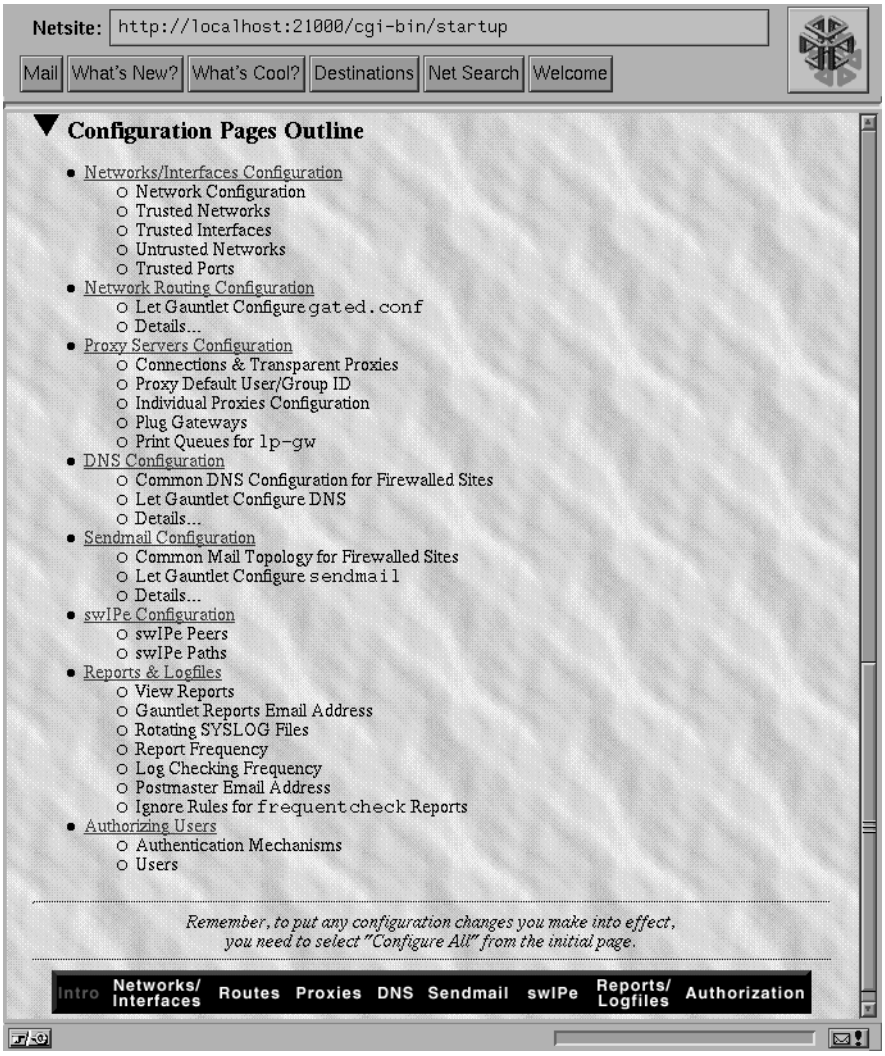

Figure 17-5 Gauntlet Introductory Management Form (3 of 3)

The section of the form called “Getting Started” provides a *Minimize Exposure* button that you can click to reduce possible security risks. If you click *Minimize Exposure*, the system reports on what it looks for and on any changes that you make during the session. If there are areas where it cannot make changes that you requested, the unchanged items are reported too.

You begin configuring your firewall in the “First Time Configuration” section by clicking *Begin Configuration*. But before you begin, read “Managing Your Firewall” for some information regarding direct file editing.

The last part of the introductory management form lists the names of the other browser forms. A list of links to these forms appears on the menu bar at the bottom of the form so you can go directly to another form, if you wish. This chapter explains each configuration form in the order that it appears if you click *Begin Configuration* on the introductory management form and then click the *Continue* button on each form that follows.

Networks and Interfaces Configuration Form

The Gauntlet networks and interfaces configuration form (Figure 17-6 and Figure 17-7) uses standard Silicon Graphics Network Setup tools to configure network interfaces on the firewall. If you have not already configured your network setup with these tools, click *Network Setup* to configure the firewall hostname, network interfaces, and IP addresses; click *ISDN Setup* if you want to configure ISDN. Click *PPP Setup* if you want to configure PPP.

Note: To run the Network Setup tools directly from the Gauntlet forms-based interface, you must be working at the Gauntlet host console. You can use the Network Setup tools without the Gauntlet interface from any location.

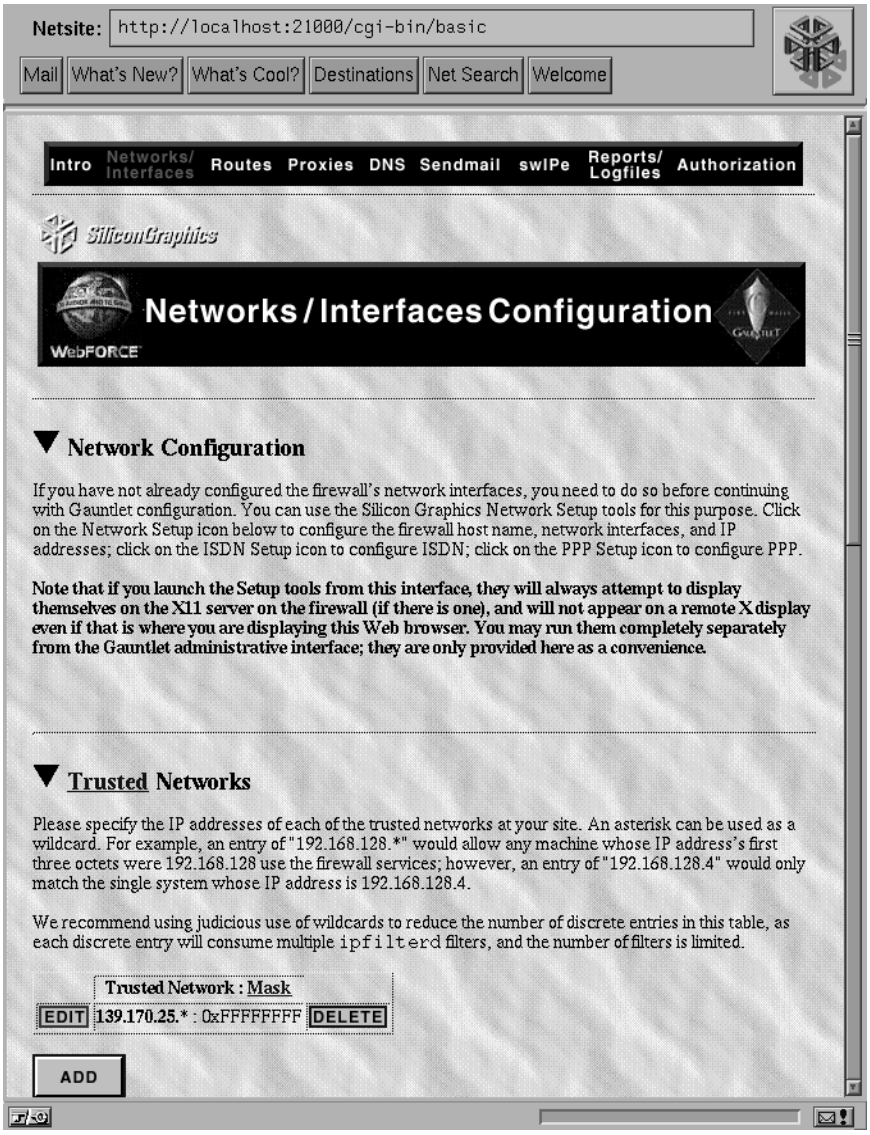

Figure 17-6 Networks and Interfaces Configuration Form (1 of 2)

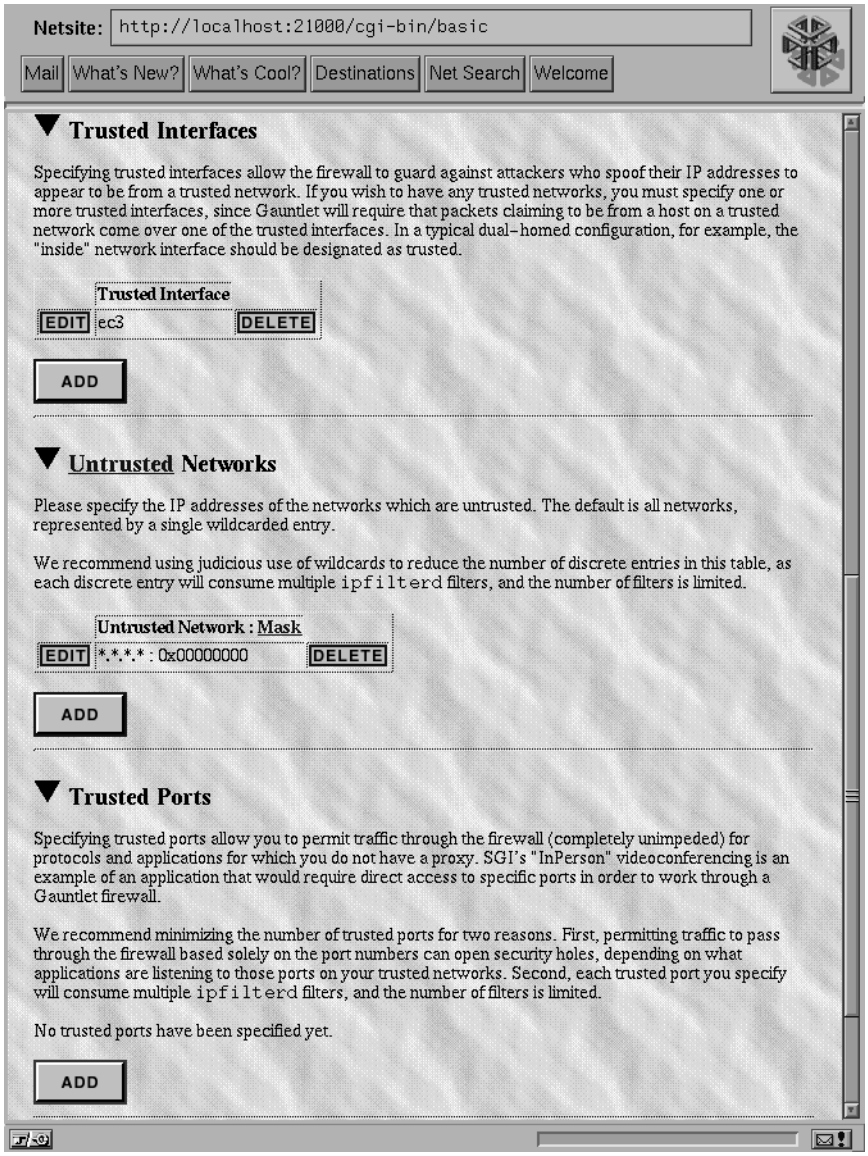

Figure 17-7 Networks and Interfaces Configuration Form (2 of 2)

The Gauntlet networks and interfaces configuration form allows you to specify trusted and untrusted networks. Until you make changes on this form, all networks are considered untrusted, and only the Gauntlet system itself is trusted.

You can use a terminating asterisk as a wild card to represent “all” in network addresses. These examples illustrate the use of the asterisk:

- 192.168.128.*—all IP addresses beginning with “192.168.128”
- 192.168.*—all IP addresses beginning with “192.168”
- *—all IP addresses

Note: Only a terminating asterisk is allowed: an entry such as 192.*.128.* is not valid.

The default subnet mask automatically provided by the GUI for trusted and untrusted networks is 0xFFFFFF00, which is the correct mask for a non-subnetted Class C network. If this mask is not correct for your configuration, click *Edit* and modify the mask field to change it.

Trusted Networks

The Gauntlet firewall supports the concept of “trusted networks,” networks whose users are permitted to access firewall services without user authentication (see “Authorizing Users Form” on page 163). Typically, trusted networks are your internal, local networks.

To add networks to the trusted list, click the *ADD* button; then specify the IP address of each network that you want to add to the trusted list.

Trusted Interfaces

Specifying trusted interfaces (interfaces where trusted networks are connected) allows the firewall to guard against IP address spoofing, a ruse in which network packets are tagged with a falsified trusted network address. When you designate trusted interfaces, Gauntlet verifies that packets tagged with a trusted network address actually arrived on a trusted interface.

Note: Specifying trusted interfaces is required if you have any trusted networks.

On a dual-homed configuration, the inside interface (the connection to your internal network) is normally designated as trusted. For example, in Figure 17-9, the inside interface connects to internal network 192.132.134.*, a trusted network. To designate a trusted interface, click the *ADD* button and enter its interface name (such as ec1).

Untrusted Networks

Untrusted networks are those whose users are permitted to access network services after performing authentication. Networks that are neither trusted nor untrusted are considered unknown. Not only are users on unknown networks denied network services, they are not even prompted for authentication information—access is immediately refused.

The default entry for untrusted networks is *.*.*.*, which designates all networks that are not trusted (including unknown networks) as untrusted.When this default is in effect, all users from outside networks are permitted access to network services provided they pass authentication.

You can add to the list of untrusted networks by clicking the *ADD* button. Remember that when you designate one or more untrusted networks, users on these networks are allowed access with authentication; all remaining outside networks are considered unknown and their users are refused connections. If you leave the list of untrusted networks blank, all network are considered unknown; all network users other than those on trusted networks are refused access.

Note: Individual users from an untrusted or unknown network can be permitted to access network resources. These users must be listed in the authentication database (see “Authorizing Users Form” on page 163).

Trusted Ports

Specifying trusted ports allows you to permit unimpeded traffic through the firewall from protocols and applications that have no proxy. InPerson™ is an example of such an application—it requires direct access to specific ports to work through a Gauntlet firewall.

Note: Trusted ports can be configured only when the Gauntlet firewall is acting as the router between internal and external networks.

Routing Configuration Form

Use the routing configuration form (Figure 17-8) to specify your routing implementation. If you already customized a routing configuration file, *routed.options*, and want to continue using it on the Gauntlet host, check the box for "Preserve the routed configuration?"

If you plan to let Gauntlet create a new *routed.options* file, click *ADD* under *Explicit Routes;* then add the network, gateway, and "hop" metric to each network you add. Use a metric of "0" if the gateway is an interface on the Gauntlet host, and a metric of "1" if it is anywhere else. Explicit routes are stored in */usr/gauntlet/config/explicit_routes*.

To set the default route to a network, enter "default" as the destination network and 0X00000000 as a network mask.

The default subnet mask automatically provided by the GUI for the destination network(s) is 0xFFFFFF00, which is the correct mask for a non-subnetted Class C network. If this mask is not correct for your configuration, click *Edit* and modify the mask field to change it.

By default, the */etc/gateways* configuration file is not set to advertise routes on untrusted networks.

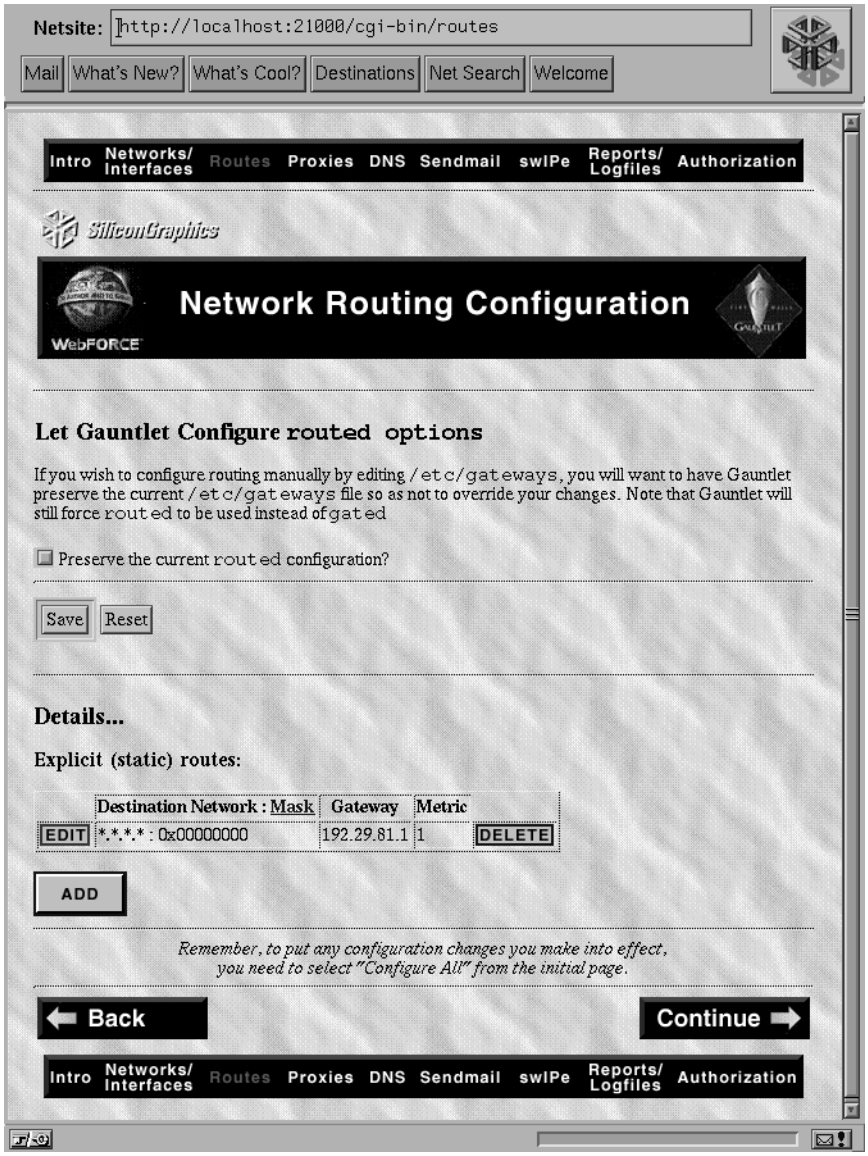

Figure 17-8 Routing Configuration Form

Figure 17-9 illustrates an example routing configuration for a Gauntlet firewall with two network interfaces. In this example, the firewall forwards packets from the internal network (192.132.134.*) to interface is 192.132.122.11 (the router) by way of interface 192.132.122.12 (a gateway to the external network). One hop is required for packets to reach this destination. Gateway 192.132.122.12 forwards packets from external networks to hosts on the internal network by way of router 192.132.134.11, which is the default destination for all inbound packets.

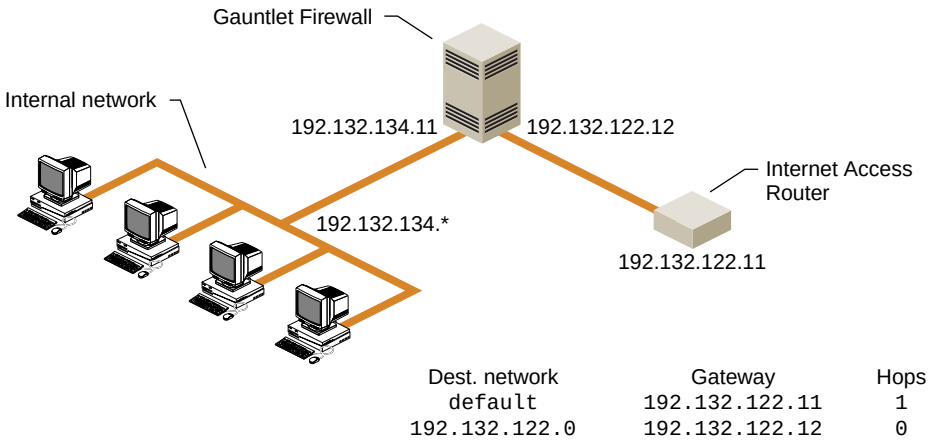

Figure 17-9 Example Gauntlet Host Routing Configuration

If hosts on your internal network are running a routing daemon, they eventually acquire the default route from the Gauntlet host. The default route can also be explicitly assigned to those hosts by their administrators.

Additional Routing Information

For additional general routing information or information on routing using IRIX commands, refer to the section “Setting Up a Router” in *IRIX Admin: Networking and Mail* and the reference page for routed(1M).

Proxy Servers Configuration Form

The proxy server configuration form (Figure 17-10 and Figure 17-11) allows you to control network services that are available through the Gauntlet firewall. You can enable and disable particular services, specify timeout values and port numbers, and so on. Each service can be configured separately.

Remote (Network) Connections

If you want to allow network logins to the firewall, check “Do you want connections allowed to the firewall itself?” If this box is not checked, you must configure the firewall at the system console—not from a remote location. Remote logins are convenient, but they can lessen the security of the firewall.

When logins are enabled, administrators can connect to the firewall by accessing the rlogin or TELNET proxies. Example 17-1 illustrates a sample TELNET session.

Note: The preferred method for managing the firewall remotely is described in “Introductory Management Form” on page 118 and “Configuring Gauntlet for Secure Remote Administration” on page 170.

Example 17-1 Administrative TELNET Connection to Firewall

```
Trying 204.254.155.253
Connected to firewall.yoyodyne.com.
Escape characther is ^]
tn-gw> connect localhost
Trying 127.0.0.1 port 23...
Connected to localhost.

IRIX System V.4 (firewall)

login: root
Password:
IRIX Release 6.1 IP22 firewall
Copyright 1987-1994 Silicon Graphics, Inc. All Rights Reserved.
Last login: Wed Aug 16 14:05:49 PDT 1995 by UNKNOWN@localhost
You have mail.
firewall 1# setenv DISPLAY magnolia.yoyodyne.com:0
firewall 2# gauntlet-admin
```

Note: If you log in from the network to the firewall host (you must have enabled network logins to do so), you may need to set the DISPLAY environment variable to your host to use *gauntlet-admin*.

Caution: Use remote logins only over secure links when absolutely necessary. You can also allow remote access to the firewall by connecting a modem to one of the serial ports to enable controlled dial-in access for administrators only.

Enabling Transparent Proxies

You must specify whether you want to enable transparent proxies. Transparent proxies allow users to connect to supported services on outside hosts through a proxy server, as if the outside hosts were local. If you do not enable transparent proxies, users must first connect to the proxy server and then to the network host providing the service.

Enabling Individual Proxy Services

Click the enable box beside any service name to enable a proxy for the service. When you enable a service, the firewall runs a daemon to support it. For example, enabling TELNET means that a proxy TELNET server will run on the Gauntlet firewall to mediate and enable TELNET connections. The proxy will be a transparent TELNET proxy if you have enabled transparent proxies.

Note: You must also have configured the Networks/Interfaces Configuration Form correctly for the TELNET service to work.

Many services allow you to specify a timeout value—the timeout value is the number of seconds that the server maintains an idle connection before dropping it.If you do not see a timeout field, click the *Unhide* button to display it. Change the default timeout value of any service if it does not suit your needs.

FTP Server Configuration

If you enable FTP on the firewall, you can specify a timeout value and also specify if you want to enable anonymous FTP. The Gauntlet configuration sets up anonymous FTP according to the recommendations in “Setting Up Anonymous FTP” in *IRIX Admin: Networking and Mail*. If enabled, anonymous FTP prevents users on untrusted networks from using the FTP application proxy.

You can select which services, if any, to offer untrusted network users on the FTP port: ftp-gw, anonymous FTP using IRIX *ftpd*, and anonymous FTP using Gauntlet the info server.

TELNET

If you enable the TELNET proxy, enter a timeout value (number of seconds) for idle connections; or, accept the default value of 3600 seconds—one hour.

rlogin

If you enable the rlogin proxy, enter a timeout value (number of seconds) for idle connections; or, accept the default value of 3600 seconds—one hour.

X Windows, finger, gopher, and whois

Check the enable box beside any service name to enable the corresponding proxy server. X Windows is for use in conjunction with TELNET and rlogin proxies only. See x-gw(1M) for an example session. No further configuration is required.

HTTP Proxy Server Configuration

If you enable HTTP (hypertext transfer protocol for World Wide Web access), you must also specify the following:

- which port the HTTP server should use—the default is “8080”.
- which server the HTTP proxy defaults to for unqualified URLs (“unqualified” URLs are HTTP request from a browser that do not include a server name, just a path.)

If you want users inside the firewall to pass authentication to access the Web, click *Enable* to enable the authenticating proxy (AHTTP) also.

NNTP Proxy Server Configuration

Enable NNTP for USENET News access. If configured with the addresses of an internal and external news server, the firewall gateways NNTP traffic between the two systems in both directions. You can use either host IP addresses or DNS names in your entries.

When you configure news on the internal and external servers, set both systems to feed news to the firewall, rather than attempting to exchange it directly. For example, assume

that the internal news server is “nntp.sgi.com” with IP address 192.33.112.100 and the external news feed is “news.uu.net” with IP address 11.11.11.11. In this case, configure the proxy with the names (or addresses) of the news server and news feed, and then configure the news software on “nntp.sgi.com” to transfer articles to the firewall. The upstream news feed “news.uu.net” would also transfer articles to the firewall.

SMTP Proxy Configuration

If you enable smap (for *sendmail*), you should specify the following:

- an idle timeout for SMTP connections—the default is 3600 seconds
- which directory the SMTP server should use—the default is */var/spool/smap*
- which address to send bad e-mail to—the default is “root”

POP3 Proxy Configuration

The POP3 proxy allows users to retrieve e-mail from a company POP3 server on the internal network. This can be extremely useful if users are traveling, for example. Remote users must be using client software that supports POP3 APOP authentication. This allows users to authenticate themselves to the Gauntlet firewall, so the firewall can then “plug” the connection through to the internal POP3 server, performing the identical authentication exchange with the internal POP3 server. The user’s password to the POP3 server is independent of the firewall’s primary user authentication database. However, the password must be made known to the firewall (using the *apopkey* program), and it must be identical to the user’s password on the internal POP3 server. Refer to Chapter 3 for more information.

To enable the POP3 proxy, click *Enable* and specify a destination POP3 server. You might also want to specify a different timeout value.

lp Proxy Configuration

The *lp* proxy allows users inside the firewall to use printers outside the firewall; it also allows outside users to use inside printers. Click *Enable* if you want to proxy print requests through the firewall. Refer to Chapter 8 for more information.

Info Server Configuration

Click *Enable* if you want to use an info server instead of the usual FTP and/or HTTP servers as described in Chapter 12. If you enable an info server, you must also enter the location of an information directory on this form.

