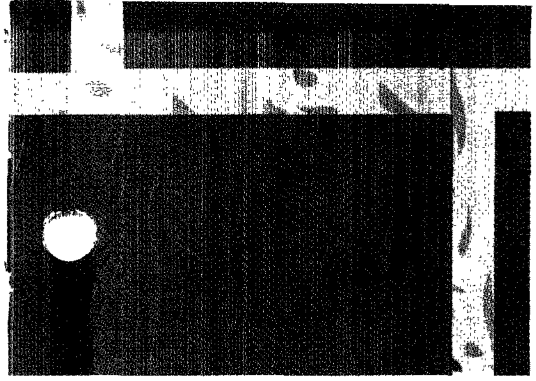
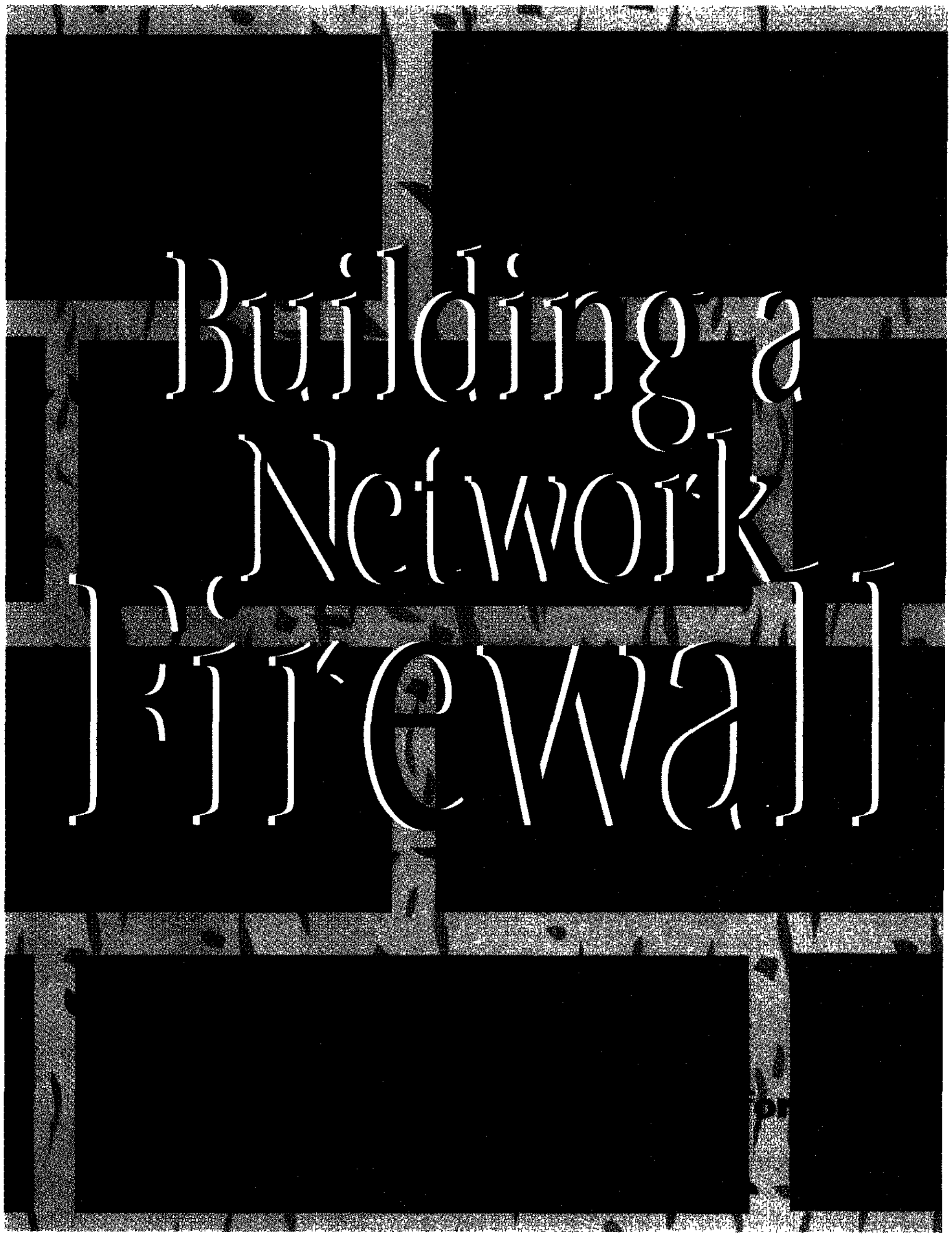