Remote Gauntlet Administration Proxy

Click *Enable* if you want to administer the firewall from a remote host using a Web browser. If you configured the firewall for SSL, your remote administration sessions will be security protected (see "Introductory Management Form" on page 118 and "Configuring Gauntlet for Secure Remote Administration" on page 170 for details).

Custom Configured Plug Gateways

Custom plug gateways allow you to create proxies for protocols that are not specifically included in the Gauntlet proxies group (see "Configuring the Firewall for Other Protocols" in Chapter 11 for more information). If you configured custom plug gateways, click *Enable* to enable them.

RealAudio Proxy

The RealAudio proxy allows clients inside the firewall to listen to audio files on outside servers. You cannot configure the proxy to allow outside clients access to RealAudio servers inside the firewall (see Chapter 13 for more information). Click *Enable* if you want to enable the RealAudio proxy.

MediaBase Proxy

The MediaBase proxy allows clients inside the firewall to access videos on a MediaBase server outside the firewall and play the videos as they are broadcast; it also allows outside clients to play videos from MediaBase servers inside the firewall. Click *Enable* if you want to enable the MediaBase proxy.

Sybase Proxy

The Sybase proxy allows clients inside the firewall access to database servers outside the firewall; it also allows outside clients access to Sybase database servers inside the firewall. Click *Enable* if you want to enable the Sybase proxy.

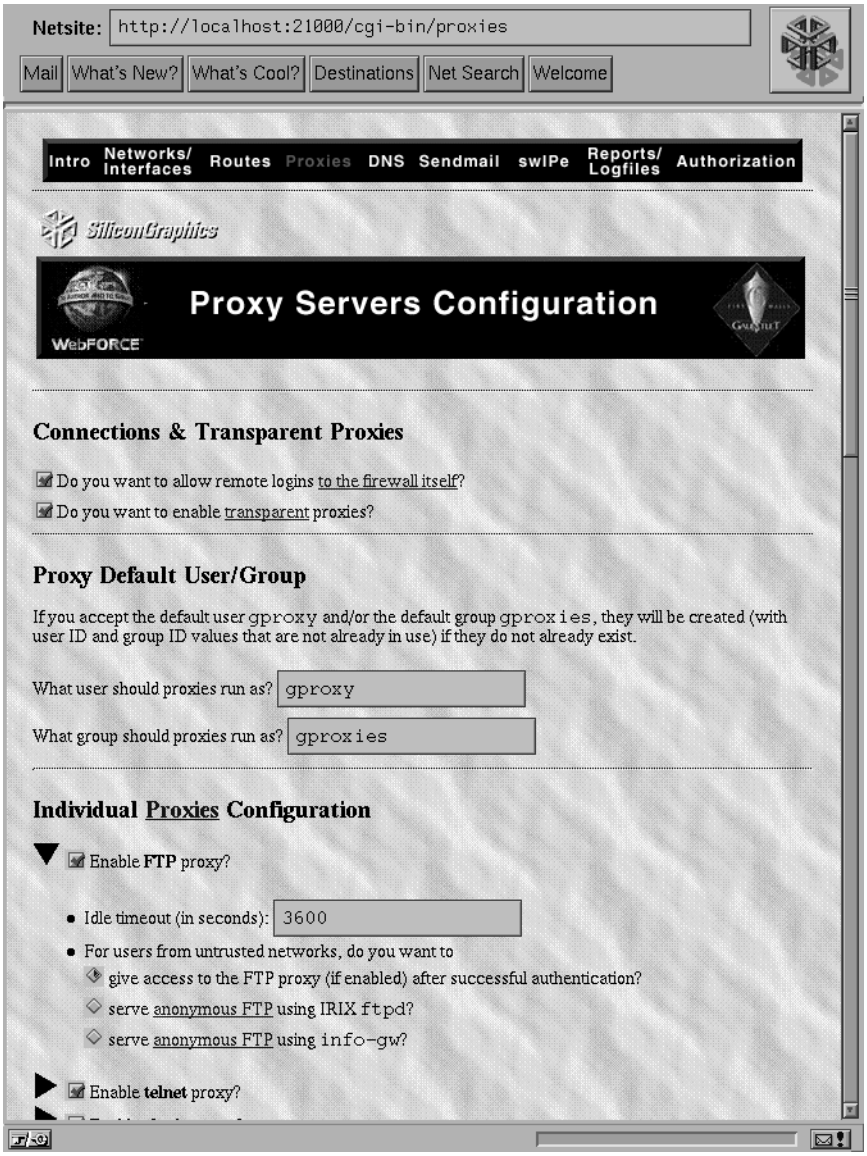

Figure 17-10 Proxy Servers Configuration Form (1 of 3)

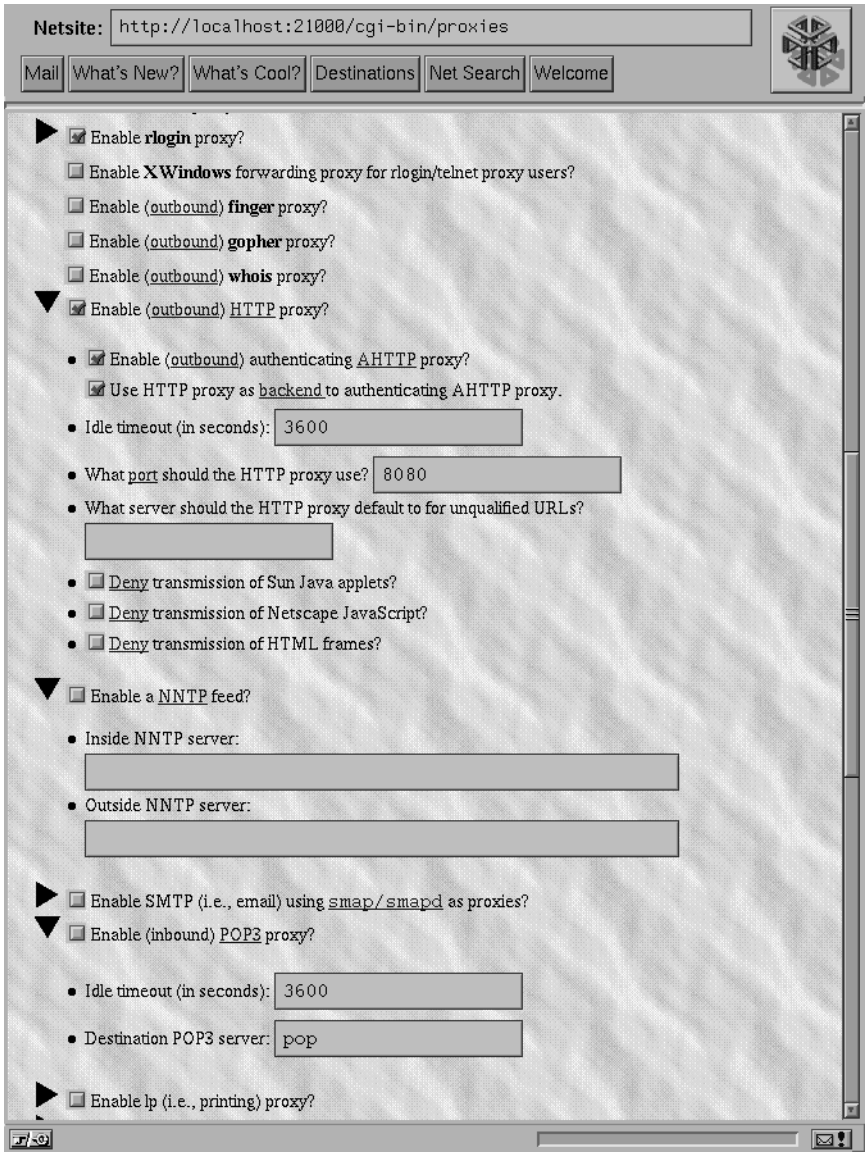

Figure 17-11 Proxy Servers Configuration Form (2 of 3)

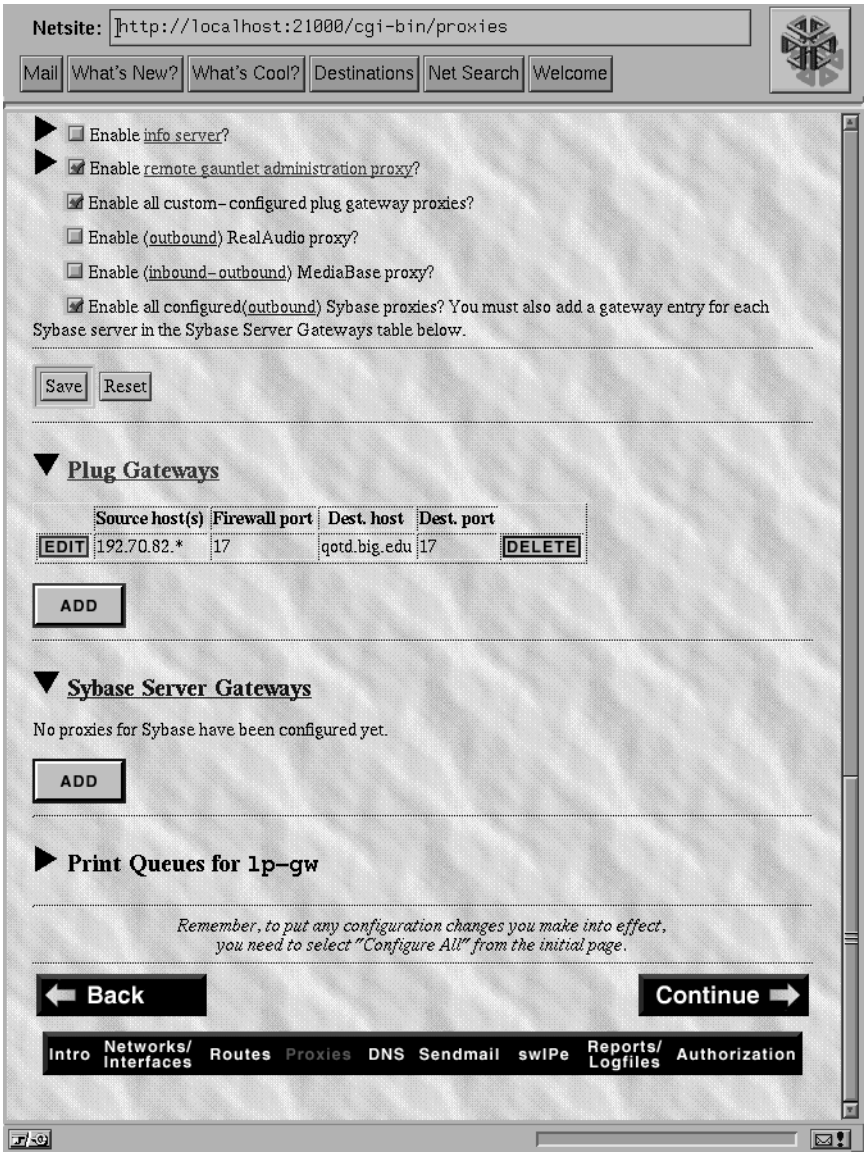

Figure 17-12 Proxy Servers Configuration Form (3 of 3)

Domain Name Service (DNS) and Gauntlet

When you join the Internet, you must participate in the Internet-wide DNS hierarchy (the name service used on the Internet). There are several popular methods of deploying your site's DNS information on the Internet: some sites have their service provider serve the information for them, while others run a local DNS server.

For sites that choose to run their own DNS server, there are two common firewall configurations. One involves running two DNS servers, an internal and an external server. This is often referred to as a split-DNS or dual-DNS configuration. The other configuration involves running a fully-populated DNS server on the external host. In either case, the Gauntlet host is commonly chosen to run a DNS server, either as the external member of a dual-DNS configuration, or as the single DNS server for the site.

DNS should be configured to provide the addresses that other sites need to contact you. This might include the address of your router, your firewall host, and any other hosts that must communicate with others. In the case of a simple firewall composed of a dual-homed host, the dual-homed host would be the DNS server that provides the address of the Internet side of its network connection (192.132.122 in Figure 17-9). In the case of a screened subnet, the DNS server could be any of the "public" hosts in the subnet, and it could provide addresses for all of these hosts and the router.

You should also set up the DNS Mail eXchanger (MX) record to advertise the name of the host(s) responsible for mail at your site. This might be the firewall host or some other host. Do not publish internal hostnames and addresses on the firewall host. If you have a single firewall host performing multiple services, say FTP and WWW serving, use CNAME records to "alias" the services to the hostname. This makes it easy to move these services to different hosts if you want to separate them later.

Configuring DNS is a task that is very difficult to automate reliably because DNS configurations vary widely among different sites. The purpose of the DNS configuration tools included with the Gauntlet firewall is to give the administrator a quick means of setting up a basic, working DNS. More advanced DNS management requires careful administrator attention and familiarity with the DNS software.

Gauntlet uses the Silicon Graphics example DNS configuration files to configure DNS for your firewall. If you are not sure how to fill in the DNS configuration form, refer to the chapter on "The BIND Name Server" in the *IRIX Admin: Networking and Mail*.

DNS Configuration Form

The DNS configuration form (Figure 17-13) helps you configure the files necessary to run a minimal DNS master server configuration for your site. This minimal configuration functions as the external server in a dual-DNS configuration, or as the basis for a site-wide server or other site-specific server. If you are the site-wide DNS server, add appropriate entries for each of the hosts on your network.

Caution: If you prefer to preserve your existing DNS configuration, be sure that the "Preserve the current DNS configuration?" box at the top of this form is checked. The default is to overwrite the current DNS configuration.

Configuring Fully Populated DNS Server

Enter the host name of your DNS server:

If you are planning to run an externally-visible DNS server on the firewall itself, enter one of the canonical external host names for the firewall. If you are running a separate externally-visible DNS server on a host on your DMZ, you should enter its host name here instead (if your Internet access provider provides your name service, specify their name server's host name.) Do not enter the host name of any internal DNS servers you may be running, as outside hosts cannot access them through the firewall.

The result is that the host name you enter is reported to be the authoritative name server for your domain name by the firewall's DNS server.

Note: The firewall must run a DNS server, because that's the only way that inside hosts can possibly get name resolution for the outside world. If part of your security policy includes hiding the names and IP addresses for your internal hosts, then you will also want to run an internal DNS server. See "Configuring a Split DNS Server" on page 142 for more information on how to configure the internal DNS server in this case.

Enter the IP address of your DNS server:

Enter the IP address which corresponds to the host name you gave above. If that's your Gauntlet firewall, then enter one of the external IP addresses for the firewall.

Enter the Internet domain name of your network:

Enter your domain name. For a Gauntlet firewall located “between” the Internet and the corporate network, the domain name would look something like example.com. For a Gauntlet firewall located “between” a divisional network and a corporate backbone network, the domain name might look something like corp.example.com.

The DNS server running on the firewall will claim to be authoritative for names and subdomains of the domain name you enter.

Enter the address of your network:

Enter the network address where the gateway portion of your Gauntlet firewall is attached. For example, in Figure 17-9, this address is 192.132.122.* The DNS server running on the firewall will claim to be authoritative for IP address lookups for the network with the address you enter.

Note: You may have internal hosts that use other IP addresses (registered or unregistered) in additional networks; that is acceptable.

Enter the host name of your mail hub:

The mail hub is the server where mail from your domain is collected, or “focused,” before it is distributed (see “Mail Hubs” on page 146 for possible mail hub configurations). The DNS server running on the firewall will advertise MX resource records that focus email addressed to any recipient in your domain (or any subdomain in it) to be delivered to the mail hub here specified.

Note: If you choose to run an internal domain-level mail hub, you still must specify the firewall’s external host name and IP address here. After initially selecting *Configure All*, select the button to preserve the existing configuration on the Sendmail (not DNS) page, and then consult *IRIX Admin: Networking and Mail* for details on how to get your Gauntlet firewall host to deliver all internal email through your separate domain-level mail hub.

Enter the domain name of your mail hub:

The domain name is ordinarily the name by which your organization is known on the Internet, such as *yoyodyne.com*.

Configuring a Split DNS Server

Split DNS is a configuration of two name servers: the inside server supplies name-and-address information only to internal hosts and to the firewall; the outside server is the firewall, which supplies name-and-address information to outside hosts to support applications such as mail and proxy connections.

The examples that follow use the following notation:

```
hostname:/path/filename
```

means

```
/path/filename
```

on the host hostname. The hostname “firewall” is used for the Gauntlet firewall host; “ns” is used for the internal DNS server host, assumed to be running the bind DNS server. (It is possible that various *bind* configuration files are not located at the paths given below on your ns host; consult system documentation and reference manual pages for correct locations.)

Use the following procedure to configure a split DNS configuration:

1. After initially selecting *Configure All* using the Gauntlet administrative interface, select and save the option on the DNS page to preserve the current DNS configuration files.
2. Edit the nameserver line in *firewall:/etc/resolv.conf* which currently lists the IP address for your firewall to list the IP address for ns.
3. Edit *ns:/etc/named.boot* to contain the additional lines

   ```
   forwarders GAUNTLET_IPADDRESS
   slave
   ```

   where GAUNTLET_IPADDRESS is one of the internal IP addresses for your Gauntlet firewall.
4. You should configure (for example, by editing */etc/resolv.conf*) all your internal machines, including on the host ns, to consider ns the DNS name server. For additional reliability or performance, you may, of course, configure additional machines to be DNS secondaries for the host ns; consult DNS and *bind* documentation for details on how to do so.
5. Edit *ns:/var/named/** files so the host ns can resolve all your internal hosts and internal IP addresses.

The configuration you have set up is often known as split-DNS, and is commonly used at firewalled sites. Outside hosts cannot successfully query your internal DNS server for internal host names and IP addresses. However, on the firewall itself, applications can resolve internal host names; this is necessary for using host names to direct email delivery and for inbound application proxy connections.

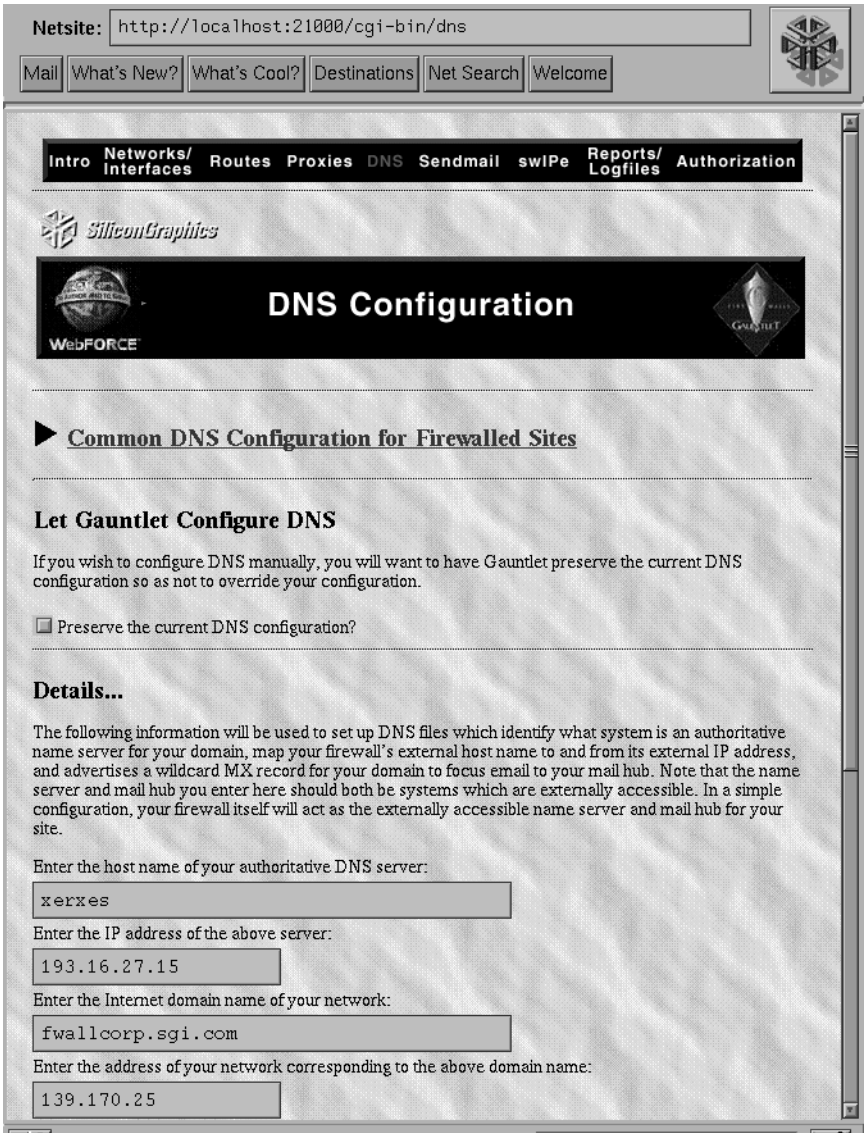

Figure 17-13 DNS Configuration Form (1 of 2)

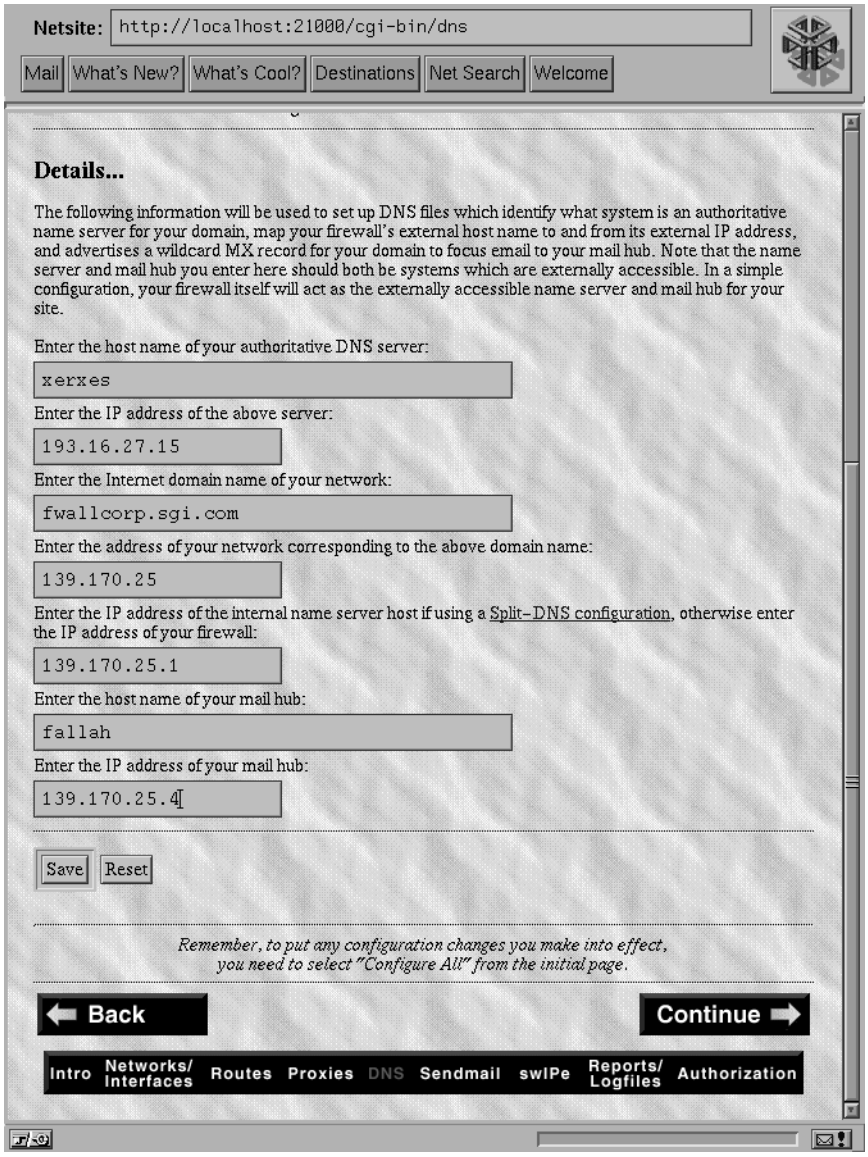

Figure 17-14 DNS Configuration Form (2 of 2)

Sendmail on Gauntlet Servers

Your mail system should be configured for compatibility with your DNS configuration. That is, the *sendmail.cf* file on the host that your DNS server advertises as your Mail eXchanger (MX) must be configured to accept mail for your network. This file must also specify what to do with mail after it is received. Usually, mail is forwarded to a master mail host on the internal network. This host knows the addresses of internal users and how to deliver mail to them.

Note: By convention, the domain name of your network is your electronic mail address. For example, user "Harry" at XYZ corporation, whose domain name is XYZ.com has the electronic mail address "harry@XYZ.com." To reinforce the electronic mail address of your sight and to make it easy for others to reply to your users' mail, configure your sendmail.cf to rewrite all addresses to conform to this convention.

Mail Hubs

When you have installed a firewall at your site, mail to any of the users on internal hosts must be focused—brought together—to pass through the firewall, and then delivered to the appropriate destinations. Whether or not in a firewall context, that is essentially what a mail hub is: mail bound for different destinations is focused together and delivered to the mail hub, and the mail hub figures out where the mail should go next.

You have three choices for where you can locate your domain-level main mail hub:

- outside the firewall (i.e., in your DMZ)
- on your firewall
- or inside your firewall.

Running a mail hub outside your firewall doesn't make very much sense, since it is more exposed (vulnerable to attack), and ultimately it needs to deliver mail through the firewall anyway, so it does not substantially improve security on the firewall. Furthermore, there do exist mailers on the Internet which do not follow the proper RFC's, and try to deliver email destined for username@domain_name to the machine domain_name, even if there is an MX record for domain_name pointing at your external mail hub machine. Unless you're willing to let such email bounce, you're going to have to be able to deal with email directly sent to your Gauntlet firewall anyway.

Having your firewall itself act as a mail hub is perhaps the simplest solution, and the Gauntlet administration interface sets up such a configuration without requiring any additional manual configuration file changes. The *sendmail* program has had a very checkered history—because of its complexity, it has historically been exploited and used to attack the machine it runs on. However, Gauntlet does not run *sendmail* to accept incoming email; rather, it runs a very simple program called *smap* that accepts and queues incoming email. This makes it much more difficult to attack the firewall using weaknesses in *sendmail*. If you implement this option, however, ensure that your */etc/aliases* file never executes arbitrary programs on the Gauntlet host (for example, `|/usr/local/bin/vacation ....`").

Mail Relays

In cases where mail delivery would impose a burden on the mail hub, the mail hub is frequently assisted by a mail relay. The mail relay is a host inside the firewall that receives mail from the hub and delivers it to another relay or to its final destination. When a network contains several relays, each relay is responsible for delivery to a particular group of hosts within the network.

Gauntlet and Subdomains

Using an internal machine as a domain-level main mail hub has some advantages if you have extremely complex mail processing needs. However, Gauntlet's support for recognized subdomains makes it easy for you to hand off complex mail processing tasks to specific machines while keeping your Gauntlet firewall as the main mail hub.

As an extended example of the Gauntlet recognized subdomains support, suppose your domain name is example.com and you want two recognized subdomains, one for corporate users and another for engineers.

- On the Sendmail page in the administrative interface, configure the firewall to recognize the two subdomains corp and engr.
- On your "internal" DNS server (which might be the firewall itself), you should create CNAME resource records for relay.corp.example.com and relay.engr.example.com to point to the mail hubs for the two subdomains. (The Silicon Graphics *sendmail.cf* file, used by Gauntlet for IRIX on the firewall host, considers the name relay to be special.) Also create MX resource records: for corp.example.com, list relay.corp.example.com; for engr.example.com, list

relay.corp.example.com. They are as much for informing internal sendmail or other analogous mail transfer agents as for informing sendmail on the firewall.

- The mail hubs that resource records point to should be able to deliver email to domains ending in SUBDOMAIN.example.com for the corresponding subdomain. (that is, in addition to handling addresses of the form name@SUBDOMAIN.example.com, they should be able to handle name@SOMEHOST.SUBDOMAIN.example.com, since your users might inadvertently give out their email addresses in fully-qualified form, for example, in their email signatures). Those hosts should also be prepared to forward all email to recipients with non-example.com addresses to the firewall for delivery.

  Then the SMTP From and the From: header lines in the email messages' headers on outgoing email messages (messages to non-example.com recipients that have been forwarded to the firewall host for delivery) will then be rewritten as documented in "Subdomain names to be recognized for your site:" on page 150.

Sendmail Configuration Form

Use the Sendmail configuration form (Figure 17-15) to modify the firewall's Sendmail configuration with a browser-based interface. If you prefer, you can use the IRIX *configmail* tool or edit the */etc/sendmail.cf* file directly. Be sure to check the *Preserve the current sendmail configuration?* button if you do this, because the default is to overwrite the current configuration.

If you plan to create a new configuration, enter the host and domain names of the firewall in the appropriate fields. If you use a mail relay, you must also enter the relay hostname.

Refer to the sendmail(1M) and configmail(1M) reference pages and *IRIX Admin: Networking and Mail* for additional information on configuring mail services on IRIX systems.

This section provides more detail on some of the information you are asked to provide on the Sendmail Configuration form. Each heading is a request for more information on the form, and is followed by additional material that should help you respond appropriately.

Enter the host name of your firewall:

The value in this field is the hostname of the interface to the external network. For example, in Figure 17-9, this is hostname assigned to the interface whose address is 192.132.122.12.

Enter the domain name of your firewall:

Values in the `host name` and `domain name` fields set particular *configmail* values. Used in conjunction with the *sendmail.cf.auto* file, *configmail* makes it possible to customize *sendmail* behavior without editing the *sendmail.cf* file. When you use *configmail*, *sendmail* is not used to accept email on the firewall; instead, a simpler, more secure, program called *smap* accepts and queues incoming email messages, and *sendmail* is periodically invoked to deliver messages in the queue.

Enter the hostname or alias of all relay hosts:

The entries in this field identify all hosts that transfer mail between a mail hub and other relays or between the hub and its final destination.

Subdomain names to be recognized for your site:

Gauntlet provides enhanced versions of the *configmail* utility and the *sendmail.cf.auto* file. Subdomains names specified on this page are used as follows:

- If no recognized subdomains are set, the Gauntlet firewall rewrites the sender's email address before delivery: if the address reads username@some_host.some_subdomain.DOMAIN_NAME, or just username@some_host.DOMAIN_NAME, it is rewritten to username@DOMAIN_NAME before the message is delivered.
- If recognized subdomains are set, the Gauntlet firewall rewrites username@some_host.some_subdomain.DOMAIN_NAME to username@some_subdomain.DOMAIN_NAME if some_subdomain is one of the recognized subdomains listed here; otherwise, it still rewrites the address to username@DOMAIN_NAME.

This behavior and the fact that the Silicon Graphics *sendmail.cf.auto* file tries to deliver email to username@sub_domain.DOMAIN_NAME to username@relay.sub_domain.DOMAIN_NAME if it can find a host named relay.sub_domain.DOMAIN_NAME, makes it easy to support multiple subdomains for large internal sites. See "Mail Hubs" on page 146 for more details.

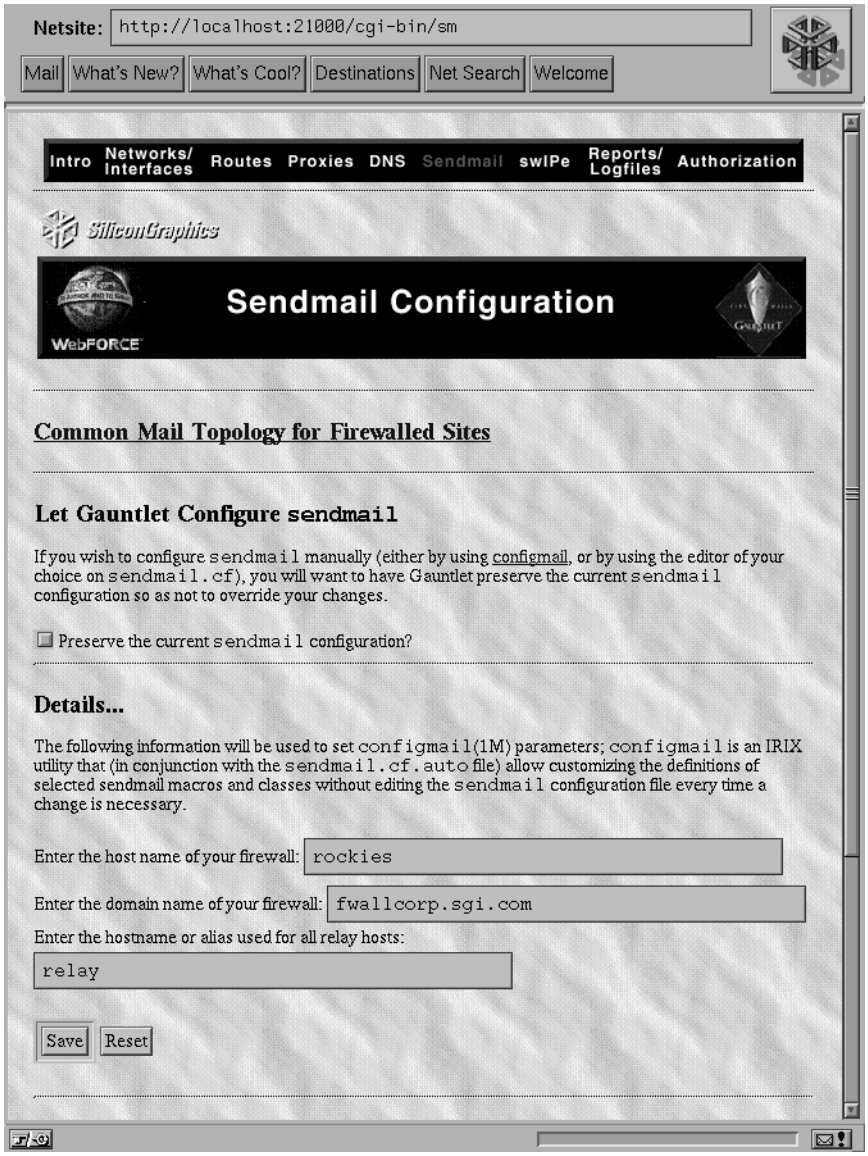

Figure 17-15 Sendmail Configuration Form

swIPe Configuration Form

The swIPe protocol creates a virtual private network (VPN) between two firewalls that are configured to support authentication and encryption between them. A VPN extends the security perimeter of the individual networks, each protected by a participating firewall, to encompass both networks. In such a configuration, the firewalls are considered *peers*. Both peers in the VPN must be running Gauntlet software. See Appendix C for detailed information on swIPe and VPNs.

Figure 17-16 illustrates two Gauntlet servers acting as peers in a VPN. Notice that in this figure the path connecting the peers is the Internet.

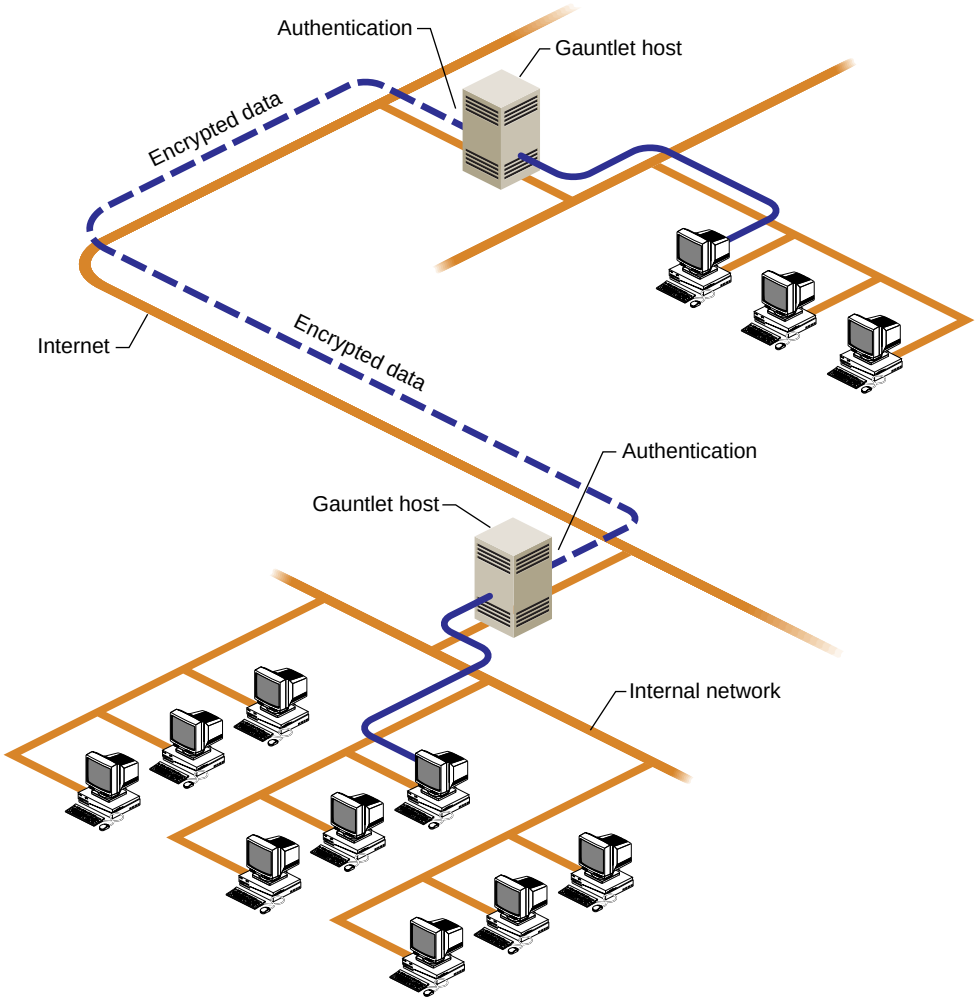

Figure 17-16 Gauntlet Hosts Using swIPe in a VPN

Authentication and Encryption Schemes

The swIPe protocol verifies that IP packets contain authentic source and destination addresses. This verification protects against IP address spoofing; it can be used in conjunction with permission sets to guarantee that interaction is occurring only between

confirmed entities. swIPe uses an hashing algorithm called MD5 to perform integrity checking and authentication. Both peers in a VPN must use the same authentication algorithm, which is identified by a key ID.

Whereas authentication protects against untrusted host interactions and data alterations, encryption protects against unauthorized access to the data. Encryption is a reliable way to protect data that crosses over untrusted networks and that must be kept secret and free from alteration. Peers participating in encryption schemes use a key to encode and decode data. Both peers must use the same encryption key, which is identified by a key ID.

VPN Paths

The path connecting two peers in a VPN may be one of three types: a *trusted path* carries encrypted data, and data is exchanged without user authentication; a *private path* carries encrypted data, but user authentication is required; a *passthrough path* forwards data freely to a destination that is not on the immediate VPN. A path is identified by the addresses of the peer servers that it connects. A key ID identifies the authentication algorithm and encryption key that are used to protect data on the path.

Preparing a Server for swIPe Configuration

Prepare for swIPe configuration by performing the following steps:

1. Ensure that your firewall is working as you would like before you add another network to create a VPN.
2. Determine whether you wish to use privacy with trust (no user authentication) or just privacy (user authentication required).
3. Determine whether you need to create any passthrough links on the firewalls that lie between your encrypted links.
4. Coordinate your efforts with the administrator of the remote network. Discuss your security policies and procedures. Prepare to synchronize the firewalls as you configure them.

You do not need to modify the IRIX configuration files on the firewall to support encrypted traffic. This is a standard service, included in the default versions of these configuration files on the Gauntlet Firewall.

Figure 17-17 illustrates the configuration form for swIPe.

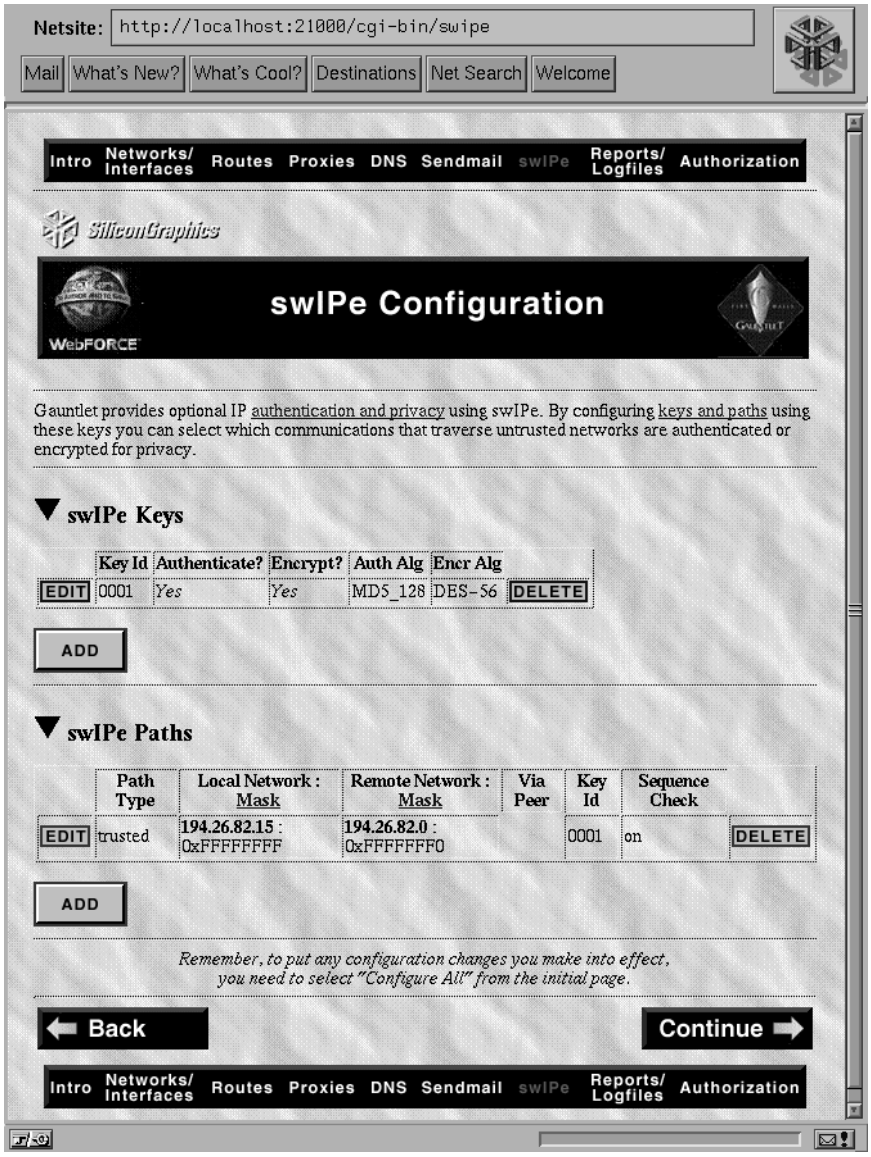

Figure 17-17 swIPe Configuration Form

Configuring a Server for swIPe

Perform the following procedure while the administrator of the remote network does the same at the remote peer site. It is important that both firewalls are configured at the same time because the encrypted packets must stay synchronized. The path does not work unless both ends have the same keys. Both firewalls discard any packets that unexpectedly arrive encrypted.

The swIPe configuration form (shown in Figure 17-17) consists of two parts: the top of the form contains authentication and encryption parameters; the bottom of the form identifies each path connecting the firewall to a peer. A separate entry form is used to provide the information for each part.

1. Select *ADD* to specify authentication and encryption key information.

   After your selection, the swIPe add key form is displayed:

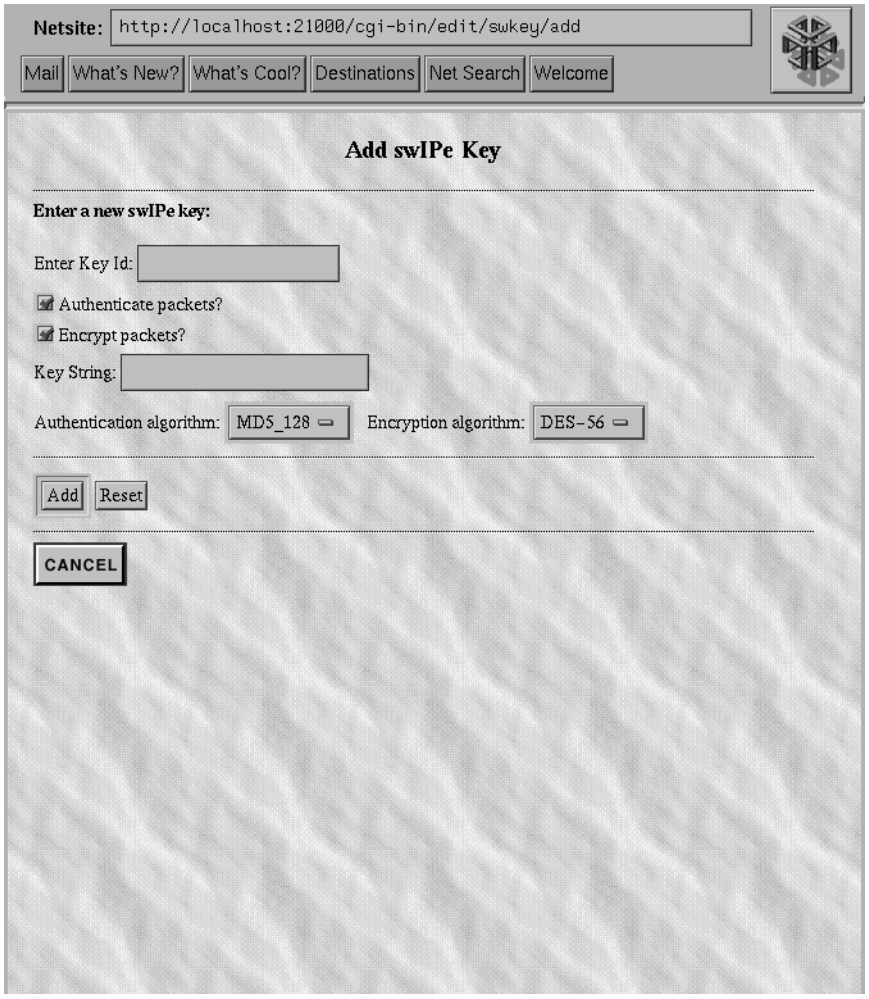

Figure 17-18 Add swIPe Key Form

2. Enter the key ID for authentication and encryption.

   To create a trusted or private link, you must specify the key you wish to use by its Key ID. Enter a number from 1 to 99.

   Click *Authenticate packets* and *Encrypt packets* to put either or both of these protection schemes into effect on this peer connection.

3. Type an alphanumeric entry to create a key string.
4. Select *Add* to configure the path between this peer and its remote counterpart.

   After your selection, the Add swIPe Path Form is displayed:

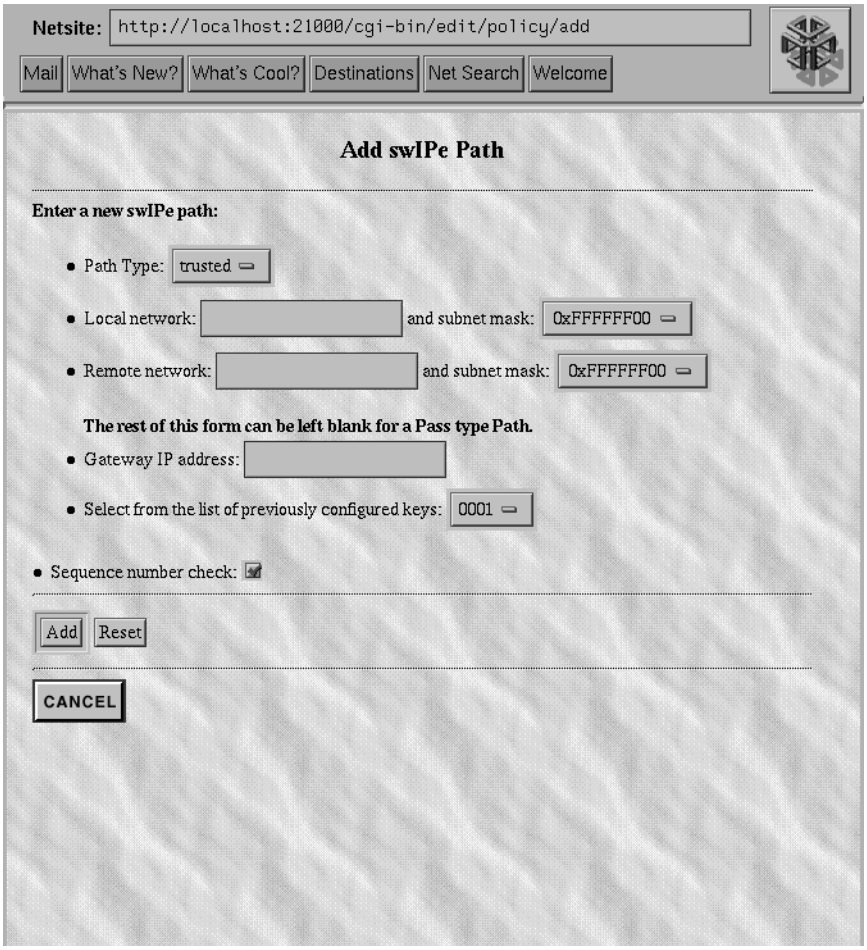

Figure 17-19 Add swIPe Path Form

5. Select the path type.
6. Enter the local and remote addresses of the peers in this VPN.

The default subnet mask automatically provided by the GUI for the local and remote networks is 0xFFFFFF00, which is the correct mask for a non-subnetted Class C network. If the mask is not correct for your configuration, click *Edit* and modify the mask field to change it.

7. For trusted and private paths, enter the gateway address.

   The gateway address is the address of the outside network interface. After the swipe driver encrypts the outgoing packets, the firewall sends them to this address to be forwarded to their destination outside the local network. This entry is unnecessary for passthrough paths.

8. Select the key ID for this path.

   Your selection should match the previous key ID entries.

9. Repeat the previous steps for additional networks that are behind the defense perimeter that you wish to add to your VPN.

10. Coordinate your configuration with the administrator of the remote network. Ensure that each firewall has the same encryption key for your VPN.

11. Reboot your firewall at the same time as the other administrator reboots the remote firewall.

Verifying Your Setup

If you are using a VPN with privacy and trust, issue the *ping* command to ensure that packets are flowing properly (*ping* uses ICMP packets and ICMP is built on IP.) Issue the *ping* command from a host within the local network (not the firewall) to a host within the remote network. For example, the Yoyodyne administrator in Maryland *ping*s the mail hub on the California office network.

Logfiles and Reports Configuration Form

Use the reports and logfiles form (Figure 17-20) to configure basic reporting mechanisms on the Gauntlet firewall.

The system automatically generates reports, and you can specify yourself and other users (in a comma-separated list) to receive these reports by e-mail. You can select the first three links on the Reports form at any time to view the most recently generated reports.

You can also specify which reports you want to receive, their frequency (daily, weekly, or both), how often you want the report software to run, and how long you want system log files to be saved. Save the files for at least seven days if you want to receive full weekly reports.

You should assign either yourself or another trusted user as the system Postmaster. This user receives any generic mail addressed to “Postmaster” at the Gauntlet host.

Example 17-2 contains an example of Gauntet log file entries (lines have been shortened for readability). If you do not want certain types of entries to be reported, you can specify them using *egrep* syntax in the field provided on this form (see the egrep(1) reference page). For example, enter “localhost” in the egrep field to keep lines that include the string “localhost” from appearing in the log file output. Be careful not to specify filters that are too broad; this might obscure warnings and notices that you want to see.