# Building a Network Firewall

**LAN security is weakest at the point of entry. A gate and choke can give you protection.**



*With the recent publication of the page-turning saga* Cyberpunk: Outlaws and Hackers on the Computer Frontier, *by Katie Hafner and John Markoff, the computer industry has been keenly reminded of the vulnerability of networks and the damage that unauthorized users can cause. As the book dramatically points out, Unix networks and LANs linked to them are susceptible to security breaches. This article describes specific techniques you can use to protect your Unix LAN from costly invasions. The article has been adapted from another book,* Practical Unix Security, *(1991), by Simson Garfinkel and Gene Spafford, O'Reilly & Associates Inc., Sebastopol, Calif.; 800-338-6887. — Editors*

When apartment houses or office buildings are built, they are often equipped with firewalls — specially constructed walls that are resistant to fire. If a fire should start in the building, it may burn out of control in one portion, but the firewall will stop or slow the progress of the fire until help arrives.

The same philosophy can be applied to the protection of local area networks from outside attack. On networks, firewalls make it difficult for attackers to jump from network to network. Installation of firewall machines can help stop or reduce malicious damage and intrusion.

## Internal and External Firewalls

The simplest approach to firewalls is to keep your local networks small and independent. Once an intruder compromises one machine on a network, it's often trivial for him or her to compromise others. The task of compromising these systems is often made simpler by having all the machines at a site on the same physical and logical networks.

With the Network Information System (NIS) in Unix, a server carries the network addresses of all the machines

ILLUS N AND DESIGN BY MARGARET A. ANDERSON

on the LAN in a simple lookup directory, as well as information about the files exported via the Network File System (NFS). This exported file information can display who has read and write access, which files are protected, and which are not. NIS and NFS deliver powerful networking capabilities, but, undefended, are ripe for invasion from unwanted intruders.

There are effective ways to protect your network, your users, and their files. To start, instead of putting all your machines on one network, separate your installation into sets of LANs communicating through gateway machines or routers. To accomplish this, follow these guidelines:

1. If you use NIS, each local area network should have its own server. Each server and its clients should have their own group domain.
2. No server or workstation on one network should trust hosts in any other network (or any gateway machine).
3. Users who have accounts on more than one LAN should have different passwords for each subnet, and should *not* have **.rhost** files to allow access between local networks without providing a password. (.rhost files describe trusted machines that are allowed access without requiring a password.)
4. The gateways should have the highest level of logging enabled, and the most restrictive security possible. If possible, do not allow user accounts on the gateway machines.
5. Do not NFS-mount file systems from one LAN onto another LAN.

Internal firewall machines have many benefits. For example:

- They help isolate physical failure of the network to a smaller number of machines.
- They limit the number of machines putting information on any physical segment of the network, thus limiting the damage that can be done by eavesdropping.
- They limit the number of machines that will be affected by flooding attacks, which are a bombardment of packets on a network (generated by, for example, Telnet requests) that can potentially bring the network down.
- They create barriers for attackers, both external and internal, who are trying to break in to specific machines at a particular installation.