Example 17-2 Partial Log File Listing

```
Aug 10 02:00:08 6F:rfwall syslogd: restart
Aug 10 06:56:22 5D:rfwall netacl[1355]: permit host=boston.esd.abc.com...
Aug 10 06:56:22 5D:rfwall tn-gw[1355]: permit host=boston.esd.abc.com/...
Aug 10 06:56:32 5D:rfwall tn-gw[1355]: permit host=boston.esd.abc.com/...
Aug 10 06:56:32 5D:rfwall tn-gw[1355]: connected host=boston.esd.abc.c...
Aug 10 06:56:32 5D:rfwall netacl[1356]: permit host=localhost/127.0.0....
Aug 10 10:45:41 5D:rfwall authsrv[1893]: BADAUTH smith (tn-gw midas.xy...
Aug 10 10:45:45 5D:rfwall authsrv[1893]: BADAUTH exit (tn-gw midas.xyz...
<etc>
```

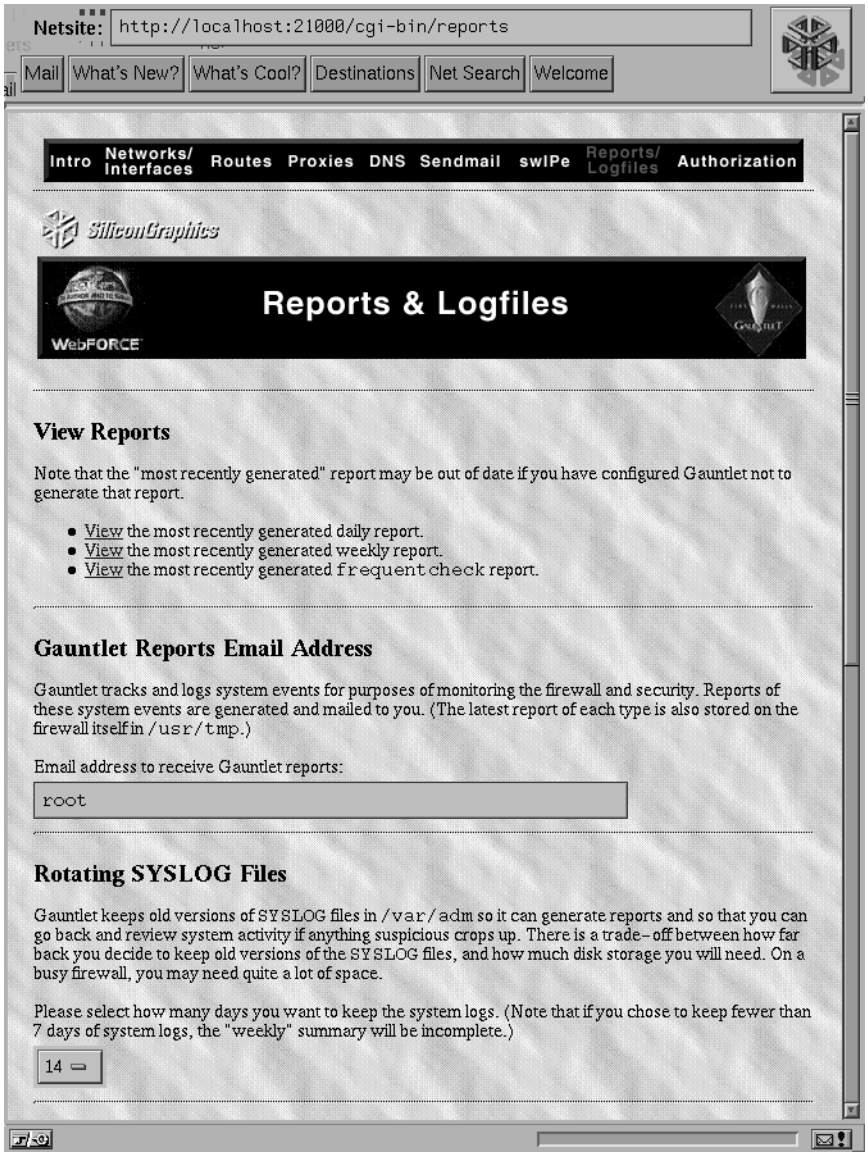

Figure 17-20 Reports and Logfiles Form (1 of 2)

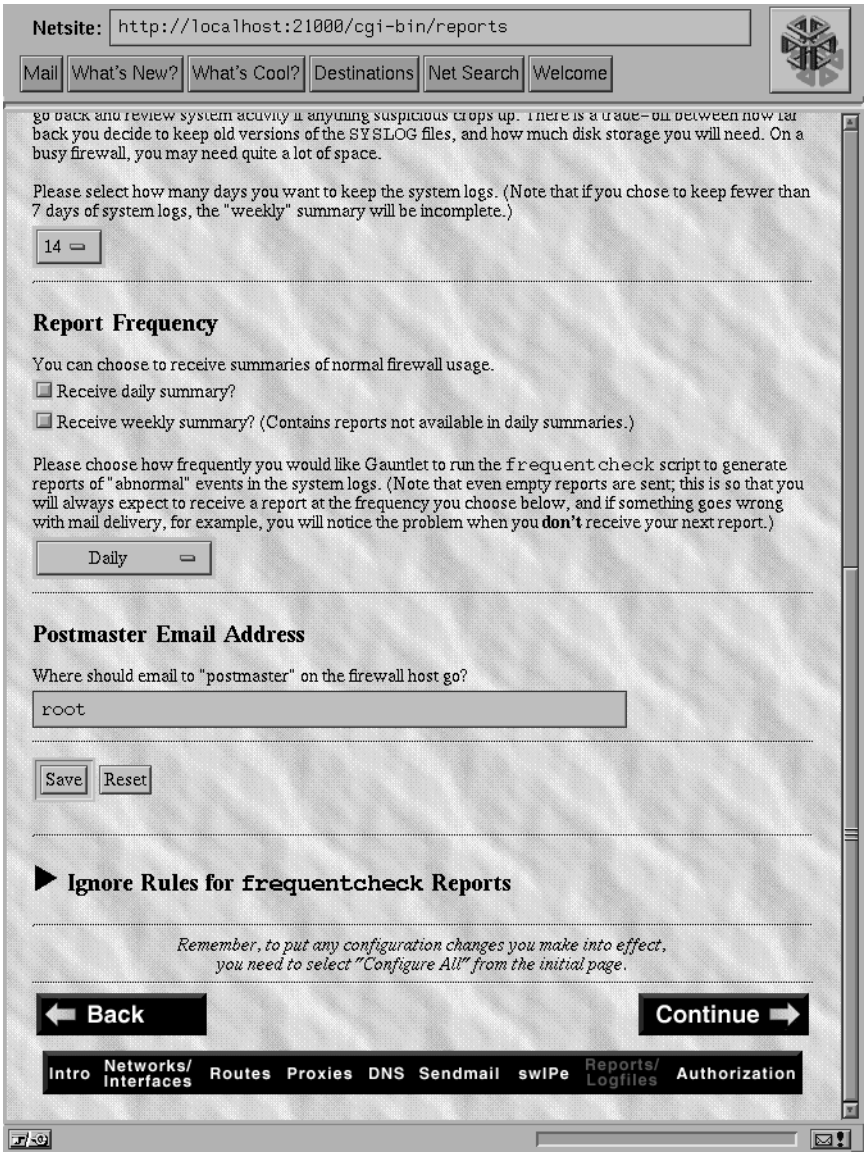

Figure 17-21 Reports and Logfiles Form (2 of 2)

Refer to Appendix A for command-line and file information on reports.

Authorizing Users Form

The authorizing users form (Figure 17-22) allows you to specify which users can access services from an untrusted network if they successfully authenticate themselves. Several different authentication mechanisms are supported.

User Authentication

You have several choices in setting a user's authentication protocol:

- skey—S/Key software is a free software authentication system from Bellcore that uses a challenge-response model to implement authentication. S/Key is included "as is" with the Gauntlet firewall. The IRIX executable that users need for generating responses is */usr/bin/key*; it can be copied to other IRIX 5.3 or later systems. If you want to use S/Key on other systems as well as IRIX, you can download source code from the site listed in "Additional Resources" on page xxiii. Refer to Example 17-3 for an example of an S/Key authentication session.
- EnigmaLogic SafeWord—support for the EnigmaLogic (Secure Computing) Safeword Authentication Server (see *http://www.safeword.com* for more information.)
- MDauth—a less widely known authentication system than S/Key, is based on MD5 checksums. MDauth is also a software-based system that uses challenge response. MDauth is included "as is" with the Gauntlet firewall. The IRIX executable that users need to generate responses is */usr/etc/softmd5*. S/Key might be preferable to MDauth, however, especially in heterogeneous environments.
- APOP— A system included with APOP-compliant applications, uses an MD5 secure hash algorithm. The application generates a random challenge and includes it as part of the initial banner. This option is currently only used by the POP3 proxy.
- Access Key II— a system from VASCO Data Security that uses a random challenge password. When the firewall prompts for authentication, it provides a challenge. The user enters his/her PIN (if one is required) and the challenge into the Access Key II. The Access Key II responds with a password. The user enters this value at the Gauntlet prompt, and the Gauntlet authentication server verifies this value.
- Security Dynamics ACE—support for the security Dynamics ACE Server (see *http://www.securid.com* for more information.)
- password—Plain text passwords. This is not recommended for use under any circumstances for accessing a network from over an untrusted network. Plain text passwords are included as an option principally for sites that wish to do chargeback accounting or individual accounting of firewall use.

When editing a user record, if the *Password:* field is not empty, the new value will be used to reset the user's existing password entry for whatever authentication protocol he or she uses (unless the protocol is for a third-party authentication server, in which case you should administer user passwords using the third party's administration tools.). If you make an error when editing a user record, click the *Reset* button to abort any changes that were made.

Adding a user with the Add Users form (Figure 17-23) means that the user can use all of the enabled services. The group field lets you associate groups of users.

Note: Adding users and groups here does not create IRIX accounts or groups for the users—just proxy server authorization.

Figure 17-24 illustrates user authentication on the Gauntlet host.

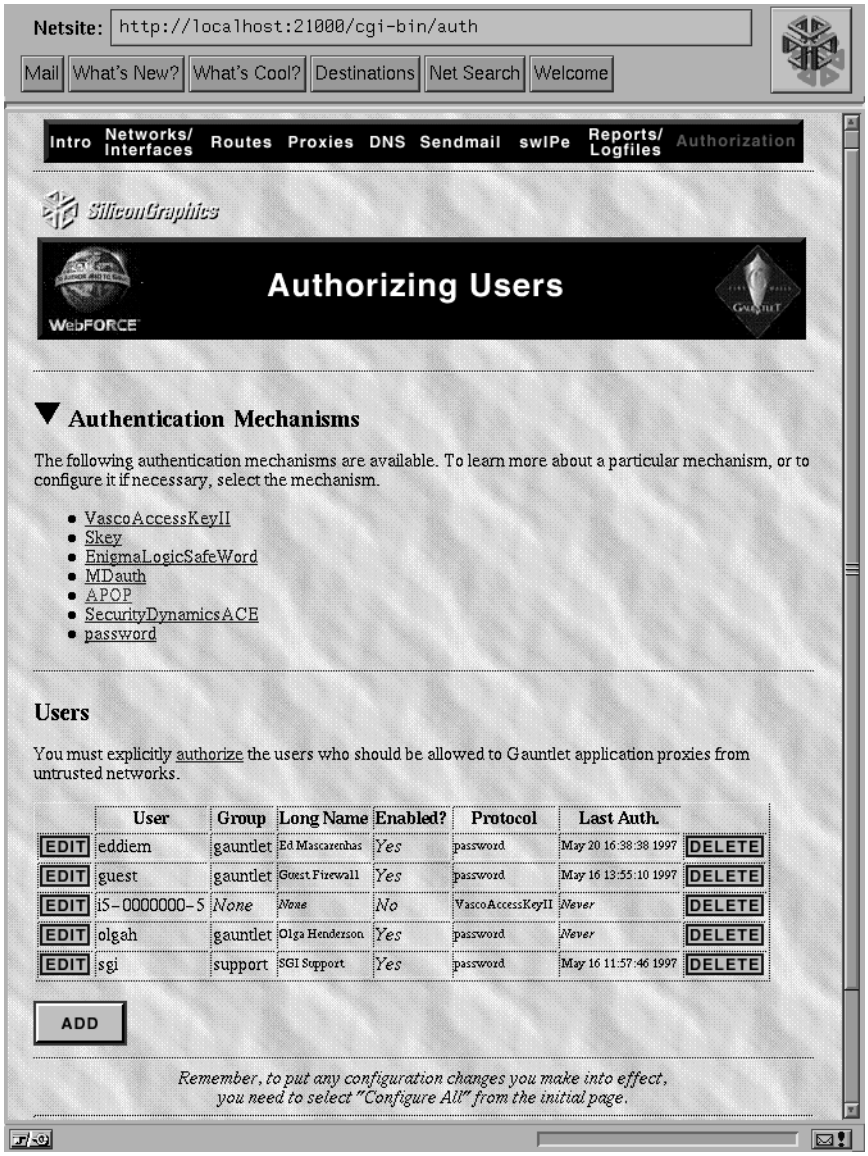

Figure 17-22 Authorizing Users Form

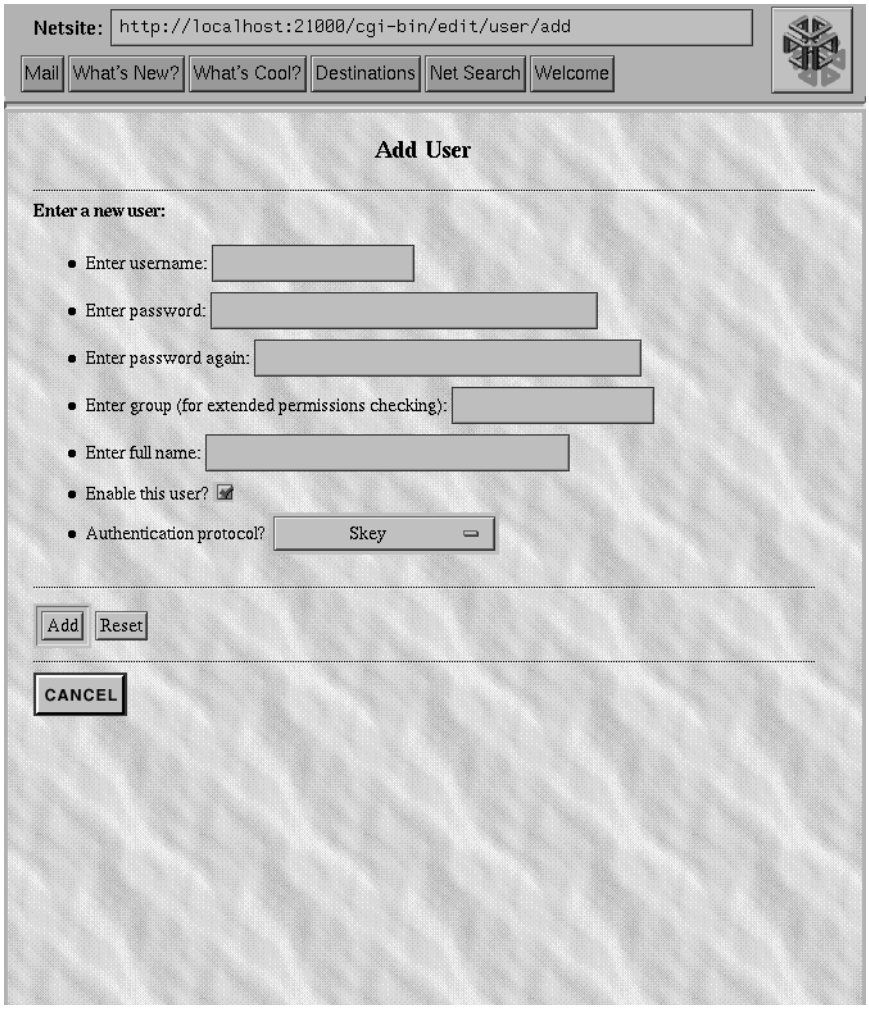

Figure 17-23 Add User Form

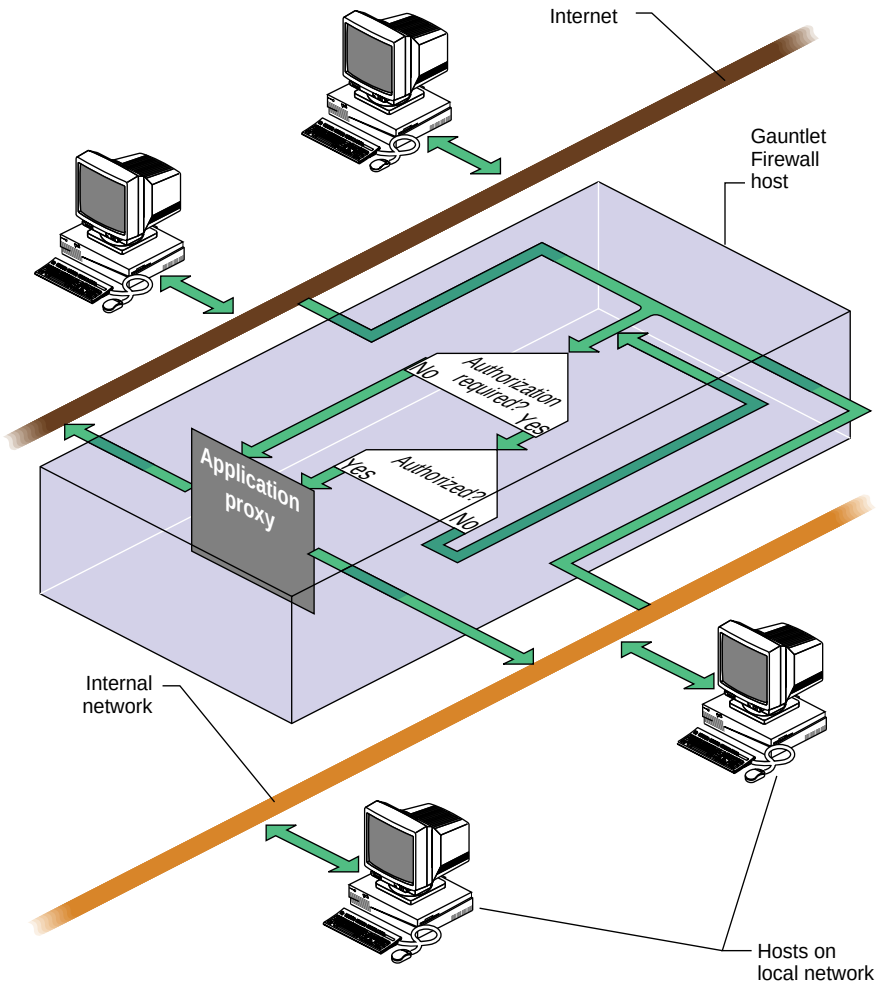

Figure 17-24 User Authentication

Example 17-3 shows an S/Key authentication session from the point of view of a user on a remote client. The figure assumes that the administrator of the system has already added the user in the authentication database as an S/Key user with a password that the user knows. It also assumes that the user has access to the */usr/bin/key* program on the client.

Example 17-3 S/Key Authentication Session Example

```
% telnet firewall.yoyodyne.com
Trying 204.255.154.100...
Connected to firewall.yoyodyne.com.
Escape character is '^]'.
Username: jones
Skey Challenge: s/key 662 rf20257:
```

At this point, the user must run the *key* program on the client to generate a response to the server challenge:

```
% key 662 rf20257
Enter secret password: fxdkiux


BUSY SWIM PIE GURU CAR DIG
```

The user then enters the response back at the server prompt:

```
Skey Challenge: s/key 662 rf20257: BUSY SWIM PIE GURU CAR DIGD

Login Accepted
tn-gw->
```

Caution: The user client must be secure. The user must be careful to always run the client locally so that his or her password is not sent over a network connection.

After a certain number of authentication sessions, a new password must be set for S/Key. The remaining number of authentication sessions for the current password is the first string in the S/Key server challenge (662 in the previous example).

Configuring Gauntlet for Remote Administration

To configure Gauntlet remotely, you must first run the management interface locally on the firewall to set the remote management option on. After this option is set, you can configure the firewall from a remote host or an X display at any time. By default, the remote administration proxy, *gadmin-gw*, supports one remote administration connection to configure Gauntlet on the firewall.

Note: Never administer your firewall from a remote host or X display on a network that is not trusted.

Use this procedure to configure the Gauntlet firewall for remote administration:

1. Working locally on the firewall, set the Gauntlet configuration flag on.

   This command sets the Gauntlet configuration flag on so that the Netscape server for Gauntlet administration starts on port 21000 whenever the system is booted:

   ```
   # chkconfig gauntlet-admin on
   ```

2. Access the Gauntlet administration interface and display the introductory management form.

   (See steps 1 through 3 of "Configuring Gauntlet Locally" on page 118 if you need instructions.) Find the Proxy Servers Configuration option on the introductory form (shown in Figure 17-4).

3. Select Proxy Servers Configuration from the introductory management form.

   After your selection, the Proxy Servers Configuration form is displayed (shown in Figure 17-10).

4. Click *Enable remote gauntlet administration proxy?* on the Proxy Servers Configuration form.

   The button to enable remote Gauntlet administration appears near the end of the Proxy Servers Configuration form (shown in Figure 17-12). To enable remote registration of the firewall, click this button.

5. Reset the port number and timeout value, if necessary.

   The default port for remote Gauntlet administration proxy is 21001; normally, it is not necessary to change this port assignment. However, if port 21001 is unavailable (because another service is using it, for example), you can assign a different port for the remote administration proxy by entering a new port number in the port number field. You can also enter a different timeout value if a different timeout interval is required.

6. Click *Configure All* on the introductory management form to put your changes in effect (see Figure 17-3).

Accessing the Administration Tool from a Browser

In subsequent administration sessions, you can use a Web browser on a remote host to administer the firewall. To access the remote administration server, do this:

1. Launch your Web browser from the remote host.
2. Set the HTTP proxy to access the remote administration proxy at port 21001.

   On the Netscape Manual Proxy Configuration page (shown in Figure 7-1), set the HTTP proxy to access the remote administration proxy at port 21001.
3. Point the browser to this URL:

   http://*firewall_hostname*:21000/cgi-bin/startup

   The introductory management form is displayed.
4. Reset the port number for the HTTP proxy.

   After the remote administration session, you should reset the remote administration port back to the setting used by the HTTP proxy: 8080. See Chapter 7, “Configuring Web Browsers” on page 56 for instructions.

Accessing the Administration Tool from an X Display

You can also use remote X display from a remote host to run the Gauntlet administration interface. To run the administration interface on a remote X display, do this:

1. Log in to the firewall from the remote host.
2. Set your DISPLAY environment variable to the host display.

   For example, `setenv DISPLAY` *remote_host*`:0`
3. Issue the *gauntlet-admin* command to display the introductory management form.

Configuring Gauntlet for Secure Remote Administration

If you wish to perform Gauntlet administration remotely over a secure connection, you must first configure a security scheme on the firewall for this type of connection. Secure Socket Layer software (SSL), which is built into the Netscape Web server, is the recommended protocol for implementing remote administration security. The SSL

protocol provides secure data communication between the remote host, the *gadmin-gw* administration proxy, and the configuration server running on the firewall.

Note: The *Netscape Commerce and Communications Servers Administrators Guide* provides complete instructions for configuring SSL on a server. In addition, a supplementary set of instructions is provided in Appendix D of this guide.

To configure secure remote management for the firewall, follow this procedure:

1. Working locally on the firewall, set the Gauntlet configuration flag on.

   This command sets the Gauntlet configuration flag on so that the firewall starts on port 21000 whenever the system is booted:

   ```
   # chkconfig gauntlet-admin on
   ```

2. Configure SSL on the firewall.

   See the *Netscape Commerce and Communications Servers Administrator's Guide* and Appendix D of this guide for instructions.

3. Launch your Web browser from the remote host.

4. Set the Security proxy to access the remote administration proxy at port 21001.

   On the Netscape Manual Proxy Configuration page (shown in Figure 7-1), set the Security proxy to access the remote administration proxy at port 21001.

5. Access the Gauntlet administration interface and display the introductory management form.

   (See steps 1 through 3 of "Configuring Gauntlet Locally" on page 118 if you need instructions.) Find the Proxy Servers Configuration option on the introductory form (shown in Figure 17-4).

6. Select Proxy Servers Configuration from the introductory management form.

7. After your selection, the Proxy Servers Configuration form is displayed (shown in Figure 17-10).

8. Click *Enable remote gauntlet administration proxy?* on the Proxy Servers Configuration form.

   The button to enable remote Gauntlet administration appears near the end of the Proxy Servers Configuration form (shown in Figure 17-12). To enable remote registration of the firewall, click this button.

9. Click *Configure All* on the introductory management form to put your changes in effect (see Figure 17-3).

10. Reset the port number and timeout value, if necessary.

    The default port for remote Gauntlet administration is 21001; normally, it is not necessary to change this port assignment. However, if port 21001 is unavailable (because another service is using it, for example), you can assign a different port for remote administration by entering a new port number in the port number field. You can also enter a different timeout value if a different timeout interval is required.

A server with security features on will require the key password to be entered when the server is started, which normally occurs at boot time. Once security is activated to access the Gauntlet administration server from your browser, use the URL: https://firewall:21000/cgi-bin/startup. If security features are not activated you can continue to use the server at the existing URL:http://firewall:21000/cgi-bin/startup. On the remote host the browser must be configured to connect to the proxy on port 21001 for remote administration. Note that for normal http access, the default proxy port is 8080.

Chapter 18

Managing User Authentication

As discussed in other chapters, the Gauntlet firewall can permit or deny access based not just on hostname, but also on user name. In addition, your security policy may require that users use some form of strong authentication each time they access a particular host or service within their perimeter. To ease the integration of users, strong authentication, and the firewall, the Gauntlet firewall provides a user authentication management system.

Use of the authentication management system is optional. However, you must use it any time you have configured your FTP, TELNET, and Rlogin proxies to require authentication from untrusted networks (the default for the Gauntlet firewall).

This chapter describes the concepts behind the user authentication management system and some common administrative tasks.

Understanding the User Authentication Management System

As part of the security policy, many sites may require some form of strong authentication, which requires users to enter a one-time password or use an authentication token. There are many systems available that can be integrated into a IRIX networking environment, each with its own programming and management interface. These are described in more detail in the section “Understanding Strong Authentication” below.

When you combine the user administration tasks for IRIX systems, a strong authentication system, and the Gauntlet firewall, you have a lot of interfaces to remember. The Gauntlet user authentication management system acts as a piece of “middleware” to provide a unified interface for several strong authentication systems and the Gauntlet firewall.

The Gauntlet user authentication management system allows you to easily integrate several different strong authentication systems into your general firewall administration. You can create, modify, disable, delete, and examine users. The authentication system maintains a database for this information.

How the Firewall Uses This Information

The various proxies use the information in the user authentication management system any time you have configured the proxies to require authentication. Using the default Gauntlet policies, this occurs any time a user from an untrusted network tries to access a service inside the perimeter. Recall that untrusted networks are those from which the firewall accepts requests only after authentication by the user.