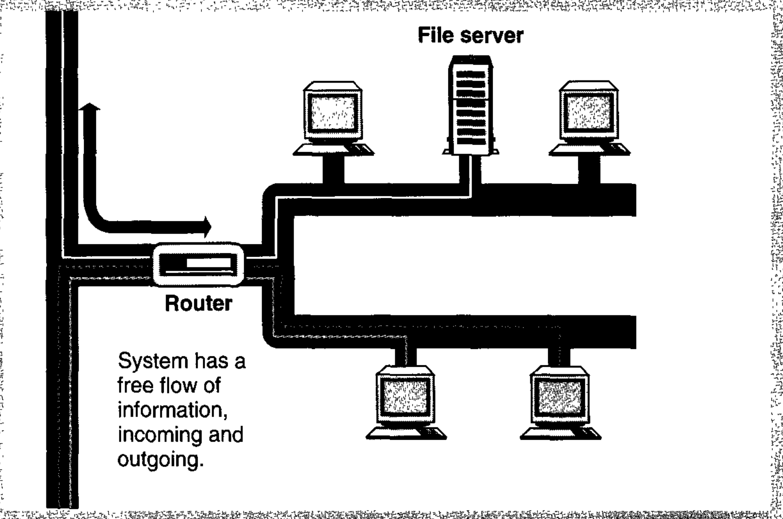

File server
Router
System has a free flow of information, incoming and outgoing.

**Figure 1. Today's networks can make it easy to connect to the outside—and vice versa.**

In addition to partitioning the network to slow or stop intrusion, it's important to install an external firewall. This is a machine (or set of machines) that puts up a wall between your local installation and the outside world. It should be configured to allow certain operations to occur, such as E-mail delivery, but to make it difficult or impossible for an attacker on the outside to use the firewall to penetrate your internal networks.

Today, most corporate and academic networks connect to the outside world with simple routers or bridges, as shown in Figure 1. This makes it possible for any workstation on the internal network to reach the outside — and for a computer on the outside to reach in and connect to any workstation. Sometimes, servers at sites are equipped with two network interfaces and use the same machine as a file server and a gateway, as shown in Figure 2.

A firewall consists of two parts that separate the outside network from the internal one: a gate and a choke. A gate passes data between the two networks. A choke blocks all packets from the outside network destined for the inside network, unless they are destined for the gate, and blocks all packets from the inside network destined for the outside, unless they originated from the gate.

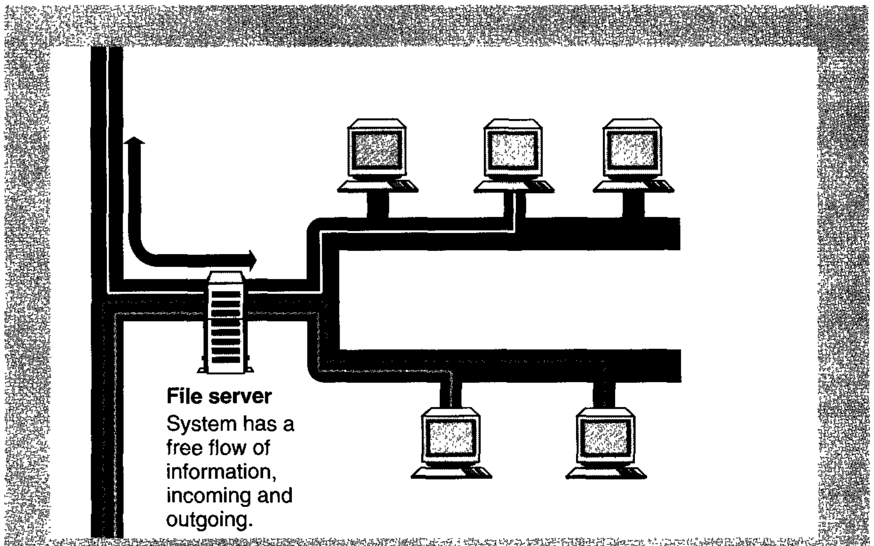

File server
System has a free flow of information, incoming and outgoing.

**Figure 2. A file server can be equipped with two network interfaces and provide gateway services as well.**

The choke and gate can be the same computer, or they can be two different machines. Similarly, the gate can be one computer, or a number of different computers, one for each protocol used on the network.

Multiple gates sometimes add a small measure of additional security to the configuration, but they also add to the delay and difficulty involved with authorized network communication. For simplicity, the following sections assume that there is a single choke and a single gate, as shown in Figure 3.