Remember that using the default policy, the proxies do *not* authenticate requests from trusted networks. The proxies operate under the assumption that users coming from trusted networks are who they say they are.

For example, consider the situation of a user, John, working at a client site (blaze.clientsite.com) who needs information stored on a machine at work (dimension.yoyodyne.com). When John tries to TELNET to dimension, which is within the perimeter, he must pass the first authentication at the firewall (firewall.yoyodyne.com).

When firewall.yoyodyne.com receives the information, the TELNET proxy determines that the connection request is from an untrusted network, and that John can access inside machines.

The TELNET proxy then prompts John for his authentication information (user name and challenge), which it verifies against the information in the user authentication database. If John provided the proper information, and his account is not disabled, the proxy provides a prompt. John can then connect to dimension on the inside network.

How Other Services Use This Information

The *login-sh* program uses the user authentication database to authenticate users logging into the firewall itself. This login shell authenticates the user before starting the users normal shell (for example, *csh*, *ksh*, or *zsh*). See login-sh(1M) for details.

Users

User names you create in the user authentication management system are used only for strong authentication. The user names must match the user names for the strong authentication system you are using.

The user names in the user authentication management system do not generally need to match user names on the firewall itself. By default, you do not create any user accounts on the firewall. The exception to this rule is the *login-sh* authentication wrapper program. The *login-sh* program authenticates users before logging them into the firewall. Then, the information in the user authentication management system must match the standard IRIX user information (in */etc/passwd*) for these users.

The user names in the user authentication management system do not need to match any user names on your internal network. For example, John Whorfin might use “john” as his user name on internal networks. He could use “whorfin” for strong authentication at the firewall. You may wish to use the same names for the convenience of your users.

Groups

The Gauntlet user authentication management system also makes use of groups. Groups allow you to permit or deny services based on groups of user names, rather than individual user names. For example, you can configure the X11 proxy to permit service to everyone in group sales.

Just as is the case with user names, the groups that you create in the Gauntlet user authentication management system are not the same as the groups you create on the firewall or on the internal network. You can of course use the same names, for easier administration.

The Pieces

The user authentication management system consists of several programs. The use of each of these components, and their options, is described in the appropriate sections of this chapter.

Authentication Server

This program (*authsrv*) is a network daemon that actually verifies information against the database. It also allows firewall administrators to modify user and group information, and disables user accounts automatically after a configurable number of failed login attempts

Authentication Editor

You can edit, add, or delete users from the *gauntlet-admin* Authentication page.

Authentication Loader

This program (*authload*) loads records in bulk into the database, and initializes the database.

Authentication Dumper

This program (*authdump*) exports the contents of the database to an ASCII file, for easy backup.

Understanding Strong Authentication

The Gauntlet firewall supports a variety of strong authentication options. The authentication management system understands the types of passwords that these systems use, and provides a consistent user interface to these systems.

Currently supported systems are shown below. Consult the system requirements card in your Gauntlet firewall package for the latest information on supported versions of the these products.

Access Key II

This system, from VASCO Data Security, uses a random challenge password. When the firewall prompts for authentication it provides a challenge. The user enters their PIN (if one is required) and the challenge into the Access Key II. The Access Key II responds with a password. The user enters this value at the Gauntlet prompt, and the Gauntlet authentication server verifies this value.

APOP

This system, supported APOP compliant (mail) applications, uses an MD5 secure hash algorithm. The application generates a random challenge and includes it as part of the initial banner.

This option is currently used only by the POP3 proxy.

SecurID

This system, available from Security Dynamics, uses a time-based password. The SecurID card generates a passcode. When the firewall prompts for authentication, the user enters his or her personal identification number (PIN) and the passcode shown on the card. The Gauntlet authentication server verifies this value with the Security Dynamics ACE server.

EnigmaLogic SafeWord

SafeWord is available from EnigmaLogic and supports numerous hardware authentication tokens. The Gauntlet authentication server communicates with SafeWord during the authentication process.

S/Key

This system, from Bellcore, uses a one-time password. Users generate a set of passwords based on a “seed” word or phrase. Each time they need to authenticate, they use a different password. When the firewall prompts for authentication, it provides a challenge value. The user enters his or her appropriate password for that challenge. The Gauntlet authentication server verifies this value.

The Gauntlet firewall distribution includes a portion of the S/Key package. The full S/Key package is available for FTP from ftp.bellcore.com in *pub/nmh/skey.*

You can also use the Naval Research Lab One-Time Password in Everything (OPIE), which is downward-compatible with Bellcore's S/Key Version 1 software. The OPIE package is available for FTP from ftp.nrl.navy.mil in */pub/security/nrl-opie/*.

Reusable Passwords

This system, a part of the user authentication system included with the Gauntlet firewall, is a reusable password option. It is designed for administrator testing only. Every time

users need to authenticate, they use the same password. Reusable passwords are also sometimes known as plain or text passwords.

Warning: ***Do not*** **use the reusable passwords option for authentication from untrusted networks. We discourage the use of reusable passwords. Reusable passwords are vulnerable to password sniffers and are easy to crack. This feature is provided for convenience and audit capability only.**

Configuring the User Authentication Management System

Configuring the user authentication management system involves planning, setting up the third-party authentication system, configuring network systems, configuring firewall services, initializing the database, and verifying that you did it all correctly.

Unless otherwise noted, you must perform all of these tasks from the firewall console as root.

Once you have configured and are using the system, all activity to the authentication database is logged and included in the weekly summary reports.

Configuring Third Party Systems

See the online configuration help available for the third-party systems by clicking on the authentication system name on the *gauntlet-admin* Authentication page.

Note: File locations may verily from those specify in these procedures. To

Safeword Authentication Server

You must create or modify one file on the firewall so it knows where the Safeword Authentication Server is (typically on a machine other than the firewall).

To configure your firewall for use with a Safeword Authentication Server:

1. Create user accounts for your users using the Safeword Authentication Server.
2. Edit the Safeword configuration file, `/usr/local/etc/swec.cfg`. Set the Safeword Authen. Server Name to the name of the machine on which the Sfaeword Authentication Server is running:

```
02 SafeWord Authen. Server Name:        localhost 0 0 7482
```

3. Kill and restart the authentication server so that your change takes effect.

SecurID

Because your ACE/Server is running on a machine other than the firewall (which is the default and recommended configuration), you must install one file on the firewall itself and modify the Gauntlet configuration information.

To configure your firewall for use with an ACE/Server:

1. Create user accounts for your users using the ACE/Server.
2. Ensure that the service name and port number specified for the SecurID service in `/etc/services` on the machine running the ACE/Server is also specified in `/etc/services` on the firewall.
3. Register the firewall as a client system on your ACE/Server. Be sure to use the IP address or host name for the inside address of the firewall if your ACE/Server is running on a machine on your inside network.
4. Copy the file `/var/ace/sdconf.rec` to the firewall as `/var/ace/sdconf.rec`. This file contains information that tells the authentication server where to find the ACE/Server.
5. Modify `/usr/local/etc/netperm-table` and add information about the ACE/Server:

   `authsrv:securidhost` *`firewall`*

   - where *firewall* is the host name of the Gauntlet firewall that you registered as the client host name on the ACE/Server.

     ```
     authsrv:securidhost fire-in.yoyodyne.com
     ```

Configuring Network Services

You do not need to modify the IRIX configuration files on the firewall to support the authentication management system.

You may wish to modify */etc/passwd* to force those with actual user login accounts on the firewall itself (these accounts should only be the administrators for the firewall) to

strongly authenticate themselves to login, using login-sh(1M), which supports any of the authentication mechanisms aforementioned.

Configuring Authentication Management System Rules

If you are using the Gauntlet Firewall default configuration, you do not need to modify the configuration rules for the user authentication management system. If you have chosen a different port or a different location for your database, you must modify */usr/gauntlet/config/template.netperm-table* to reflect your configuration. See Appendix B for more information on authentication manager options, *netperm-table* options, and order of precedence.

Verifying Your Installation

Verify your installation by accessing the firewall from a host on the outside network.

To verify an installation using TELNET:

1. On a host on the outside network, TELNET to the firewall.
2. At the TELNET proxy user name prompt, enter a user name you have created.
3. At the TELNET proxy password prompt, enter the appropriate password or response for the user you have created.
4. When you see the “Login Accepted” banner, you have verified your installation. You are now ready to begin creating groups, adding users and assigning them to groups. These tasks are described in the sections on managing groups and users in this chapter.

Managing Groups

As with IRIX systems, the Gauntlet user authentication management system makes use of groups. Groups allow you to permit or deny services based on groups, rather than individual user names. For example, you can configure the TELNET proxy to require authentication for everyone in the group “sales”. See “Extended Permissions” in the authserv(1M) reference page for details.

Remember that the groups that you create in the Gauntlet system are not necessarily the same as the IRIX groups you create on the firewall or on your internal network. You can of course use the same names, for easier administration.

Creating Groups

Groups can be created with the authorization server or the *gauntlet-admin* interface.

To create a group, assign a user to a group that did not exist before. Remember that you may want to make your group names the same as existing IRIX groups.

Disabling Groups

You cannot disable entire groups. You must disable usage based on individual users.

Deleting Groups

To delete a group, you must reassign all users in that group to another group, or to no group at all.

Managing Users

Creating Users

Users can be created with the *gauntlet-admin* interface.

If you need to create a large number of users, use the authentication loader. The authentication database dumper will show you the format that the loader expects.

Remember that the users that you create in the Gauntlet system are not necessarily the same as the users you create on the firewall or on your internal network. You can of course use the same names, for easier administration.

To create a user, follow these steps:

1. Enter the user ID and full user name.
2. If you want to assign the user to a group, enter the name of the group. If the group does not exist, creating the user creates the group. If a group administrator creates users, those users will inherit the group information.
3. Enter the strong-authentication protocol for this user. Current options are selectable from the selection box.

Warning: ***Do not*** **use the reusable passwords option for authentication from untrusted networks. Reusable passwords are vulnerable to password sniffers and are easy to crack. This feature is provided for convenience and audit capability only.**

4. Enter the authentication information for the user if applicable. (In some cases, this will be a password.)
5. Verify the authentication information by entering it again.
6. Make the information active by saving these changes (in *gauntlet-admin*).

Creating Default Users

Creating a default user allows you to authenticate users without manually creating entries for every user in the Gauntlet authentication database. Note that this option is only available for:

- Safeword Authentication Server
- SecurID

You can only have one default user. When a user logs in and the authentication server does not find the information in the Gauntlet authentication database, the authentication server sends the user information to the remote authentication server. The authentication server also creates a record for that user in the Gauntlet authentication database.

To create a default user:

1. Enter **default** as the userid.
2. Enter the name of a group to assign the user to a group, if desired. If the group does not exist, creating the user creates the group. If a group administrator creates users, those users will inherit the group information.

3. Enter the strong-authentication protocol for this user. Valid options are:

protocol	value
Safeword Authentication Server	Safeword
SecurID	SecurID

4. Leave the password for this user empty. The authentication server uses the value registered with the appropriate server.
5. Make the information active by saving these changes (in the *gauntlet-admin* and *gauntlet-gui* menuing systems) or exiting the authentication server.

Creating Users with Access Key II

To create a user with Access Key II authentication, use this procedure:

1. Create a key for the Access Key II according to the documentation included with the key. This creates the keyfile.log file which contains the key.
2. Log in to the firewall and become root.
3. Copy this file to a temporary directory (such as*/tmp/vasco*) on the firewall.
4. Load the key information into the user authentication management system using the key initialization tool (*/usr/etc/vasco_init*)

   ```
   firebird# vasco_init /tmp/vasco/keyfile.long
   ```

   This tool creates a user in the authentication management system and loads the key for this user. It creates the user name by prepending the letter i to the serial number for that Access Key II. This user is initially disabled.

   If you are using multiple authentication servers, or are running your authentication server under a different name, consult the vasco_init(8) reference page for additional command line options.
5. Note the user name that the initialization program displays so that you can change it to something easier for the user to remember.

   ```
   record loaded for user: i2-0005899-4
   ```
6. Use and authentication management tool to change the name of the user:

   ```
   authmgr->rename i2-0005899-for john 'John Whorfin'
   ```

7. Enable the new user:

   ```
   authmgr->enable john
   ```

8. Make the information active by saving these changes or exiting the authentication server.
9. Provide users with their Access Key II and user name.

Changing User Names

You cannot actually change a user name using the authentication management system. You must create a new user name, assign appropriate groups and privileges, and delete the old user name.

You can, however, change the long name information for a user using the *gauntlet-admin* interface

To change the long name information, follow these steps:

1. Select the record for the user name you wish to modify.
2. Tab to the name field and change the information.
3. Make these changes active by saving these changes.

Changing Groups

Users can only belong to one group at a time.

To change groups, follow these steps:

1. Select the record for the user name you wish to modify.
2. Enter the name of the new group in the group field.
3. Make the information active by saving these changes.

Changing Protocols

To change protocols, follow these steps:

1. Configure the user information in the third party authentication system if you want the user to use that system.
2. Select the record for the user name you wish to modify.
3. Enter the new protocol in the protocol field.
4. Enter the new password in the password field if applicable.
5. Make the information active by saving these changes.

Changing Passwords

Several strong authentication systems (for example, MDauth, S/Key, and reusable passwords) allow passwords that can be set (and reset) by the user.

For other authentication systems, you must use the third party authentication server tools to allow a user to change passwords or change something equivalent, such as a PIN for a hardware token device, or to change devices altogether.

Allowing Users to Change Their Password

Because users are generally not allowed to log directly into the firewall, they must change their password from another machine. The default policy allows users connecting to the firewall from the inside network to change their passwords for non-third party systems.

Users can change their passwords through either the TELNET or Rlogin proxies.

To change passwords as a user, follow these steps:

1. From a machine on the inside network, TELNET or Rlogin to the firewall.
2. Use the password command.
3. Authenticate to the proxy.
4. Enter the new password.
5. Verify the new password.

The example below shows a sample S/Key password change from the TELNET proxy:

```
dimension-83: telnet firewall
Trying...
Connected to firewall.yoyodyne.com
Escape character is '^'.
tn-gw-> password
Changing passwords

Username: john
Skey Challenge: s/key 644 fi58297 LOAM WOOD BOIL VASE TELL TINY
New Password: ###############
Retype New Password: ###############
ID john s/key is 664 fi582901
```

Enabling Users

Enabling users also allows users who have been disabled to use the system again.

To enable a user, follow these steps:

1. Select the record for the user name you wish to modify.
2. Check the "Enable" box.
3. Save your changes.

Disabling Users

Disabling users allows you to keep the user information in the system, but does not allow the user to use the system. The user authentication system disables users after a set number (configurable by the administrator) of failed login attempts. You can disable a user by unchecking the Enable box.

Deleting Users

Deleting users removes them from the user authentication management system. It does not remove users from your firewall or from your internal network.

To delete a user, follow these steps:

1. Select the delete option for the record for the user name you wish to delete.
2. Confirm your deletion action.

Chapter 19

Using the Login Shell

You need to log into the firewall occasionally to manage it. You may log in directly at the console or remotely via TELNET or Rlogin. Occasionally, you may also need to FTP files to or from the firewall. Whenever you access the firewall remotely, you're sending your password (and your root password) in the clear across your internal network to the firewall. While you'd like to believe that this is secure, you want to be prudent.

One way of doing this is to login to the firewall using some form of strong authentication that uses one-time passwords or time-based responses. The login shell program included with the Gauntlet firewall allows you to use the same strong authentication scheme for logging into the firewall itself as you do for activity between opposite sides of your security perimeter.

This section explains the concepts behind the login shell program and how it works, how to configure the program, and how to use it.

Understanding the Login Shell Program

The login shell program is a wrapper program that authenticates the user (using strong authentication) before passing control to the real login shell. It provides authentication and logging.

How It Works

A user logs into the firewall via the console, TELNET or Rlogin. This calls the standard login program (*/bin/login*) to process the login. The login program asks for a user name. The login program reads the */etc/passwd* file and determines that this user does not require a password (because the password field is empty). It then passes the information to the program specified in the shell field, the login shell program (*/usr/etc/login-sh*).

The login shell program prompts the user for the appropriate response for their authentication method (S/Key password, etc.) The login shell program checks its configuration information (in the *netperm-table*). It authenticates the user using the authentication server specified in the *netperm-table*. The login shell program then reads the shell file configuration file (which must be specified in the netperm table. Normally this file is in */usr/etc/login-shellfile*). It passes the login information on to the executable specified for the user in the shell configuration file, which is normally the user's shell. The user is now logged into the firewall and ready to work.

Note that the standard FTP daemon does not use */bin/login*, so will not invoke the login shell program for authentication. This is not generally a problem, as running the standard FTP daemon on the firewall is strongly discouraged.

Configuring the Firewall to use the Login Shell Program

Configuring the Gauntlet firewall involves planning, enabling remote login, creating user accounts, configuring the proxy to enforce your policy, and securing other applications.

Planning

Determine your policies for who you allow to access the firewall remotely

Enabling Remote Login

You must configure the firewall to allow remote login from other hosts.

To enable remote login, use the gauntlet administration tools to turn on remote login to allow the firewall to run the TELNET daemon.

Adding Support for the Login Shell

You must add support for the login shell so that the operating system recognizes the login shell as a valid shell.

To add support for the login shell, edit */etc/shells* and add a line indicating the location and path of the login shell

```
/usr/etc/login-sh
```

Creating User Accounts

To create user accounts:

1. Create or modify the user account on the firewall. Use *vipw* or another account creation program available for your operating system.

   ```
   scooter::518:10:Scooter Lindley:/home/scooter:/usr/etc/login-sh
   ```

   - Leave the password empty, because the login shell uses your strong authentication information
   - Note that if you include a password, you are prompted to authenticate twice: once for the information you enter here, and once for your strong authentication information.

2. Specify *login-sh* as the shell
3. Create the user's home directory, if necessary:

   ```
   mkdir /home/scooter
   ```

4. Add the user to group wheel so that they can *su* to **root**. Use *vi* to edit */etc/groups*.

Configuring the Proxy Rules

If you are running the Gauntlet firewall default configuration, you do not need to modify configuration rules for the login shell. If you have chosen a different authentication server or a different location for your shell file information, you must modify */usr/gauntlet/config/netperm-table* to reflect your configuration. See Appendix B for more information on *login-sh* options, *netperm-table* options and order of precedence.

Configuring the Shell

You must provide information for each user indicating their final (real) shell. After the login shell authenticates the user, it starts the user's final shell.

To configure the shell:

- Edit the shellfile file (*/usr/etc/login-shellfile*) and add information about the final shell for that user:

  username executable parameters

  where

username	same user name that you specified when you created the user's account on the firewall.
executable	name of the executable that the login program executes after authenticating the user. This is typically the user's shell
parameters	parameters to the executable program. The first parameter is typically a dash (-) and the shell name (csh, ksh, etc.)

For example:

```
scooter   /usr/bin/tcsh   -tcsh
```

Creating User Authentication Records

To create the record in the authentication database:

- Use the authentication management system to create authentication user entries for all users who will use the login shell on the firewall. Use the same user name that you specified when you created the user on the firewall. Consult Chapter 18, “Managing User Authentication,” for more information on creating users in the authentication management system.

Securing Other Applications

To secure other applications:

1. Disable programs (such as *chsh*) that allow users to change their shells. Either remove the executable or change the file permissions to 700

   ```
   chmod 700 chsh
   ```

   - Note that you should only create accounts on the firewall for people who need to administer the firewall. They will all generally have access to the root password. Changing file permissions will not prevent them from changing their shell. If you are creating accounts for other users on the firewall (which is

not recommended), changing file permissions will prevent them from changing their shell.

2. Verify that the *su* command is not aliased to *'su -m'* in your account (*.cshrc*, *.login*, etc.) on the firewall. The *-m* option attempts to retain the current environment. This causes your login shell (in this case, *login-sh*) to be executed by user **root**. Because there is no entry for **root** in the *login-shellfile*, *'su -m'* does not work.

Verifying Your Setup

Verify your installation by TELNETing to the firewall and connecting to the firewall itself. Connect to the firewall directly:

```
tn-gw-> c localhost
```

Note that after you enter your user name, you are prompted for your strong authentication information.

Using the Login Shell Program

Accessing the Firewall from Trusted Networks

Login to the firewall (via the console, TELNET or Rlogin) as you did before. Note that after you enter your user name, you are prompted for the response or password specified for your authentication scheme. Become root (via su) to do work as needed.

Accessing the Firewall from Untrusted Networks

Connect to the firewall as you did before, providing your strong authentication information to connect to the proxy. If you login to the firewall itself, you will need to authenticate again.

Note that we do not recommend logging onto the firewall from an untrusted network if you need to do work as root. When you su to root, you are sending your password in the clear across the untrusted network.

Changing Password for User Account

When you are using the login shell, the password is actually the strong authentication password, not the standard UNIX password.

Do not use the *passwd* or *chpass* programs on your UNIX system. To change your password, you must follow the instructions for changing your strong authentication information as described on page 135.

If you use the *passwd* or *chpass* programs, you will create a UNIX password. You will then need to provide both your UNIX password and your strong authentication information when you login to the firewall.

Chapter 20

Logging and Reporting

Logging is an important part of a properly configured firewall. Administrators can use the information in logs to gather usage statistics, monitor activities, check for problems, and investigate potential attacks. The logging features of the Gauntlet Internet Firewall provide administrators with a wealth of information about activities to and through the firewall. The logging features present the information in several formats. You should, of course, configure both the logging and reporting features to match your security policy.

This chapter describes the concepts behind logging and reporting systems, configuring these systems, and understanding the log and report formats.

Understanding Logging and Reporting

The Gauntlet Firewall follows the philosophy that it is easy to compress, consolidate, summarize, and delete log information; it is impossible to retroactively gather log information on an event that has already occurred. Disk space is a lot cheaper than spending many hours debugging a problem that a program would have written to the logs. For these reasons, the components of the Gauntlet Firewall log a wide variety of activities and attributes.

These are the components of the Gauntlet Firewall:

- firewall kernel
- proxies
- authentication management system
- DNS
- sendmail

These are the attributes logged:

- source IP address
- destination IP address
- source port
- destination port
- user name
- session date and time
- number of bytes transferred
- individual commands (for some activities)
- successful access attempts
- unsuccessful access attempts

Creating Logs

The proxies, kernel and authentication management system automatically write information to the logs. These programs call the standard IRIX system log command (*syslog*) to write information to the standard IRIX log file in */var/adm/SYSLOG*. You don't need to do anything special to create the logs. Even if you choose not to do anything with the information in the logs, the programs still write the information. You never know when you might need it.

The message log file also contains information from other programs, such as *bind*, *cron* and other IRIX utilities that use the *syslog* command

As with any other information that the *syslog* function writes, the firewall log information is ASCII text. People and shell scripts can easily parse the information.

Every night the *cron* daemon runs a shell script that rotates, compresses, truncates, and removes the log files. The Gauntlet script */usr/gauntlet/bin/daily* rotates the reports and compresses (using *gzip*) older log files.

Configuring Logs

The default logging options included with the Gauntlet Firewall meet the needs of most security policies. You do not need to set or modify any options if you wish to use the default configuration, which logs all of the information described above, and retains the logs for 14 days. You can customize the contents and retention of the log, however.

Configuring Additional Logging

Many of the proxies can log specific commands. For example, the FTP proxy can create a log entry for each command (STOR, RETR, CWD, LIST) it receives.

To modify the commands that the proxies log, add the *-log* parameter and appropriate options for the proxy in the *netperm-table* file. Consult Appendix B for more information on editing the *netperm-table* file and proxy-specific logging options.

Configuring Log Retention Time

If you wish to change the length of time the firewall retains log files, you may do so with the *gauntlet-admin* interface.

To set the retention time, set the number of days to retain the logs.

Creating Reports

The Gauntlet Internet Firewall contains several reporting mechanisms that sort through the log files and summarize the information. The firewall automatically generates the reports that are selected in *gauntlet-admin*. The *cron* daemon is used to run a set of shell scripts that parse the information in */var/adm/SYSLOG*. You do not need to do anything special to create the reports; the firewall does it automatically.

The firewall includes two main types of reports: Service Summary Reports and Exception Reports.

Service Summary Reports

The Service Summary Reports include usage and user information on a per service basis. For example, the default report for the TELNET gateway indicates the top 100 clients by connections, the top 100 clients by amount of traffic, and the top 100 denied clients.

Each night the *cron* daemon on the firewall runs the daily script (*/usr/gauntlet/bin/daily*). When the daily report option is turned on (it is on by default), this script calls a daily report script (*/usr/gauntlet/bin/daily-report*) which calls other shell scripts to summarize the logs for each service. The firewall mails the reports to the firewalladmin alias as configured with *gauntlet-admin*. Note that the firewall stores the daily report in */usr/tmp/daily-report*.

When the weekly report is turned on, the *cron* daemon on the firewall runs the weekly script (*/usr/gauntlet/bin/weekly*). This script calls the weekly reporting script (*/usr/gauntlet/bin/weekly-report*) to summarize the services for the past week. The firewall mails the reports to the firewalladmin alias. Note that the firewall stores the weekly report in */usr/tmp/weekly-report*.

Exception Reports

Exception Reports include noteworthy items. The Gauntlet Firewall defines a list of items that are not noteworthy and ignores those sorts of entries in the logs. The firewall considers all other events as possible security events. Thus, any item that you have not specifically told the firewall to ignore, it reports. This report includes information that could indicate a possible attack or other problems.

For example, the firewall default is to ignore successful authentications when parsing the log file. Successful authentication attempts are a normal part of firewall activity. However, unsuccessful authentication attempts could be a sign of a potential attack. Therefore, the exception report includes all unsuccessful authentication attempts from the logs.

To create the Exception Reports, the *cron* daemon periodically (the default is four times a day but this can be configured in *gauntlet-admin*) runs a reporting script (*/usr/gauntlet/bin/frequentcheck*). This script scans the log files for events that it can ignore, as defined in another configuration file (*/usr/gauntlet/config/frequentcheck.ignore*). The script summarizes all of the noteworthy items since the last time it created a report. The firewall mails the reports to the firewalladmin alias. The firewall stores the exception report in */usr/tmp/frequentcheck-report*.

Configuring Reports

The default reporting options included with the Gauntlet Firewall meet the needs of most security policies. You do not need to set or modify any options if you wish to use the default configuration, which e-mails weekly Service Summary reports and the Exception report to root as the default recipient of email sent to firewalladmin.