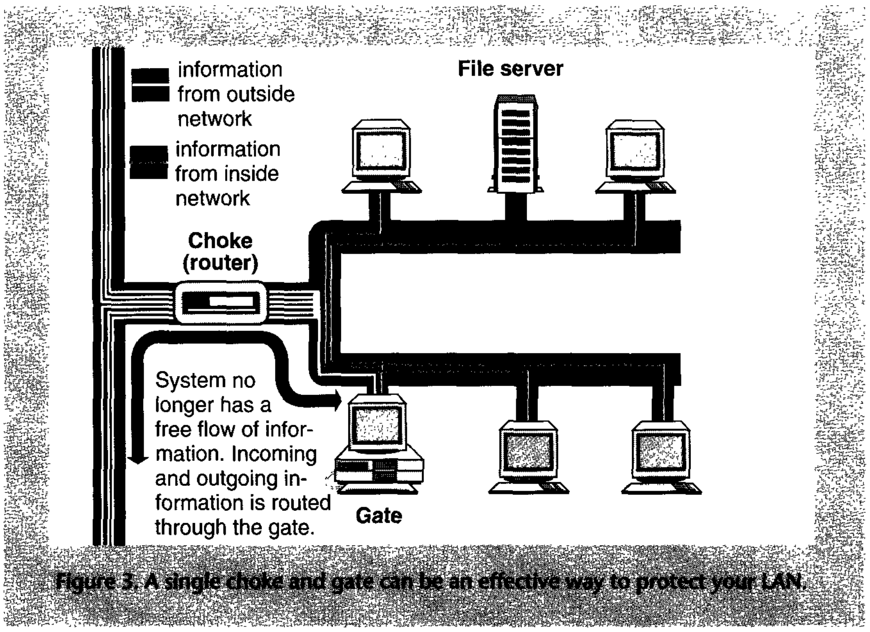

information from outside network
information from inside network
File server
Choke (router)
System no longer has a free flow of information. Incoming and outgoing information is routed through the gate.
Gate

Figure 3. A single choke and gate can be an effective way to protect your LAN.

To set up a firewall you will need to set up both the choke and the gate. Usually, both the choke and the gate will be a single Unix computer with two network interfaces, specifically set up so that it does not forward packets from one network to the other. However, you may wish to separate the choke and the gate for increased security and control.

The choke is the bridge between the inside network and the outside network. It does not forward packets between the two networks unless the packets have the gate computer as either their destination or their origination address. You can optionally set up the choke so it forwards only packets for particular protocols — for example, packets used for mail transfer, but not for **telnet** or **rlogin**.

There are three main ways to set up a choke:

1. Use a standard Unix computer with two network interfaces. Do not run the program **/usr/etc/routed** (the network routing daemon) on this computer. Set the computer up so it does not forward packets from one network interface to the other. A computer set up in this fashion is both the choke and the gate.
2. Use an "intelligent router." Many of these routers can be used to forward only certain kinds of packets and only between certain addresses.
3. If you have access to the source code for your version of Unix, alter the gate computer's network driver. Some vendors may offer Unix operating systems that can be configured with this feature in the not-too-distant future. This modification is not recommended for the average site, however, because it is difficult and the chance of error is large.

The details of how you set up your choke will vary greatly, depending on the hardware you use and that hardware's software. The following sections, therefore, are only general guidelines.

## Choosing Choke Protocols

The choke is an intelligent filter. It makes certain that only the gate machine can talk to the outside world. All messages from the outside (whether they're simple E-mail or ingenious attempts to break in) that are directed to internal machines other than the gate are rejected. Attempts by local machines to contact sites outside the LAN are similarly denied.

The gate determines destinations, then handles requests or forwards them as appropriate. For instance, Simple Mail Transfer Protocol (SMTP) requests may be sent to the gate, which resolves local aliases and then sends the E-mail to the appropriate internal machine. Furthermore, you can set up your choke so that only specific kinds of messages are sent through. The way you configure your choke, however, will depend on the particular router that you are using for a choke; consult your router's documentation for detail.

You should configure the choke to reject messages that use unknown protocols. You may also wish to configure the choke to specifically reject known protocols that are too dangerous because of the ease by which intruders can abuse them:

- **tftp**
- **sunrpc**
- **printer**
- **rlogin**
- **rexec**

Protocols that you might want to allow through the choke to the gate include:

- **telnet**
- **ftp**
- **SMTP**
- **name**
- **time**
- **domain**
- **NNTP**

(The **finger** protocol is problematic, and will be discussed later.)

The choke also prevents local users from connecting to the outside machines through unrestricted channels. This prevents Trojan horse programs from installing network back doors on your local machines. (Imagine a public-domain data analysis program that surreptitiously listens on Port 49372 for connections and then creates an unauthorized shell through **/bin/csh** for the intruder to use.) It also makes it difficult for someone who does manage to penetrate one of your local machines to send information back to the outside world.

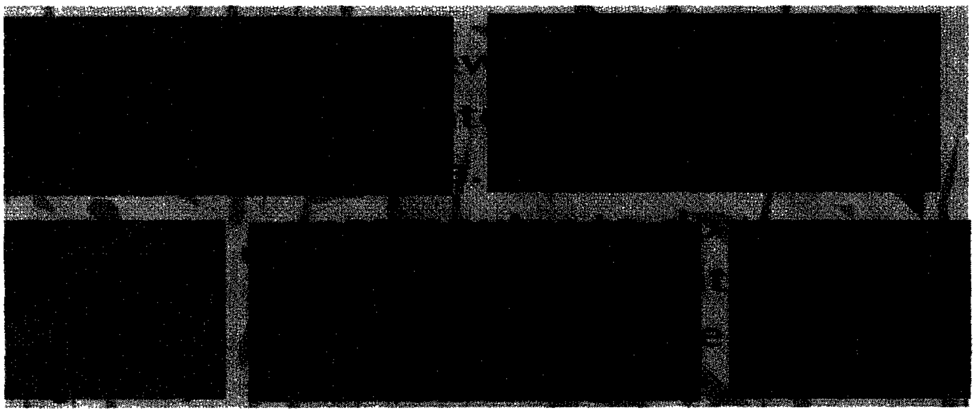
There should be no way to change your choke's configuration from the network. An attacker trying to tap into your network will be stuck if your choke is a personal computer–based router that can be reprogrammed only from its keyboard.

## Setting Up the Gate

The gate machine is the other half of the firewall. The choke forces all communication between the inside network and the outside network to take place through the gate. The gate enforces security, authenticates users, sanitizes data (if necessary), and passes the data along.

The gate should have a very stripped-down version of your operating system. It should not have any compilers, to prevent attackers from compiling programs on it. It should have no regular user accounts, to limit the places an outsider can enter.

You can concentrate most of your security effort on setting up and maintaining the gate. Usually, the gate will act as your E-mail server, your Usenet server (if you support the worldwide network news bulletin board available to Unix machines), and your anonymous File Transfer Protocol (FTP) repository (if you maintain one). *It should not be your file server.* We'll discuss how you configure each of these services, and then how to protect the gate. For these examples, we use a hypothetical domain called **company.com**. We've named the gate machine **keeper.company.com** and an internal user machine **office.company.com.**



Either the choke or the gate must provide Internet Domain Name Service (DNS) to the outside network for the **company.com** domain. Usually, you will do this by running the name server from UC Berkeley's version of Unix on one of these machines.

Occasionally, the names of computers on your internal network will be sent outside; your name server should be set up so that when people on the outside try to send E-mail back to the internal computers, it is sent to the gate instead. The simplest way to do this is with a name server MX record. An MX record causes E-mail destined for one machine to be sent to another.

Configure your name server on the gate so that there is an MX record for every computer on the inside net, each pointing to the gate. For example, the MX record for **office.company.com** might look like this:

```
office.company.com IN HINFO
NEXT MACH ; 604800
office.company.com IN MX 10
KEEPER.COMPANY.COM ; 604800
```

This way, people on the outside network will be able to reply to any E-mail that escapes with an internal name.