You can customize the events that your firewall ignores in the exception reports from *gauntlet-admin*. You can also customize the report recipient, enable and disable daily and weekly Service Summary reports, and customize the Exception reporting interval.

Configuring Events to Ignore

You can configure the events that the reporting scripts ignore when parsing the logs. This allows you to configure the firewall to ignore events that you know are routine for your situation.

To modify the events that the reporting scripts ignore, modify the list of events on the Proxies form in *gauntlet-admin*. Use regular expressions to denote events that are not significant.

Configuring the Firewall

To change your reporting options, use the *gauntlet-admin* interface.

To set reporting options, follow these steps:

1. Set the recipient of the report to the person or alias to which the firewall should e-mail reports.
2. Enable the daily reports option if you want the firewall to mail copies of the daily Service Summary reports.
3. Enable the weekly reports option if you want the firewall to mail copies of the weekly Service Summary reports.
4. Set the frequentcheck interval option to the frequency at which the firewall will scan the logs, then create and mail the Exception report.

Reading Logs and Reports

The logs and reports that the firewall writes are in ASCII, easy for you and reporting scripts to read. This section presents a brief overview of what the logs and reports look like, and what the items indicate.

Logs

The log file (*/var/log/SYSLOG*) contains a chronological list of events written by the kernel, proxies, authentication management system, and other processes. The sample below shows all of the events that the firewall logged in a two-minute period between 10:47:00 and 10:48:59.

```
Oct 30 10:47:22 firewall http-gw[12079]: permit host=unknown/10.0.1.17 use of gateway (Ver g3.0.3 /   0)
Oct 30 10:47:22 firewall http-gw[12079]: log host=unknown/10.0.1.17 protocol=HTTP cmd=dir
dest=www.tis.com path=/
Oct 30 10:47:23 firewall http-gw[12079]: content-type= text/html
Oct 30 10:47:23 firewall http-gw[12079]: exit host=unknown/10.0.1.17 cmds=1 in=2392 out=0 user=unauth
duration=6
Oct 30 10:47:23 firewall http-gw[12080]: permit host=unknown/10.0.1.17 use of gateway (Ver g3.0.3 /   0)
Oct 30 10:47:23 firewall http-gw[12080]: log host=unknown/10.0.1.17 protocol=HTTP cmd=get
dest=www.tis.com path=/art/actual/title.gif
Oct 30 10:47:25 firewall http-gw[12080]: content-type= image/gif
Oct 30 10:47:27 firewall http-gw[12080]: exit host=unknown/10.0.1.17 cmds=1 in=5581 out=0 user=unauth
duration=4
Oct 30 10:47:28 firewall http-gw[12081]: permit host=unknown/10.0.1.17 use of gateway (Ver g3.0.3 /   0)
Oct 30 10:47:28 firewall http-gw[12081]: log host=unknown/10.0.1.17 protocol=HTTP cmd=get
dest=www.tis.com path=/art/buttons/2.netsec.gif
Oct 30 10:47:28 firewall http-gw[12081]: content-type= image/gif
Oct 30 10:47:28 firewall http-gw[12081]: exit host=unknown/10.0.1.17 cmds=1 in=135 out=0 user=unauth
duration=0
Oct 30 10:48:24 firewall smap[12082]: connect host=cosmo.clientsite.com/192.94.214.96
Oct 30 10:48:24 firewall smap[12082]: host=cosmo.clientsite.com/192.94.214.96 bytes=1005
from=<bob@clientsite.com> to=<@firewall.trusted.com:clancy@yoyodyne.com >
Oct 30 10:48:24 firewall smap[12082]: exiting host=cosmo.clientsite.com/192.94.214.96 bytes=1005
Oct 30 10:48:39 firewall sendmail[12084]: KAA12084: from=<bob@clientsite.com>, size=921, class=0,
pri=30921, nrcpts=1, msgid=<9510301544.AA04030@clientsite.com>, relay=uucp@localhost
Oct 30 10:48:39 firewall smapd[12083]: delivered file=sma012082
Oct 30 10:48:40 firewall sendmail[12086]: KAA12084: to=<@firewall.yoyodyne.com:clancy@yoyodyne.com>,
ctladdr=<bob@clientsite.COM> (6/0), delay=00:00:01, mailer=smtp, relay=mail.yoyodyne.com. [10.0.1.126],
stat=Sent (Ok)
```

Service Summary Reports

The Service Summary reports contain a concise overview of events by service. The example below shows the weekly information for TELNET activity through the firewall:

```
Telnet/Rlogin Proxy Usage
-----------------------------------------------------------------

Top 100 telnet gateway clients (total: 308)
Connects      Host/Address               Input    Output     Total
--------      ------------               -----    ------     -----
   287        dimension.yoyodyne.com/    267484     11412    278896
     6        eight.yoyodyne.com/10.0    495575      2249    497824
     6        jersey.yoyodyne.com/10.    291915      3608    295523
     3        lizardo.yoyodyne.com/10      4204       318      4522
     2        john.yoyodyne.com/10.0.    472366      4719    477085
     2        planet10.yoyodyne.com/1       123        64       187
     1        blaze.clientsite.com/20    169588      1473    171061
     1          unknown/204.254.155.2         0         0         0

Top 100 telnet gateway clients in terms of traffic
Connects      Host/Address               Input    Output     Total
--------      ------------               -----    ------     -----
   287        dimension.yoyodyne.com/    267484     11412    278896
     2        john.yoyodyne.com/10.0.    472366      4719    477085
     6        jersey.yoyodyne.com/10.    291915      3608    295523
     6        eight.yoyodyne.com/10.0    495575      2249    497824
     1        blaze.clientsite.com/20    169588      1473    171061
     3        lizardo.yoyodyne.com/10      4204       318      4522
     2        planet10.yoyodyne.com/1       123        64       187
     1          unknown/204.254.155.2         0         0         0
```

Exception Reports

The Exception reports contain a chronological summary of security alerts and potential items of interest. The example below shows the information for a fifteen-minute interval on the firewall:

```
Security Alerts
---------------
Dec 12 10:18:35 gauntlet kernel: securityalert: tcp from 10.0.1.17 on unserved port
191
Dec 12 10:19:13 localhost authsrv[2190]: securityalert: repeated bad auth attempts penny
(rlogin-gw unknown/10.0.1.17)
```

```
Possible Items of Interest
-------------------------
Dec 12 10:16:11 localhost authsrv[2176]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:16:13 localhost authsrv[2176]: BADAUTH root (rlogin-gw unknown/10.0.1.17)
Dec 12 10:18:12 localhost authedit[2185]: root ENABLED USER penny
Dec 12 10:18:52 localhost authsrv[2188]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:18:55 localhost authsrv[2188]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:19:03 localhost authsrv[2188]: BADAUTH nobody (rlogin-gw unknown/10.0.1.17)
Dec 12 10:19:05 localhost authsrv[2188]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:19:10 localhost authsrv[2190]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:19:13 localhost authsrv[2190]: BADAUTH penny (rlogin-gw unknown/10.0.1.17)
Dec 12 10:19:14 localhost authsrv[2190]: BADAUTH penny too many tries (rlogin-gw
unknown/10.0.1.17)
Dec 12 10:20:00 gauntlet kernel: uid 0 on /: file system full
```

Chapter 21

Backups and System Integrity

Backing Up Your Firewall

Your firewall is an integral part of your system, configured to pass traffic between your internal network and all external networks. If the hard disk crashes on the firewall, you want to restore your system as quickly as possible. Backing up your firewall is an essential administrative task.

Backup Considerations

To back up the data on your firewall, use standard IRIX backup procedures as described in the *IRIX Advanced Site and Server Administration Guide*.

In particular, you should be sure to back up the following:

- */usr/gauntlet/cgi-data*
- */usr/gauntlet/config*
- */usr/etc/fw-authdb**
- */etc/apop.pass*
- */etc/skeykeys*
- */usr/gauntlet/checksums*
- */var/adm*

Note that if you perform normal backups of the firewall system as you would any IRIX system, these files are going to be backed up, but be sure to verify that, because these are the most crucial.

Since relatively few files (except for logs) are going to change often on the firewall, incremental backups require little space; therefore frequent backups should not be a painful task.

Restoring the Firewall

You hope that you never have to restore your firewall, but you may need to. For most activities, restoring your Gauntlet firewall is much like restoring any IRIX system. You can also create your own scripts or integrate these restoration activities into your normal restore routine for other IRIX machines.

Restoring the Logs and Reports

You will generally not restore the logs and reports onto their original locations on the firewall, as this would overwrite current information with old data. You might, however, need the backup copies of logs and reports to track usage trends or look for signs of an attack. Restore the logs and information to another machine for this sort of analysis.

Verifying System Integrity

Even though you've created only one account on the firewall for the administrator, you still want to ensure that no person or process has modified your system. The Gauntlet Internet firewall is designed to make it easy to verify system integrity.

Understanding System Integrity

The Gauntlet integrity database is collection of cryptographic checksums or message digests for many files on your filesystem. The database contains a checksum for each file, using information about the file size, date, user ID, group ID, and mode. The database does not contain information about files that can change often, such as the mail spool, the log files, and system aliases. You expect these files to change, so the checksums would always be different.

The integrity database, an ASCII file, is automatically created (unless it already exists) during the weekly report.

Configuring the Files to Ignore

You can modify the list of files and directories that the scan program ignores when creating and checking databases. This allows you to ignore directories and files that you know are volatile.

To configure the files to ignore, modify the list of directories and files in */usr/gauntlet/checksums/scan.conf*.

Protecting the Integrity Database

You use the integrity database to verify that nothing has modified your system. Therefore you must protect the database itself from tampering. You can leave the database on-line. You should also copy it to removable media that you can keep off-line for safekeeping.

Store a copy of the initial integrity database created during the first weekly report with your original distribution media.

Verifying System Integrity

If you elect to receive weekly reports, you will automatically receive the results of a system integrity check. If you do not elect to receive these reports, integrity checking is not performed.

Understanding the Results

Review the changes noted in the weekly report and ensure that they are acceptable changes. For example, you may have changed the root password on the Gauntlet firewall during the past week, resulting in the report of a change in */etc/passwd*. This would be an acceptable change.

Appendixes

Appendix A

Gauntlet System Files

This chapter appendix discusses some of the files that you would normally manipulate through the GUI, and provides details on editing the network permission tables.

Warning: Unless you are quite familiar with these files, you should use the GUI and never edit these files directly. Note also that editing these files directly can effectively force you to no longer use the GUI to configure them as they will no longer be in a state and format known to the GUI.

Viewing the Gauntlet File List

If you want to see a list of the files that the Gauntlet software manipulates, click the *view* link in the "Managing Your Firewall" portion of the introductory form. If you do not want to use the forms-based interface, you can directly edit these files, although that is not recommended.

Table A-1 lists files that may be modified through this interface. Some of these files are safe for you to modify, as long as nobody else is also running gauntlet-admin at the same time. "Safe" here means that your changes will not be lost. Other files are unsafe for you to modify; "unsafe" here means that the *gauntlet-admin* interface, including the *configure_all* script, may overwrite any changes you make. Filenames without a leading "/" are relative to the path */usr/gauntlet/*.

Note: The online list of Gauntlet files is always the most current list. Use the online *view* link to see the current Gaunlet file list.

Table A-1 The Gauntlet File List

Filename	Safe?	Description
//*.old.12345*	Yes	To save copies of certain configuration files, Gauntlet will use the convention of appending ".old." to the filename and then the process ID of whatever Gauntlet program is actually making the change.
//*.new.12345*	No	While creating new versions of configuration files, Gauntlet uses the same convention as when saving copies of configuration files. Such files should be removed by Gauntlet when it is done performing whatever task it is up to.
cgi-data/.g*	Yes	Stores settings from the configuration pages.
config/trusted-networks	Yes	Lists networks which are to be considered trusted.
config/untrusted-networks	Yes	Lists networks which are to be considered untrusted.
config/trusted-ports	Yes	Lists ports on which traffic will be permitted to pass through the firewall unimpeded.
config/trusted-interfaces	Yes	Lists interfaces on which traffic from trusted networks will be accepted. All other packets claiming to be from trusted networks but which come in over other interfaces will be rejected.
config/.txt*	Yes	Text files which are displayed by the respective application proxies under certain circumstances. For example, rlogin-deny.txt would be displayed by the rlogin application proxy if access is denied.

Table A-1 (continued) The Gauntlet File List

Filename	Safe?	Description
config/subdomain	Yes	Subdomains which will be accepted by the firewall for mail delivery if you have selected to let Gauntlet rewrite sendmail.cf.
config/explicit-routes	Yes	Lists explicit (static) routes to be installed into the routing tables via /etc/gated.conf.
config/frequentcheck.ignore	Yes	Lists egrep-style regular expressions which will be used to filter the system logs. Lines which match expressions listed in this file will not be displayed in the "Possible Items of Interest" part of the Gauntlet reports.
config/swipe.conf	Yes[*]	Stores information about configured swIPe peers and paths. Editing this file is not recommended, although it is safe to do so, because the format of this file is obscure.
config/authserver-protocols	No	Lists DSO (Dynamic Shared Object) files which support additional authentication mechanisms. This will be updated by Gauntlet when you install or remove Gauntlet authentication software subsystems using *inst*.
config/netperm-table	No	Used by application proxies to decide whether to permit access or not. Gauntlet creates this file by performing substitutions on the file */config/template.netperm-table*, which is safe to edit.
server/web_passwd	No	Updated by gauntlet-admin using the password for the user gauntlet in */etc/passwd*.

Table A-1 (continued) The Gauntlet File List

Filename	Safe?	Description
/etc/ipfilterd.conf	No	Configuration file for the ipfilterd daemon. Gauntlet creates this file by performing substitutions on the file *config/template.ipfilterd.conf*, which is safe to edit.
/etc/config/routed. options	No	Configuration file for the *routed* daemon.
/etc/gateways	No	Configuration file for the gated routing daemon.
/etc/passwd	Yes	System password file. Gauntlet modifies this file in several different ways: it inserts '*' into the password field for accounts which do not have a password, so that all accounts which can be used for login are passworded; it forces root to have a password; and it inserts a gauntlet user which cannot log in but whose password is used to control access to gauntlet-admin.
/etc/sendmail.cf	Maybe	Sendmail configuration file. It is safe to modify this file only if you have selected preserving sendmail.cf on the sendmail page.
/etc/aliases	Yes	Gauntlet modifies the alias for root on the firewall machine, and adds a firewalladmin alias.
/etc/group	Yes	Gauntlet may need to add groups to this file for various application proxies.
*/etc/config/**	[*]	Gauntlet forces certain chkconfig settings on or off. Among some of the settings are smap, sendmail, named, ipfilterd, gated, routed, outbox, and nfs.

Table A-1 (continued) The Gauntlet File List

Filename	Safe?	Description
/etc/default/login	Yes[*]	The "minimize_exposure" script on the initial page will adjust the variable settings in this file to make login more secure on the firewall host. Since you don't need to run "minimize_exposure" more than once, afterwards you may tweak this file to suit your needs.
/etc/inetd.conf	No	Gauntlet will comment out all but a few IRIX-specific services which it itself needs to run.
/etc/skeykeys	Yes	If you add or edit a user's authentication password, Gauntlet will invoke S/Key keyinit for you, which stores information in this file.
/etc/named.boot	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.
*/tmp/retry.**	Yes	Retry files are created to support data entry validation in the gauntlet-admin interface.
/var/named/localhost.rev	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.
/var/named/named.hosts	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.
/var/named/named.rev	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.

Table A-1 (continued) The Gauntlet File List

Filename	Safe?	Description
/var/named/root.cache	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.
/var/spool/cron/crontabs/root	Yes	Gauntlet adds various jobs to run at regular intervals.
/usr/etc/resolv.conf	Maybe	DNS configuration file. It is safe to modify this file only if you have selected preserving your DNS configuration on the DNS page.

Netperm Table

The network permissions table (*/usr/gauntlet/config/netperm-table*) contains configuration information for the Gauntlet Internet Firewall. The kernel, proxies and other applications read their configuration information from this table. The rules in the table include two types of information: policy rules and application-specific rules.

Note: This structure differs from previous versions of the *netperm-table* file. The proxies and other applications still recognize *netperm-table* files from version 2.0 and higher. You may wish to convert your *netperm-table* file to the new format soon for two reasons. First, the new policy-based table is much easier to use because you can use generic rules. Second, future versions of the proxies and applications will not always support the older table format. Remember to make a backup copy of your working *netperm-table* file before you attempt any conversions.

Note: Gauntlet uses */usr/gauntlet/config/template.netperm-table* to create (thus overwriting) */usr/gauntlet/config/netperm-table*. Any modifications you wish to be permanent must be made to the *template.netperm-table* file.

Policy Rules

Policies are collections of general configuration information. These allow you to closely map your security policy to policies for the Gauntlet Firewall. Gauntlet configuration policies often include information such as

- types of proxies that the firewall can start
- permitted (or denied) destinations for requests
- authentication requirements

The source address of the request is the basis for a policy. You define policies for a set of hosts, rather than defining rules on a proxy-by-proxy basis as in previous versions. You can easily use the same set of rules for a group of hosts by creating a generic policy

describing what these hosts can and cannot do. The default Gauntlet configuration defines two policies: an inside policy and an outside policy.

The inside policy defines the general policies for requests from the inside (trusted) networks. This policy indicates that proxies can send requests to any destination. By default it permits some of the more commonly used proxies for inside requests: TELNET, rlogin, FTP, NNTP, HTTP, and X11. This policy also allows users to change their passwords for non-third party authentication systems from the inside networks.

The outside policy defines the general policies for requests from the outside (untrusted) networks. This policy indicates that proxies can send requests to any destination. By default it permits some of the more commonly used proxies for outside requests: TELNET, rlogin, FTP, NNTP, POP3, X11, and Info Server. It requires strong authentication for all outside requests with the authentication server that is on the firewall.

Notice that the outside policy does not permit the HTTP proxy, because you generally do not want people all over the Internet accessing Web servers on your internal network. It does however allow the Info Server, which allows you to run an HTTP, Gopher, or FTP server on your firewall.

Application-Specific Rules

The *netperm-table* file also includes configuration information for proxies and other firewall applications. These include

- user ID and group ID under which a proxy should run
- directories which the proxies should use as their root directories
- text files that proxies should display when denying or accepting requests
- length of idle time before the proxies should terminate the connection
- more specific lists of permitted and denied destination networks for a particular proxy

Proxies

For example, the smap proxy reads the *netperm-table* file and determines the user ID under which it should run and the directory into which it should place mail. The

TELNET proxy reads the *netperm-table* file to determine how long a session must be idle before it should disconnect the session.

You can also include rules to permit or deny a particular service for requests to specific addresses or networks. For example, you can configure the HTTP proxy to deny requests to a particular host or network. All of the other proxies, such as the *smapd* server, continue to use the generic policy and send information to that site, while the HTTP proxy denies requests to that site.

Because the proxies and applications read the *netperm-table* file from top to bottom, you must put proxy-specific rules before the generic policies. When the relevant proxy parses the configuration information, it uses the proxy-specific rule rather than the more general policy rule.

For example, the FTP proxy includes a generic rule that denies requests to the destination ftp.bigu.edu. The general outside policy, near the bottom of the *netperm-table* file includes a rule that allows all proxies and applications to send to any destination. Because the more restrictive rule is above the generic policy in the *netperm-table* file, the FTP proxy uses the restrictive rule and denies requests to ftp.bigu.edu.

Applications

Other Gauntlet applications such as the authentication server also read configuration information from the *netperm-table* file.

Using This Information

As part of the startup process a proxy or application reads the *netperm-table* file looking for applicable configuration rules. It parses the table from top to bottom, looking for rules that match its name. It also matches wildcard rules that apply to all applications. For example, the TELNET proxy (*tn-gw*) looks for rules that match tn-gw and *.

The proxy first uses these rules to determine if it can accept the request from the source address. It then determines whether the requested service is an explicitly permitted service. If it is not, the proxy denies the request. If it can accept the request, it uses the other rules to determine whether it needs to authenticate the request, and whether it can send the request to the specified destination. The application also finds and uses rules for that specific application.

For example, using the default outside policy, the TELNET proxy allows TELNET requests from any outside network to any destination. The proxy also uses the outside policy to determine that it needs to authenticate the user and gets information about which server it should use to authenticate the user.

Modifying the Netperm Table File

Modify the */usr/gauntlet/config/template.netperm-table* file using your favorite text editor. Be sure to make a backup copy. You do not need to restart the proxies to make the changes take effect. The proxies reread the table anytime the file date and time change.

Netperm table Syntax

Precedence

Applications and proxies read the tables from the top of the table to the bottom. They use the first rule that applies for a particular attribute. If there are multiple rules in the table that could apply for an attribute, the application uses the first one it finds. For example, suppose a *netperm-table* file contains the following rule:

```
smapd: userid uucp
```

Later in the file, it also contains this rule:

```
smapd: userid mail
```

When *smapd* parses the *netperm-table* file, it uses the first rule it finds, and runs as the user uucp.

Format

Each line in the *netperm-table* file contains a separate configuration rule in the format:

keyword: attribute valuelist

where

- *keyword* indicates the application to which the rule on that line applies. The wildcard (*) indicates that the rule is valid for all applications and proxies. Comma-separated lists of multiple keywords indicates that the rules apply to the proxies or applications listed. The keyword usually matches the name of the service. It can also match the value for the **-as** name flag used when starting the proxy.
- *attribute* is a configuration parameter for that application or proxy.
- *valuelist* is the value for the specific configuration parameter. Some attributes allow multiple values.

A rule must fit on a single line. The length of a line varies by operating system, but is generally around 1,024 bytes. There is no provision for continuing lines.

White space and tabs are both valid separators.

Comments

A hash mark (#) at the beginning of a line indicates a comment. Applications ignore any text between the # and the end of the line.

Substitution Lines

Some lines start with GAUNTLET_SUBSTITUTE. *Do not* delete these lines.

Keywords

This table lists some default and common keywords for policies, proxies and other applications. You can create your own keywords. Be sure that the keyword matches the value for the *-as* name flag you used when starting any custom proxies.

Table B-1 Default and Common Keywords

Keyword	Application
authsrv	authentication server
ftp-gw	FTP proxy
gopher-gw	Gopher proxy (using the http-gw proxy)
http-gw	HTTP proxy
lp-gw	line printer proxy
netacl-fingerd	network access control proxy running finger service
netacl-ftpd	network access control proxy running FTP service
netacl-rlogind	network access control proxy running rlogin service
netacl-telnetd	network access control proxy running TELNET service
nntp-gw	NNTP news proxy (using the plug-gw proxy)
policy-trusted	policy for requests from inside networks
policy-name	policy
policy-untrusted	policy for requests from outside networks
pop3-gw	POP3 mail proxy
rap-gw	RealAudio proxy
rlogin-gw	Rlogin proxy

Table B-1 (continued) Default and Common Keywords

Keyword	Application
smap	SMTP mail server
smapd	SMTP mail client
tn-gw	TELNET proxy

Attributes

Attributes vary by proxy and application, though many use the same attributes. Consult the reference information at the end of this chapter for more information on applicable attributes and values.

Creating New Policies

You can create additional policies to fit your security policies for different groups of inside hosts and networks. Remember that all policies are based on the source address of the request.

Creating a new policy involves modifying the *netperm-table* file.

To create a new policy, follow these steps:

1. Add a line indicating
   - source networks that use the policy
   - the name of the policy
2. Add rules indicating which proxies this policy allows.
3. Add rules indicating permitted destinations, authentication, and logging.
4. Place the policy lines above or below the generic policies as appropriate.

For example, the generic policy for Yoyodyne uses the default Gauntlet inside policy. The security policy for Yoyodyne calls for restricting a particular group of machines (and set of addresses) to TELNET and restricting rlogin to a particular set of outside networks.

To implement this policy, you could create a more restrictive policy:

1. #define inside hosts who will use the policy
2. *: permit-hosts 204.255.154.0:255.255.255.128 -policy restrictive
3. #define the policy
4. policy-restrictive: permit-proxy tn-gw rlogin-gw
5. policy-restrictive: permit-destination 192.33.112.*
6. policy-restrictive: authenticate *
7. policy-restrictive: auth server 127.0.0.1

Line 2 indicates that all proxies and applications (*) should use the restrictive policy for requests from the designated subnet. If you specify the policy for only the TELNET (*tn-gw*) and rlogin (*rlogin-gw*) proxies instead of for all (*), all other proxies (such as the HTTP and FTP proxies) skip this policy and use another policy.

Line 4 indicates that this policy permits the TELNET and rlogin proxies. All other proxies with requests from hosts within 204.255.154.0:255.255.255.128 deny the request after parsing this line.

Line 5 indicates that these proxies can send requests to the set of destinations: 192.33.112.*. The TELNET and rlogin proxies deny requests to any other destinations after parsing this line.

Lines 6 and 7 indicate that users on these networks must authenticate with the authentication server on the firewall.

Put this policy above the inside policy so the proxies will use these rules rather than the more generous inside policy. You may also want to create a matching restrictive outside policy to restrict access from outside networks to this internal subnet.

Note that this type of policy may not prevent users on this inside network from reading news and sending e-mail. The recommended setup for the Gauntlet firewall calls for central mail and news servers on the inside networks. The news readers and mail agents on the restricted subnet communicate directly with the news and mail servers. These servers, which are not on the restricted subnet, communicate with the firewall.

If you are running mail and news servers on the firewall, this more restrictive policy denies email and news activities from the restricted subnet.

Adding Proxy Services

You can add or remove proxy services at any point as your security policies change. This section addresses the changes you must make to the *netperm-table* file to use the proxy. Consult the chapter for each proxy for more information on other configuration requirements.

To add a proxy service, follow these steps:

1. Add the name of the proxy to the permit-proxy line of the appropriate policy.
2. Add a section for proxy-specific rules above the policy sections. These can include items such as user ID, group ID, time-out, and denial messages. Consult the reference information for the proxy for information on proxy options.

For example, after careful analysis, Yoyodyne wants to add support for Quote of the Day (*qotd*) service for users on its inside networks. This involves using the proxy. First, add a line to the inside policy:

```
135     policy-inside:     permit-proxy qotd-gw
```

Then create a section above the policies in which you define the communications rules for the Quote of the Day connection:

```
95     # QotD (through plug proxy) rules
96     # ----------------------------------------
97     qotd-gw: port qotd * desthost qotd.bigu.edu -destport qotd
```

Denying Services By Network or Host

You can deny services to and from specific networks and hosts. You can do this for all the proxies through a policy, or for individual proxies.