Configure the gate so all outgoing E-mail appears to come from the gate machine. That is:

- All E-mail messages sent from the inside network must have the **To:**, **From:**, and **cc:** fields of their headers rewritten so an address in the form **user@office.company.com** is translated to the form **user@company.com.**
- Because all E-mail from the outside is sent through the gate, the gate must have a full set of E-mail aliases to allow E-mail to be redirected to the appropriate internal site and user.
- E-mail on the internal machines, like **office**, must have their mailers configured so that all E-mail not destined for an internal machine (anything not to a **company.com** machine) is sent to the gate, where the message's headers will be rewritten, and then forwarded through the choke to the external network.
- All uucp E-mail must be run from the gate machine. All outgoing uucp messages must have their return paths rewritten from **company!office!user** to **company!user.**

There are many advantages to configuring your E-mail system with a central "post office."

- Only one machine has to have a complex mailer configuration.
- Only one machine needs to handle automatic **uucp** path routing.
- Only one machine needs to have a complete set of user aliases in place.
- If a user changes the name of his or her computer, that change needs to be made only on the gate machine. Nobody in the outside world, including electronic correspondents, needs to update his or her information; the change can easily be installed by the administrator at the gate machine.
- You can use aliases on user accounts: all E-mail sent off site can have first-name_lastname in its mail header.
- If a user leaves the organization and needs to have his or her E-mail forwarded, E-mail forwarding can be done on the gate machine. This eliminates the need to leave old accounts in place after someone has left the company simply to allow a **.forward** file to point at his or her new address.

Configure Netnews (the connection software to the bulletin board system maintained on Usenet) so the gate machine is the main news machine in the organization. Use the following procedures:

- All outgoing articles must have the **Path:** and **From:** lines set to show only the gate machine. This is not difficult to do if the news is present only on the gate machine — the B News software provides definitions in the configuration file to build the headers this way.
- Internally, news can be read with **NNTP** and **rrn.**
- Alternatively, the news spool directory (usually **/usr/spool/news**) may be exported read-only by the gate machine to the internal machines. Posting internally would still be via **NNTP** and **inews.**

Again, there are advantages to this configuration beyond the security considerations. One benefit is that news is maintained on a central machine, thus simplifying maintenance and storage considerations. Furthermore, it is easier to regulate local-only groups because the gate machine can be set to prevent local groups from being sent outside. The administrator can also regulate which internal machines are allowed to read and post news.

### Safe, Anonymous FTP

If you wish to use anonymous FTP from the outside network, make sure the ˜**ftp/pub** directory resides on the gate machine. FTP lets you transfer complete files between systems. When you log in to the remote machine to transfer a file, the login and password codes can be intercepted as they are transmitted across the network. To prevent this, set up FTP to accept anonymous as your user name and your real identity as the password.

Internal users can access the ˜**ftp/pub** directory via NFS. By leaving files in this directory, internal users can make their files available to users on the outside. Users from the outside use FTP to connect to the gate computer to read and write files.

To make it possible for internal users to use FTP to transfer files from remote sites, create a special account on the gate machine named **ftpout**. Internal users connect via Telnet to the gate and log in as **ftpout**. (The Telnet protocol uses the **telnet** or **telnetd** Unix programs to let LAN users log in to a computer and use it as though they were connected via a directly attached terminal.) Only logins from internal machines should be allowed into this account.

The **ftpout** account is a special account constructed for the purpose of using the **ftp** program. If you want added security, you can even set this account shell to be the **/usr/ucb/ftp** program. When users wish to transfer files from the outside, they will **rlogin** to the **ftpout** account on the gate, use FTP to transfer the files to the gate, log out of the gate computer, and then use NFS to read the files from the gate. The **ftpout** account should have a UID (user id) that is different from every other user on the system — including the **ftp** user.

There are a number of different ways that you can protect the **ftpout** account from unauthorized use. One simple approach follows:

1. Create the **ftpout** account on the gate with an asterisk (*) for a password (this prevents logins).
2. Make the **ftpout** account's home directory owned by **root**, known as mode 755.
3. Create a file ˜**ftp/.rhosts**, owned by root, that contains a list of the local users who are allowed to use the **ftpout** service.