Denying Access From a Host or Network

You can deny access from a particular host or network on a proxy or general basis.

Denying Access by Proxy

To deny access by proxy, add a deny-hosts line to the specific proxy.

For example, Yoyodyne does not want anyone on a host at Big University to have TELNET access to Yoyodyne:

```
50      tn-gw:      deny-hosts *.bigu.edu
```

Later, Yoyodyne determines they only need to deny access from the dialin machines at Big University:

```
50      tn-gw:      deny-hosts dial*.bigu.edu
```

Denying Access in General

You can also deny access from a particular host or network for all proxies and applications.

To deny access for all applications, add a `deny-hosts` line above the outside policies. Use a wildcard as the keyword to indicate that the rule applies to all policies.

You must include this rule above the policy rules. The policies are based on permitted hosts. Including the deny-hosts rule in a policy has no effect because the application is using the permit-hosts rule that defines the policy.

Note that the *smap* proxies do not use the policy rules, so you can still receive mail from the denied host or network.

For example, Yoyodyne does not want anyone or any service at Big University to communicate with Yoyodyne:

```
103     *:      deny-hosts *.bigu.edu
  ...
140  *:      permit-hosts * -policy outside
```

Controlling Services by User, Group or Time

You can control access to the following proxies on a per user, per group or time of day basis:

- ck-gw Circuit proxy
- ftp-gw FTP proxy
- rlogin-gw Rlogin proxy

- rsh-gw Rsh proxy
- tn-gw TELNET proxy

User or Group

You can permit or deny access to certain proxies by user or group.

To control access by user or group:

1. Add the **operations** attribute to your **authsrv** configuration specifying who can perform the operation and what services they can access.
2. Add the **authenticate** attribute to the appropriate policy or proxy, requiring users to autehnticate before using the service.
3. Add the **extended-permissions** attribute to the appropriate policy or proxy, indicating that the authentication server should check information specified by the operations keyword.

For example, Yoyodyne wants to permit only members of the group developer to use the Rlogin proxy when accessing outside hosts:

```
55      authsrv:     permit-operation group developer rlogin-gw *
100     rlogin-gw:   authenticate *
101     rlogin-gw:   extended-permissions *
```

These commands prevent any other users who are not members of group developer (in the Gauntlet authentication database) from using the Rlogin proxy.

Operation

You can permit or deny access to certain proxies by time of day:

To control access by time of day:

1. Add the **operations** attribute to your **authsrv** configuration specifying who can perform what operations, and what services they can access, and when.
2. Add the **authenticate** attribute to the appropriate policy or proxy, requiring users to autehnticate before using the service.

3. Add the **extended-permissions** attribute to the appropriate policy or proxy, indicating that the authentication server should check information specified by the operations keyword.

For example, Yoyodyne wants to deny TELNET between 5:00 pm and 11:00 pm:

```
55      authsrv:     deny-operation user * tn-gw * * time 17:00 23:00
56      authsrv:     permit-operation user * tn-gw * *

100     tn-gw:       authenticate *
101     tn-gw:       extended-permissions *
```

Line 55 denies TELNET access between 5:00 pm and 11:00 pm.

Line 56 permits TELNET access. You must include this rule because you must explicitly permit operations when you specify extended permissions.

The deny rule must appear before the permit rule because the proxies use the first matching rule. If you specify the permit rule before the deny rule, the authentication server would never read the deny rule, because the permit rule matches all TELNET operations.

Denying Access to a Host or Network

You can deny access to a particular host or network on a proxy or general basis.

Denying Access by Proxy

To deny access by proxy:

- Add a `deny-destination` line to the specific proxy.

For example, Yoyodyne does not want anyone on the inside networks to FTP files from any hosts at Big University:

```
55     ftp-gw:     deny-destination *.bigu.edu
```

Denying Access in General

You can also deny access to a particular host or network for all proxies and applications.

To deny access for all applications:, add a `deny-destination` line to the appropriate policy.

Note that the *smap* proxies do not use the policy rules, so you can still send mail to the denied host or network.

For example, Yoyodyne does not want anyone on the inside network to communicate with Big University:

```
108     policy-inside:     deny-destination *.bigu.edu
```

Attribute Reference

Attributes vary by proxy and application, though many use the same attributes. Consult the reference information on the following pages for more information on applicable attributes and values.

The bulleted list at the top of each attribute indicates which proxies, applications, or policies can use that attribute. For example, if *tn-gw* is listed, that indicates you can use this attribute for the TELNET proxy. If policy-*policy* is listed, that means you can use this attribute in a policy definition. All proxies that use this policy will then use this attribute. You can always use any attribute after the wildcard (*) keyword. All proxies read this rule.

authenticate

- ftp-gw
- policy-*policy*
- pop3-gw
- rlogin-gw
- tn-gw

Specifies whether or not users must authenticate when accessing these proxies. Proxies that do not support authentication ignore this setting. This option is equivalent to the **-auth** and **-authall** options in previous versions.

Syntax

```
authenticate *
```

*	Provided for future extensibility.

Example

This example requires all requests from hosts on the outside network to authenticate:

```
policy-outside:      authenticate *
```

authserver

- ftp-gw
- policy-*policy*
- pop3-gw
- rlogin-gw
- tn-gw

Specifies the host running the authentication server that the proxies use for authenticating users.

Syntax

```
authserver host [port]
```

host	Specifies the host running the authentication server. Specify by IP address or hostname.
port	Specifies the port on the host that the proxies use for communicating with the authentication server.

Example

This example requires proxies to use the authentication server on the firewall itself using port 7777:

```
policy-outside:      authserver 127.0.0.1 7777
```

authtype

- ck-gw

Specifies the host running the authentication server that circuit proxy uses. The **authtype** attribute takes precedence over the **authserver** attribute.

Syntax

`authtype` *`hosts`* `[-authhost` *`host`*`] [-authport` *`port`*

host	Specifies indicates the hosts for which the circuit proxy authenticates. Specify individual machines, entire networks, or subnets. Use IP addresses or host names. The * wildcard is valid.
-authhost *host*	Specifies the host running the authentication server. Specify by IP address or host name.
-authport *port*	Specifies the port on the host that the circuit proxy uses for communicating with the authentication server.

Example

This example indicates that all hosts authenticate with the authentication server on the firewall itself using port 7777.

```
ck-gw: authtype * -authhost 127.0.0.1 -authport 7777
```

backend

- ahttp-gw

Specifies the name of the executable to which the authenticating HTTP proxy passes requests after handling the authentication. The executable handles FTP, Gopher, and other protocols.

Syntax

`backend` *`executable`*

executable	Specifies the name of the executable to which the authenticating HTTP proxy passes requests after handling the authentication.

Example

This example indicates that the authenticating HTTP proxy passes processing to */usr/local/etc/http-gw:*

```
ahttp:backend /usr/local/etc/http-gw
```

badadmin

- policy-*policy*
- smapd

Specifies the user name to which the *smapd* server forwards mail that it cannot deliver.

Syntax

`badadmin` *`user`*

user	Specifies the name of a user or alias.

Example

This example sends mail to the firewalladmin alias:

```
smapd:      badadmin firewalladmin
```

baddir

- policy-*policy*
- smapd

Specifies the directory in which the *smapd* server places any spooled mail that it cannot deliver normally.

Syntax

`baddir` *`directory`*

directory	Specifies the name of a directory on the same device as the spool directory. Do not include a trailing slash (/) character. Ensure that this directory has the same owner and permission as the normal directory that smap uses.

Example

This example places the undelivered mail in the */var/spool/smap/badmail* directory:

```
smapd:      baddir /var/spool/smap/badmail
```

badsleep

- authsrv

Specifies the amount of time the authentication server disallows logins from a user who has attempted (and failed) to login five times in a row.

Syntax

`badsleep` *`seconds`*

seconds	Specifies the number of seconds the authentication server sleeps before allowing login attempts from a user who has attempted (and failed) to login five times in a row. If this option is set to 0, the authentication server allows an unlimited number of unsuccessful login attempts. If this option is not set, the authentication server disables the account after the user attempts (and fails) to login five times in a row.

Example

This example indicates that the authentication server sleeps for twenty minutes (1200 seconds) after five unsuccessful login attempts:

```
authsrv: badsleep 1200
```

child-limit

- authsrv
- ck-gw
- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl

- plug-gw
- pop3-gw
- rap
- rlogin-gw
- rsh-gw
- syb-gw
- tn-gw
- policy-*policy*

Specifies the maximum number of child processes that each daemon allows to run at a given time.

Syntax

`child-limit` *`processes`*

processes	Specifies the maximum number of child processes that each daemon allows to run at a given time. If this option is set to 0 or not set, each daemon allows an unlimited number of child processes to run at a given time.

Example

This example indicates that the TELNET proxy allows only 10 child processes to run at a single time:

```
tn-gw:      child-limit 10
```

circuitexec

- ck-gw

Specifies the location of the program that the circuit proxy runs once it allows a connection from the client program.

Syntax

```
circuitexec programs
```

programs	Specifies the location and name of the program that the circuit proxy runs once it allows a connection from the client program.

Example

This example indicates that the circuit proxy is in */usr/local/etc*:

```
ck-gw: circuitexec /usr/local/etc/circuit
```

circuitsperuser

- ck-gw

Specifies the maximum number of client / server connections that can be active in one user session.

Syntax

```
circuitsperuser circuits
```

circuits	Specifies the maximum number of client / server connections that can be active in one user session.

Example

This example indicates that a user can have 12 active sessions:

```
ck-gw: circuitsperuser 12
```

circuit-timeout

- ck-gw

Specifies the amount of time the client / server connection is idle (with no network activity) before disconnecting. Overridden by the **-timeout** option for a particular server (as set with the **server** attribute).

Syntax

```
circuit-timeout minutes
```

minutes	Specifies the number of minutes that there is no client/server activity before disconnecting.

Example

This example indicates that the client/server activity can be idle for 15 minutes before disconnecting:

```
ck-gw: circuit-timeout 15
```

client

- lp-gw
- policy-*policy*

Specifies the clients that can print to a particular printer queue and the commands they can execute.

Syntax

```
client clients -printer queue [ [-deny | -log] \
[{lpcommands } | all] ]
```

clients	Specifies single hosts, entire networks, or subnets. Specify by IP address or hostname. The wildcard * is valid.
printer	Indicates the printer queue to which this rule applies.
queue	Specifies the name of the printer queue to which this rule applies.
deny	Indicates commands that clients cannot execute. The default allows users to issue all lp commands.
log	Indicates extended logging applies. Extended logging includes the number of bytes transferred from client to server and time duration. Extended logging does not include data transfer from server to client, as this consists mostly of acknowledgments to client's command.

lpcommands	Specifies the lp commands that the clients can issue when sending jobs through the proxy. The space between the "{" and "}" and the list entries is required. Valid keywords, which correspond to the first level lp protocol commands, are: restart, print, status_sh, status_ln, remove
all	Indicates that the deny or log command applies to all lp commands.

connect-timeout

- ck-gw

Specifies the amount of time the user has to start the client application before the proxy stops listening at the service port. This attribute also controls the amount of time the user has to respond to the query asking them to allow the connection.

Syntax

`connect-timeout` *`minutes`*

minutes	Specifies the number of minutes the proxy waits at the service port for a client application connection before disconnecting.

Example

This example indicates that the user has 3 minutes to start the client application before the proxy stops listening:

```
ck-gw: connect-timeout 3
```

database

- authsrv

Specifies the pathname of the database that the authentication server uses. This option is mandatory, unless you compile the authentication server with a specific database path.

Syntax

`database` *`path`*

path	Specifies the path of the database that the authentication server uses.

Example

This example indicates that the authentication server uses the authentication database in */usr/local/etc/fw-authdb*:

```
authsrv: database /usr/local/etc/fw-authdb
```

denial-msg

- ftp-gw
- policy-*policy*
- rlogin-gw
- tn-gw

Specifies the file that the proxy displays when it denies access to a user because they do not have permission to use the proxy.

Syntax

```
denial-msg file
```

file	Specifies the name of the file the proxy displays when it denies access to a user because they do not have permission to use the proxy. If no file is specified, the proxy generates a default message.

Example

This example displays the file */usr/local/etc/ftp-deny.txt* when the FTP proxy denies access to a user:

```
ftp-gw:     denial-msg     /usr/local/etc/ftp-deny.txt
```

denydest-msg

- ftp-gw
- http-gw
- policy-*policy*
- rlogin-gw
- tn-gw

Specifies the file that the proxy displays when it denies access to a user because they are trying to access a destination they are not permitted to access.

Syntax

```
denydest-msg file
```

file	Specifies the name of the file the proxy displays when it denies access to a user because they are trying to access a destination that they are not permitted to access. If no file is specified, the proxy generates a default message.

Example

This example displays the file */usr/local/etc/tn-denydest.txt* when the TELNET proxy denies access to a user:

```
tn-gw:      denydest-msg      /usr/local/etc/tn-denydest.txt
```

destination

- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw
- policy-*policy*
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw
- tn-gw

Specifies destination hosts and networks permissions.

Syntax

`[permit |deny]-destination` *`destination-list`*

permit	Indicates hosts to which the proxies and applications can send requests.
deny	Indicates hosts to which the proxies and applications cannot send requests.
destination-list	Specifies single hosts, entire networks, or subnets. Specify by IP address or hostname. The wildcard * is valid. If no destination-list is specified, no destinations are valid.

Example

This example permits applications to send requests to hosts on the **192.3.4** network:

```
policy-restrictive:     permit-destination 192.3.4.*
```

directory

- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw
- smap
- smapd
- tn-gw
- x-gw

Specifies the directory that the proxy makes its root directory before providing service. This option is equivalent to the *-chroot* option in previous versions.

Syntax

`directory` *`directory`*

directory	Specifies the directory that the proxy makes its root directory before providing service.

Example

This example indicates that the *smap* and *smapd* proxies use the directory */var/spool/smap* as their root directories:

```
smap, smapd:     directory     /var/spool/smap
```

display

- policy-*policy*
- x-gw

Specifies the destination display on which applications display.

Syntax

`display` *`host:displaynumber.screennumber`*

host	Specifies the name of the machine to which the display is physically connected.
displaynumber	Number of the display on the machine.
screennumber	Number of the screen for the display.

Example

This example indicates that the X gateway displays all X applications on the display attached to dimension:

```
x-gw:     display          dimension:10.0
```

exec

- netacl

Specifies a program that the proxy invokes to handle the service. This option is equivalent to the *-exec* option in previous versions.

Syntax

```
exec program [options]
```

program	Specifies the name of the program to invoke.
options	Specifies the command line options for the program.

Example

This example indicates that the *netacl* daemon invokes the *cat* program to display the file */usr/local/etc/finger.txt* for *finger* requests:

```
netacl-fingerd:     exec     /bin/cat /usr/local/etc/finger.txt
```

extended-permissions

- policy-*policy*
- rlogin-gw
- rsh-gw
- tn-gw

Specifies whether the proxies check for extended permissions for users as they authenticate. This option is equivalent to the *-extend* and *-extnd* options in previous versions.

Syntax

```
extended-permissions
```

Example

This example indicates that the proxies check for extended permissions when authenticating users from the outside network:

```
policy-outside:      extended-permissions
```

feature

- http-gw

Allows the proxy to control general features rather than specific portions of the HTTP protocol.

Syntax 1

Specifies particular features of that are explicitly permitted or denied. Denying a feature causes the HTTP proxy to remove the related tags from within the HTML code.

```
{permit | deny}-feature features
```

features	Lists particular HTTP features. Valid features are: frames, java, javascript.

Example 1

This example indicates that the HTTP proxy removes Java or Javascript tags from within any HTML accessed through the proxy:

```
http-gw: deny-feature java javascript
```

Syntax 2

```
feature features
```

features	Lists particular HTTP features. Valid features are: html2.

Example 2

This example indicates that the HTTP proxy removes from any HTML it accesses all HTML that does not meet the HTML2 standards:

```
http-gw: feature html2
```

force_source_address

- plug-gw

Specifies that the plug proxy uses the IP address of the originating host as the source address of the packet when sending the request on to the destination host.

Syntax

```
force_source_address true
```

If this option is not specified, the firewall uses its IP address as the source address of the packet, causing all packets to look like they originated on the firewall.

Note that you must remove or comment out this setting if you wish to disable it. The settings `force_source_address false` and `force_source_address off` are *not* valid.

You must be using officially registered, routable addresses on your trusted networks in order to use this option.

Example

This example indicates that the plug proxy for America Online™ will use the IP address of the originating host as the source address of the packet when sending the packet on to the destination host:

```
aol-gw: force_source_address true
```

forward

- http-gw

Specifies the name of a host to which the HTTP proxy forwards requests for which it can find no destination information.

Syntax

```
forward pattern -protocol protocol -tohost host:port
```

pattern	Specifies the pattern in the URL for which the HTTP uses this rule. Quotes are not required.
protocol	Specifies the protocol that the HTTP proxy uses when talking to the remote host. Valid values are: FTP, GOPHER, HTTP
host:port	Specifies the host and port to which the HTTP proxy forwards requests and the port on which it connects. Use IP addresses or host names. Specify port by port number.

The HTTP proxy uses this information as a last resort, when it cannot find any other information in the request. This is used when transparency is not enabled.

Example

```
http-gw: forward /pub* -protocol ftp -tohost ftp.bigu.edu
```

function

- ftp-gw
- http-gw

Specifies particular functions of the protocol that are explicitly permitted or denied.

Syntax

```
{permit | deny}-function functions
```

functions	Specifies functions that are permitted or denied. Consult the ftpd(1) reference manual page for a list of supported ftp functions.

Valid values for the HTTP proxy are:

- BINARY— Read Files
- DIR—List Directories
- EXEC—Exec Commands

- FTP—FTP Requests
- GOPHER—Gopher Requests
- HTTPREQ—HTTP Requests
- PLUS—Gopher+ Commands
- TEXT—Read Files
- UNKNOWN—Unknown Requests
- WAIS—Search Commands
- WRITE—Write Data

Example

This example indicates that the FTP proxy does not allow people to retrieve (RETR) files:

```
ftp-gw: deny-function RETR
```

This example indicates that the HTTP proxy does not allow people to perform FTP requests through the HTTP proxy:

```
http-gw: deny-function FTP
```

groupid

- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw
- smap

- smapd
- tn-gw
- x-gw

Specifies the group ID the proxy uses when running.

Syntax

`groupid` *`group`*

group	Specifies the name of the group as either a name or numeric id from the */etc/group* file.

Example

This example indicates that the Info Server runs using the group ID of uucp:

```
info-gw:      groupid          uucp
```

handoff

- http-gw

Specifies the name of a host to which the HTTP proxy hands the proxy request. This allows you to use several proxies, such as the HTTP proxy on the firewall and a caching proxy.

Syntax

`handoff` *`host`*`[`*`:port`*`]`

host:port	Specifies the host and port to which the HTTP proxy forwards requests and the port on which it connects. Use IP addresses or host names. Specify port by service name or port number. If no port number is specified, the proxy uses port 80 by default.

The HTTP proxy communicates with the next proxy as if it were a client, rather than as another proxy. You cannot use this setting in place of specifing the HTTP proxy in your browser.

Example

This example indicates the HTTP proxy on the firewall inside the network (fw-engineering.engineering.yoyodyne.com) hands all requests to the firewall between the corporate network and the Internet (firewall.yoyodyne.com):

```
http-gw: handoff firewall.yoyodyne.com
```

header

- http-gw

Specifies HTTP headers that the proxy permits or denies. Denying a header causes the HTTP proxy to remove that information from the request when it sends it to the destination host.

Syntax

```
http-gw: {permit | deny}-header header
```

header	Specifies the headers you wish to explicitly permit or deny (remove).

You can only specify one header per line.

Consult the HTTP 1.0/1/1 specifications a for a list of headers. Note that certain headers are always processed by the HTTP proxy and are dealt with specifically:

- Connection
- Content-Length
- Content-Type
- Location
- Proxy-Connection

Example

This example indicates that the HTTP proxy removes the user agent header and headers that begin with x- before sending the request on to the destination host:

```
http-gw: deny-header user-agent
http-gw: deny-header x-*
```

help-msg

- ftp-gw
- policy-*policy*
- rlogin-gw
- tn-gw

Specifies the file that the proxy displays when the user accesses the help command.

Syntax

`help-msg` *`file`*

file	Specifies the name of the file the proxy displays when the user accesses the help command. If no file is specified, the proxy displays a list of internal commands.

Example

This example displays the file */usr/local/etc/rlogin-help.txt* when a user requests access from the Rlogin proxy:

```
rlogin-gw:      help-msg      /usr/local/etc/rlogin-help.txt
```

hosts

- authsrv
- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw

- tn-gw
- x-gw

Specifies the hosts for which the proxy uses a particular policy, or the hosts that can use the proxy. Specifies the hosts that cannot use the proxy.

Syntax

`permit-hosts` *`hosts`* `-policy` *`policy`*

`deny-hosts` *`hosts`*

permit	Indicates hosts for which the proxy uses a particular policy, or the hosts that can use the proxy.
deny	Indicates hosts that cannot use the proxy.
hosts	Specifies the hosts for which the proxy uses the particular policy. When used without the -policy option, indicates the hosts that can use the proxy. Specifies single hosts, entire networks, or subnets. Specify by IP address or hostname. The wildcard * is valid.
policy	Indicates the name of the policy these hosts use.
policy	Specifies the name of the policy.

Example

This example indicates that all requests from the network 10.0.4.* use the policy *restrictive*:

```
*:      permit-hosts     10.0.4.* -policy restrictive
```

This example indicates that the host 10.0.1.12 can use the *rsh* proxy:

```
rsh-gw:          permit-hosts 10.0.1.12
```

This example indicates that all the hosts on the 10.0.1.0:255.255.255.0 subnet cannot use the FTP proxy:

```
ftp-gw:          deny-hosts 10.0.1.0:255.255.255.0
```

This example indicates that the authentication server only accepts connections from the firewall itself (localhost):

```
authsrv: permit-hosts 127.0.0.1
```

log

- ftp-gw
- http-gw
- info-gw
- lp-gw
- policy-*policy*

Specifies that proxies log only the operations listed, rather than all operations (the default). This option is equivalent to the *-log* command in previous versions.

Syntax

`log` *`operations`*

operations	Specifies operations that the proxies log.

Valid values for the info-gw are:

- CWD (QUIT
- LIST
- NLST
- NOOP
- PASV
- PORT
- PWD
- RETR
- SIZE
- STOR
- SYSY
- TYPE

Valid values for the HTTP proxy are:

- BINARY Read Files
- DIR List Directories
- EXEC Exec Commands
- FTP FTP Requests
- GOPHER Gopher Requests
- HTTPREQ HTTP Requests

Example

This example requests that the trusted policy log only retrieve (RETR) and storage (STOR) activities:

```
policy-inside:     log     RETR STOR
```

maxchildren

- policy-*policy*
- smapd

Specifies the maximum number of child processes the *smapd* server can fork to handle mail.

Syntax

`maxchildren` *`children`*

children	Specifies the maximum number of child processes the *smapd* server can fork to handle mail.

Example

This example indicates that the *smapd* server can fork no more than 20 children:

```
smapd:     maxchildren     20
```

nobogus

authsrv

Specifies that the authentication server indicates when a userid does not exist when users attempt to login and fail.

Syntax

```
nobogus true
```

If this option is not specified and a user enters a non-existent user name, the authentication server always responds with a bogus challenge.

Note that you must remove or comment out this setting if you wish to disable it. The settings ‘nobogus false’ and ‘nobogus off’ are not valid.

Example

This example indicates that the authentication server indicates that the userid does not exist (rather than displaying a bogus SNK challenge) when users attempt to login and fail:

```
authsrv: nobogus true
```

operation

- authsrv

Specifies explicitly permitted or denied operations for particular users or groups at particular times of day. Note that the authentication server only uses these rules when the policy or the proxy uses the extended-permissions attribute.

Syntax

```
{permit | deny}-operation [user users | group groups] service\
destinations [options] [time start end]
```

users	Specifies the names of users for which the proxies use this rule. The wildcard * is valid.
groups	Specifies the names of groups for which the proxies use this rule. The wildcard * is valid.

service	Specifies the name of a service for which this rule applies. Valid values are: ftp-gw—FTP proxy rlogin-gw—Rlogin proxy rsh-gw—Rsh proxy tn-gw—TELNET proxy *—all of these proxies
destination	Specifies the hosts to which the proxies can or cannot send requests. Specify individual machines, entire networks, or subnets. Use IP addresses or host names. The wildcard * is valid.
options	Specifies particular operations for each protocol that can be controlled. Valid values are: ftp-gw (consult the ftpd(1) reference manual page), rlogin-gw, rsh-gw, tn-gw.
time	Indicates that this rule goes into effect and stops having an effect at a particular time.
start	Specifies the time at which the proxy begins using this rule. Specify time in hours and minutes (between 00:00 and 23:59).
end	Specifies the time at which the proxy stops using this rule. Specify time in hours and minutes (between 00:00 and 23:59).

Example

This example indicates that the sales group is permitted to TELNET to any destination between the hours of 8:00 am and 5:00 pm:

```
authsrv: permit-operation group sales tn-gw * time 08:00 17:00
```

ourname

- http-gw

Specifies the host and domain name that the HTTP proxy uses when creating the URLs (links). Because the firewall may have different host names, this allows you to specify which host name to use.

Syntax

```
ourname hostname
```

hostname	Specifies the name of the host that the HTTP proxy uses when prepending URLs. Specify an individual interface. Use an IP addresses or host name.

Example

This example indicates that the HTTP proxy (if needed) prepends firewall.yoyodyne.com (the inside interface of the firewall) to all URLs when attempting to access them:

```
http-gw:  ourname firewall.yoyodyne.com
```

password change

- policy-*policy*
- rlogin-gw
- tn-gw

Specifies password change options for allowing users to change passwords in authentication management system from within the TELNET and Rlogin proxies.

Syntax

```
[permit | deny]-password change
```

permit	Indicates hosts from which users can change their passwords. This is equivalent to the -passok option in previous versions
deny	Indicates hosts from which users cannot change their passwords. Including a deny-password change rule has the same effect as not including those hosts in a permit-password change rule.

Example

This example allows users on the inside network to change their passwords from both the TELNET and Rlogin proxies:

```
policy-inside:          permit-password change
```

This example allows users to change their passwords using the TELNET proxy. If this is the only permit-password change rule in the *netperm-table* file, users can only change their password from the TELNET proxy (not from the Rlogin proxy).

```
tn-gw:      permit-password change
```

pop-server

- policy-*policy*
- pop3-gw

Specifies the name of the machine on which the POP3 server is running. This option is required for the POP3 proxy.

Syntax

`pop-server` *`host`*

host	Specifies the name of the host on which the POP3 server is running. Specify by IP address or hostname.

Example

This example indicates that the POP3 proxy accesses the POP3 server running on the inside mail hub, mail:

```
pop3-gw:     pop-server     mail
```

port

- plug-gw

Specifies the connection rule for this instance of the plug proxy, including the hosts and the ports.

Syntax

`port` *`port hosts`* `desthost` *`hosts`* `[privport *] [destport` *`port`*`]`

port	Indicates the port on which the plug proxy runs.
ports	Specifies the name or port number, as specified in */etc/services*.

hosts	Specifies hosts from which connections can originate. Specifies single hosts, entire networks, or subnets. Specify by IP address or hostname. The wildcard * is valid
desthost	Indicates hosts to which the plug proxy connects.
hosts	Specifies single hosts, entire networks, or subnets. Specify by IP address or hostname. The wildcard * is valid
privport	Indicates that the proxy uses a reserved port number when connecting.
*	Provided for future extensibility.
destport	Indicates the port on which the plug proxy connects on the remote host.
port	Specifies the name or port number, as specified in */etc/services*.

Example

This example creates allows a plug proxy rule for a Quote of the Day server (running as *qotd-gw*) which allows all hosts to connect to the Quote of the Day server at Big University on the *qotd* port:

```
qotd-gw: port qotd * desthost qotd.bigu.edu -destport qotd
```

printer

- lp-gw
- policy-*policy*

Specifies a mapping from a client's queue name to a server's host and queue

Syntax

```
printer clientqueues -host server -printer serverqueue
```

printer	Indicates the printer for which these rules apply.
clientqueues	Specifies the names of client print queues.
host	Indicates the server on which the remote printer queue is.
server	Specifies the name of the host on which the remote printer queue runs.

printer	Indicates the printer queue name.
serverqueue	Specifies the name of the remote printer queue to which proxy sends the print jobs. If server queue is not specified, the client's queue name will be used as server queue name.

Example

This example maps the printer queue remote to the queue lp3, which is running on blaze.clientsite.com:

```
lp-gw:     printer remote -host blaze.clientsite.com\ -printer lp3
```

prompt

- policy-*policy*
- rlogin-gw
- tn-gw
- x-gw

Specifies the prompt the TELNET and Rlogin proxies use in command mode.

Syntax

`prompt` *`prompt`*

prompt	Specifies a string that the proxy displays in command mode. Quotes are not required, but are recommended for strings that include spaces.

Example

This example indicates that the TELNET proxy displays the prompt `Yoyodyne TELNET proxy>` :

```
tn-gw:     prompt     “Yoyodyne TELNET proxy> ”
```

proxy

- policy-*policy*

Specifies proxy permissions.

Syntax

```
[permit | deny]-proxy proxy-list
```

permit	Indicates proxies that this policy allows to run.
deny	Indicates hosts that this policy does not allow to run. Including a deny-proxy rule has the same effect as not including those proxies in a permit-proxy rule.
proxy-list	Specifies the name of the proxy. This name must match the name specified on the command line to start the proxy. If the proxy was started using a -as flag, use that name here.

Example

This example allows the FTP and HTTP proxies to run:

```
policy-restrictive:     permit-proxy ftp-gw http-gw
```

This example allows a plug proxy configured for webster traffic to run:

```
policy-restrictive:     permit-proxy webster
```

securidhost

- authsrv

Specifies the name of the firewall that is registered as the client host name on the ACE server. Because the firewall may have different host names, this allows you to specify which host name to use.

Syntax

```
securidhost firewall
```

firewall	Specifies the name of the firewall that is registered as the client host name on the ACE Server. Specify an individual machine. Use an IP addresses or host name.

Example

This example indicates the SecurID server communicates with the firewall as firewall.yoyodyne.com:

```
authsrv: securidhost firewall.yoyodyne.com
```

sendmail

- smapd

Specifies an alternate path for *sendmail*, or another mail delivery program you are using to deliver your mail inside your perimeter.

Syntax

```
sendmail program
```

program	Specifies an alternate path for the *sendmail* executable or other program you are using to deliver mail.

Example

This example indicates that the *smapd* server uses the *sendmail* executable in */usr/sbin/sendmail*:

```
smapd:      sendmail     /usr/sbin/sendmail
```

server

- ck-gw

Specifies a server for which the proxy handles client/server connections.

Syntax

```
server service -port remote-port [-host remote-host] [-hostport port]
[-timeout minutes] [-nookay]
```

server *service*	Specifies a symbolic name for the service. Must be unique. Used by the proxy to create the menu of available services.
-port *remote -port*	Specifies the port on the remote host to which the circuit proxy connects. Specify by service name or port number.
-host *remote -host*	Specifies the name of the remote host to which the circuit proxy connects. Specify an individual machines. Use IP addresses or host names. This option is required if you are not using transparency.

-timeout *minutes*	Specifies the number of minutes the client / server connection is idle before disconnecting for this service
-nookay	Specifies that the proxy does not prompt the user to confirm before listening on the service port for a connection.

Example

This example indicates that the circuit proxy provides service for an Oracle server on the host **db.clientsite.com**:

```
ck-gw: server oracle -host db.clientsite.com -port oracle
```

shellfile

- login-sh

Specifies the name of the file in which the login shell finds information about users and their actual shells

`shellfile` *`file`*

file	The name of the file that contains a list of users and their actual shells.

Example

This example indicates that the login shell program looks in the */usr/local/etc/login-shellfile* file for information about users and their shells

```
login-sh: shellfile /usr/local/etc/login-shellfile
```

timeout

- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw

- policy-*policy*
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw
- smap
- smapd
- tn-gw
- x-gw

Specifies the amount of time the proxy is idle (with no network activity) before disconnecting.

Syntax

`timeout` *`seconds`*

seconds	Specifies the number of seconds the proxy is idle before disconnecting.

Example

This example indicates that the inside policy allows 1800 seconds (30 minutes) of idle time before the proxies disconnect:

```
policy-inside:     timeout     1800
```

unknown

- authsrv

Specifies a list of additional names that the authentication server checks (in addition to the authentication database) when checking for extended permissions on a per user basis.

Syntax

`permit-unknown` *`names`*

names	Specifies a list of names, separated by spaces. The wildcard * is valid.

If the user name is not in the authentication database, or in the list of names, the authentication server logs the attempt and indicates that the user is not valid. If the user name is found in the list of names, the authentication server assigns the user name to the group "unknown."

Example

This example indicates that the authentication server considers scooter, hikita and penny to be valid user names when it checks for extended permissions:

```
authsrv: permit-unknown scooter hikita penny
```

url-filter

- http-gw
- policy-*policy*

Specifies characters that you do not want to see in a URL.

Syntax

`url-filter` *`filterlist`*

filterlist	Specifies an xurl-encoded string of characters that you do not want to see in a URL. Consult the HTML RFC, or other HTML specification document for lists of url-encoded characters.

Example

This example indicates that you do not want to see the carriage return / line feed pair in any URLs:

```
http-gw:     url-filter     %0D%0A
```

unknown

- authsrv

Specifies a list of additional names that the authentication server checks (in addition to the authentication database) when checking for extended permissions on a per user basis.

If the user name is not in the authentication database, or in the list of names, the authentication server logs the attempt and indicates that the user is not valid. If the user name is found in the list of names, the authentication server assigns the user name to the group "unknown."

Syntax

```
permit-unknown names
```

names	Specifies a list of names, separated by spaces. The wildcard * is valid.

Example

This example indicates that the authentication server considers **scooter**, **hikita** and **penny** to be valid user names when it checks for extended permissions:

```
authsrv: permit-unknown scooter hikita penny
```

url-filter

- http-gw

Specifies characters that you want to deny in a URL.

Syntax

```
url-filter filterlist
```

urlfilter	Specifies an xurl-encoded string of characters that you want to deny in a URL. Consult the HTML RFC or other HTML specification documents for lists of xurl-encoded characters.

Example

This example indicates that you do not want to see the carriage return / line feed characters in any URLs:

```
http-gw:    url-filter    %0D%0A
```

userid

- ftp-gw
- http-gw
- info-gw
- lp-gw
- netacl
- plug-gw
- policy-*policy*
- pop3-gw
- rap-gw
- rlogin-gw
- rsh-gw
- smap
- smapd
- tn-gw
- x-gw

Specifies the user ID the proxy uses when running. This option is equivalent to the *-user* command in previous versions.

Syntax

`userid` *`user`*

user	Specifies the user as either a name or numeric ID from the */etc/passwd* file.

Example

This example indicates that the smap and smapd processes run as the uucp:

```
smap, smapd:     userid          uucp
```

user-servers

- ck-gw

Specifies the servers a particular user can access. Also specifies which services a particular users sees when they use the circuit proxy menu.

Syntax

```
user-servers { user user | group group } [-deny] service
```

user *user*	Specifies the name of a user who can access a particular service.
group *group*	Specifies the name of a group who can access a particular service.
-deny	Specifies that the user can use all services except those explicitly denied.
service	Specifies the names of particular services. Must match the name of a service specified through a server attribute.

Example

This example indicates that group **grads** can use the accounting service:

```
ck-gw: user-servers group grads accounting
```

user-timeout

- ck-gw

Specifies the amount of time the proxy is idle with no active client connections before disconnecting.

Syntax

`user-timeout` *minutes*

minutes	Specifies the number of minutes the proxy is active with no client connections before disconnecting.

Example

This example indicates that the proxy waits 10 minutes without an active client connection before disconnecting:

```
ck-gw: user-timeout 10
```

wakeup

- smapd

Specifies the amount of time that the *smapd* server sleeps between scans of the spool directory for undelivered mail.

Syntax

`wakeup` *seconds*

seconds	Specifies the number of seconds that the *smapd* server sleeps between scans of the spool directory. If no value is specified, *smapd* uses a default value of 60 seconds.

Example

This example indicates that the *smapd* server sleeps for 120 seconds between scans:

```
smapd:      wakeup          120
```

welcome-msg

- ftp-gw
- policy-*policy*
- rlogin-gw
- tn-gw

Specifies the file that the proxy displays as a welcome banner upon successful connection to the proxy.

Syntax

```
welcome-msg file
```

file	Specifies the name of the file the proxy displays as a welcome banner upon successful connection to the proxy. If no file is specified, the proxy generates a default message.

Example

This example displays the file */usr/local/etc/tn-welcome.txt* when a user successfully connects to the TELNET proxy:

```
tn-gw:     welcome-msg     /usr/local/etc/tn-welcome.txt
```

xforwarder

- policy-*policy*
- rlogin-gw
- tn-gw

Specifies the location of the executable to which the TELNET and Rlogin proxies pass requests for the X proxy. Generally specifies the location of the X proxy.

Syntax

```
xforwarder program
```

program	Specifies the location of the executable to which the TELNET and Rlogin proxies pass requests for the X proxy.

Example

This example indicates that the TELNET and Rlogin proxies use the standard X proxy for requests from the inside network:

```
policy-inside:     xforwarder     /usr/local/etc/x-gw
```

xgateway

- policy-*policy*
- rlogin-gw
- tn-gw

Specifies X11 proxy permissions.

Syntax

```
[permit | deny]-xgateway *
```

permit	Indicates that the TELNET and Rlogin proxies can accept requests to start the X11 proxy.
deny	Indicates that the TELNET and Rlogin proxies do not accept requests to start the X11 proxy.
*	Provided for future extensibility.

Example

This example allows the hosts on the inside network to start the X11 proxy:

```
policy-inside:          permit-xgateway *
```

Appendix C

Virtual Private Networks

This appendix explains how you can use your Gauntlet Internet Firewall to exchange encrypted traffic with other Gauntlet Firewalls.

Note: This feature is only available in the Unites States domestic version of the Gauntlet product.

Packets on the Internet flow through a variety of wires and fibers owned and managed by a variety of organizations. The opportunities for someone or something to monitor these packets are large.

The Gauntlet Internet Firewall can be used to create a Virtual Private Network (VPN). This VPN uses encryption to allow secure communication between various points within this network.

Understanding Virtual Private Networks

When using a single firewall, the defense perimeter includes the network of machines that sit behind the firewall, inside the perimeter. Communication with any other machines or networks outside the perimeter is over some untrusted network, such as the Internet. A Virtual Private Network extends the defense perimeter to include other networks and machines.

For example, Yoyodyne has offices in Maryland and California, each protected by a Gauntlet Internet Firewall. When they communicate, it is via the Internet. Yoyodyne can create a VPN and extend the defense perimeter from its corporate headquarters in Maryland to include the network of machines behind the defense perimeter in its California office, as shown in Figure C-1.

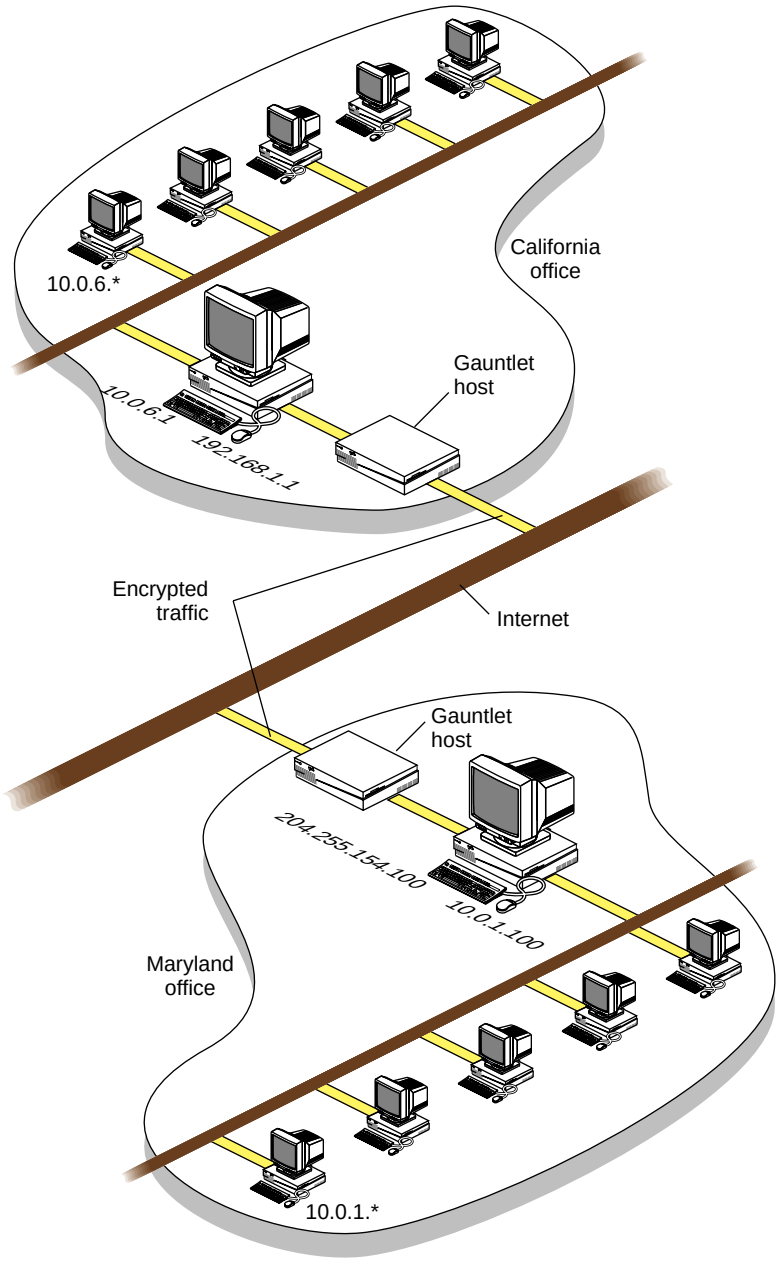

Figure C-1 Yoyodyne Virtual Private Network

A VPN is considered private because all of the traffic that passes through the firewall to another part of the virtual private network, is encrypted. Any program watching the packets flow by would simply see a stream of encrypted data. Without the key used to encrypt the data, snoopers cannot make much use of the information. Because the remote host or network shares a key with the firewall, it can decrypt and process the encrypted packets that it receives. In Figure C-1, all traffic between the firewall in the Maryland office and the firewall in the California office passing over the Internet is encrypted.

A VPN is considered a virtual network because you are extending the network from the machines that are physically within the defense perimeter to include other machines or networks that are not.

Privacy With Trust (Trusted Link)

A VPN with trust expands the concept of trust (as in trusted networks) to include not only the machines within your defense perimeter but also all of the machines within the remote defense perimeter. For all intents and purposes, all of these machines are part of the same network within the same defense perimeter. Any activities that you allow within your network can be used with machines on the remote network.

For example, Yoyodyne allows users in the Maryland office to use the network time protocol (NTP) within the network to set the clocks on their machines. If Yoyodyne sets up a VPN with the California office using privacy with trust, they can now use *ntp* with machines in the California office.

You can create trusted links for host-to-host, network-to-network, or host-to-network communications. This allows you to trust individual hosts or entire networks.

A VPN also allows any IP services you desire to pass between the two firewalls. The services simply need to be IP based. You can allow applications that use the user datagram protocol (UDP) or the transmission control protocol (TCP). You do not need an application proxy.

In addition to sharing a defense perimeter against the rest of the world, sites that create a VPN must share the security perimeter in other ways. These sites should share the same policies, procedures and administrative control. If the security policy for the Maryland office does not allow TELNET from remote locations, then the security policy for the California office should match this. If they differ, someone can simply come in through the California office and then connect directly to a machine in the Maryland office, which is part of the same VPN.

Privacy Without Trust (Private Link)

A VPN without trust does not expand the concept of trust to include the machines within the remote defense perimeter. In this case, the traffic between the two networks is encrypted, providing the privacy. Once it decrypts the traffic, the remote firewall still considers the request as being from an untrusted network. The request is the same as any other that comes from an untrusted network, but with the additional benefit of encryption.

For example, Yoyodyne sets up a VPN without trust between the Maryland and California offices. Traffic between the two offices is still encrypted. When the firewall for the California office receives and decrypts a TELNET request from a machine at the Maryland office, it will treat the request as it would any other untrusted network. They cannot send UDP packets between the two networks, or trust NTP from the other site as they could using a VPN with privacy with trust.

You can create private links for host-to-host, network-to-network, or host-to-network communications. The most common use of privacy without trust creates a private link between two networks.

Sites that create a VPN without trust must of course share the encryption key that gives them the privacy. However, they can now use different policies and procedures and have different administrative control.

Encryption Through Multiple Firewalls (Passthrough Link)

A VPN can use encryption through a series of firewalls. In this case, the traffic between the outer firewalls is encrypted, but the firewalls in between simply pass the encrypted data through. They do not decrypt the data nor do they have the encryption key.

For example, Yoyodyne sets up a VPN (with or without trust) between the firewall for the accounting department in Maryland and the firewall for the accounting department in California. On the firewall for the entire Maryland office (which includes the accounting department), Yoyodyne creates a passthrough link. This link simply passes the encrypted traffic from the accounting firewall in Maryland on to the accounting firewall in California. The administrators in the California office must create a similar passthrough link on their firewall to pass encrypted traffic to the accounting firewall in the California office.

You can create passthrough links for host-to-host, network-to-network, or host-to-network communications. The most common use of a passthrough link specifies a host-to-host link for two firewalls.

How It Works

The Firewall handles VPNs by examining all outbound traffic and encrypting any traffic between hosts that are marked as encrypted peers. The exact sequence of events varies depending on whether there is privacy with trust, or just privacy.

When the firewall is about to send a packet, it checks to see if the source and destination are listed in a table of encrypted pairs. If the source and destination match an entry in the table, the firewall hands the packet to the swIPe driver for encryption.

Encrypting the Data

The swIPe driver uses the Data Encryption Standard (DES) to encrypt the data using the key provided for this VPN during firewall-to-firewall configuration. The new packet contains encrypted data and a header that indicates this is a special encrypted protocol. The firewall then sends the encrypted packets across the Internet (or other untrusted network) to the firewall for the remote network.

When the remote firewall receives the packet on its outside interface, the IP input layer recognizes this as an encrypted packet because of the special protocol. This information indicates that the firewall should send any packets with this special protocol to the swIPe driver.

If the source and destination addresses in the packet indicate that it is part of a passthrough link, the swIPe driver forwards the packet without modifying it.

Decrypting the Data

The swIPe driver decrypts the data using the same key used to encrypt the data. The swIPe driver passes the now decrypted data back to the IP input layer. This now handles the packet as it would handle any other packet that it receives on the outside interface.

Routing the Packet

If the VPN between the two networks uses privacy with trust, the routing layer forwards the packet on to the appropriate host on the inside network. If the VPN between the two networks uses just privacy with no trust, the routing layer hands the packet to the appropriate service or proxy. The proxies treat this packet as they would any other packet from any other untrusted network.

Appendix D

Configuring SSL on the Gauntlet Firewall

Secure Socket Layer (SSL) is a security protocol that can be configured to protect the Gauntlet firewall from security breaches during remote administration sessions. To configure SSL on the firewall, you use the Netscape servers administration utility. Use the online help instructions in the Netscape utility as your primary source of information during the SSL configuration session.

The information in this appendix is supplemental to the Netscape help instructions. If you are already familiar with the SSL configuration procedure, you may not need this supplementary information.

Getting Ready for SSL Configuration

You configure SSL on the Gauntlet firewall using the Netscape administration utility. If you perform this procedure on a host other than the firewall, you can use a Netscape browser to access the firewall after you start the administration utility.

To implement SSL on the firewall, the firewall must contain a digital ID file, also known as a certificate, that identifies it as a trusted server when clients connect to it. Certificates are distributed by a Certification Authority (CA).

Note: If you have not already done so, you should contact a CA, such as Verisign, and request their email address. You will need to supply this address during the SSL configuration procedure.

This procedure explains how to start the Netscape administration utility and go to the SSL configuration pages.

1. Start the Netscape server administration utility.

   You must be the superuser to start the administration utility. (Omit this step if the administration utility is already running):

   ```
   # su
   # /usr/ns-home/start-admin
   ```

type `admin` on both entries lines unless you have changed the defaults.

2. Go to the Netscape Server Selector page.

   This page lets you choose the Netscape server that you want to configure. If you are configuring the firewall from a remote host, use this URL to access the Netscape Server Selector page:

   http://firewall_hostname:81

3. Enter the authentication information in the authentication dialog box.

   The authentication dialogue box requires the user name and password for the Netscape administration utility. The default settings for user name and password are both *admin*; type `admin` on both entry lines unless you have changed the defaults.

4. Choose *Gauntlet* from the Server Selector page.
5. Choose Encryption from the menu bar at the top of the page to configure SSL.

SSL Configuration Procedure

The SSL configuration procedure has of three parts:

- Generating the server's key pair
- Requesting a certificate from a Certification Authority
- Installing the certificate

After you complete parts one and two of the configuration, you will need to wait for the CA to return your certificate by email.Then you can complete part three of the procedure.

Note: Use the Help button on the Encryption configuration pages as the main source of instructions for configuring SSL. Review the supplementary instructions in this appendix before starting each part of the configuration procedure.

Supplementary Instructions for Generating a Key Pair

After the Encryption page is displayed (see step 5 in "Getting Ready for SSL Configuration" on page 275), use the Help button on this page as the main source of instructions for configuring SSL. The instructions below are supplementary:

1. Choose Generate Key from the menu in the sidebar of the Encryption page.

   This choice starts the process of generating a key pair file for encrypting firewall data.

2. Execute the key generation script as the superuser.

   You can run the key generation script from any UNIX shell; you must be the superuser (root) to save the key pair file.

3. Write down the password that you enter in the security key script.

   The password that you enter will be stored in the key pair file. You will be required to enter this password when you do the Request Certificate procedure.

4. Save your entries in the key pair file in the correct location.

   Use this full pathname as the keyfile location when you save it:
   /usr/ns-home/httpd-gauntlet/config/ServerKey.db

Supplementary Instructions for Generating a Certificate

After you generate the key pair (see "Supplementary Instructions for Generating a Key Pair" on page 277), choose Request Certificate to apply for a certificate from your CA. These instructions are supplementary to the instructions provided in the Help screens for the request certificate procedure:

1. Choose Request Certificate from the sidebar menu on the Encryption page.
2. Enter the email address of your CA in the Certificate authority field.

   You should have obtained this address before starting the configuration procedure (see "Getting Ready for SSL Configuration" on page 275).

3. In the Key file password field, enter the password that you wrote down in the generate key procedure.
4. In the Common name field, enter the fully qualified hostname of the firewall.

   For example, *firewall.yoyodyne.com*.

5. In the Email address field, enter the email address of the user who should receive the certificate from the CA when it arrives.
6. Enter the remaining information about your organization in the entry fields that follow.

Saving the Email Reply from Your Certificate Authority

After your certificate request is granted, you will receive a certificate by return email. Save this certificate in a file called */usr/ns-home/httpd/gauntlet/config/certificate*.

Supplementary Instructions for Installing Your Certificate

After your certificate arrives and you save it in a file, complete the certificate installation procedure and turn encryption on for the firewall. These instructions are supplementary to the instructions provided in the Help screens for the install certificate procedure:

1. Choose Request Certificate from the sidebar menu on the Encryption page.
2. In the Certificate Name field, enter the fully qualified hostname of the Gauntlet firewall.

   For example, *firewall.yoyodyne.com*.
3. In the Message in this file field, enter the complete pathname of the certificate file.

   The pathname of the certificate file is */usr/ns-home/httpd-gauntlet/config/certificate*.
4. Leave the Message text field blank.
5. Returned to the Encryption page.
6. Click Encryption to turn encryption on for the firewall.

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-2826-004.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 415-965-0964
- To send your comments by **traditional mail**, use this address:

  Technical Publications
  Silicon Graphics, Inc.
  2011 North Shoreline Boulevard, M/S 535
  Mountain View, California 94043-1389