Legitimate users can now use **ftpout** by using the **rlogin** command:

```
#rlogin gate -l ftpout
```

The **ftpout** account must log all uses, via **syslog**, console prints, or similar means. It must then run the **ftp** program to allow the user to connect out to remote machines and transfer files locally to the gate.

This configuration lets users import or export files, but it never makes a continuous FTP connection between internal and external machines. The configuration lets you keep a central repository of documents transferred via FTP, possibly with disk quotas, which are the maximum blocks of data storage space allocated to a user. This also saves on mass storage.

### Other Services

In addition to news, E-mail, and FTP, users will want to be able to **rlogin** to machines outside the local environment. You can use a scheme exactly like the one described above for FTP to let local users use Telnet with remote sites. Do not use the same UID and

group for the **telnetout** account that you used for the **ftp** command.

Many sites using gates disable the **finger** service, because **finger**, which identifies active users on a computer, often provides too much information to outsiders about your internal file system structure and account naming conventions. Unfortunately, the **finger** command provides very useful information, and disabling its operation at a large site may result in considerable frustration for legitimate outside users.

As an alternative, you can modify the **finger** service to provide a limited server that will respond with a user's mailbox name, and, optionally, other information such as phone number and whether or not the user is currently logged in. The output should not provide the home directory or the true account name to the outside, although this is not critical if the gate is otherwise well configured.

You can create additional accounts, similar to **ftpout**, for users who wish to **finger** people on the outside. Alternatively, you can create your own dedicated servers on the gate for passing this information along.

The biggest difficulty with firewall machines comes when a user is off-site and wishes to log in to his or her account on the network. After all, remote logins are exactly what a gate is designed to prevent! If such logins are infrequent, you can create a temporary account on the gate with a random name and random password that cannot be changed by the user. The account does not have a shell, but instead executes a shell script that does an **rlogin** to the user's real account. The user must not be allowed to change the password on this gate account, and is forbidden from installing the account name in his or her local **.rhosts** file. For added security, be sure to delete the account after a fixed period of time — preferably a matter of weeks.

If there are many remote users, or users who will be doing remote logins on a continuing basis, the above method will work but is unlikely to be acceptable to most users. In such a case, we recommend using the setup described above, with two changes: Let users pick a gate account name that is more mnemonic, and force them to use some type of higher-security access device, such as a smartcard ID, to access the gate. If passwords must be used on the gate accounts, be sure to age them frequently (once every two to four weeks), and let the machine generate the passwords to prevent users from setting the same password as their internal accounts.

## Special Considerations

To make the firewall setup effective, the gate should be a pain to use: Really, all you want this computer to do is forward specific kinds of information across the choke. The gate should be as impervious as possible to security threats. The list that follows summarizes configuration considerations you may want to make on the gate machine:

- No regular user accounts. Only accounts for people requiring incoming connections, system accounts for needed services, and the **root** account.
- No imported directories from NFS or RFS. Export only directories with data files (such as **ftp/pub** and news).
- Remove or rename the binaries of all commands not necessary for gate operation. This includes tools such as **cc**, **awk**, **sed**, **ld**, and **emacs**. Remove all libraries from **/usr/lib** and **/lib**. Program development for the gate can be done on another machine and copied to a gate machine; with program development tools and unnecessary commands removed, a cracker can't easily install Trojan horses or other nasty code. Rename or move all the user shells (so the ! command in **ftp** and **telnet** do not give a user interactive access). If you really don't want to remove these programs, **chmod** them from 755 to 500. The **root** user will still be able to use these programs, but no one else will. This is not as secure as removing these programs, but it is more effective than leaving the tools in place.
- **chmod** all system directories (such as /, **/usr**, **/usr/bin**, **/etc**, and **/usr/spool**) to mode 711. Users of the system (other than the superuser) do not need to list directory contents to see what is and is not present. This will really slow down someone who manages to establish a non-root shell on the machine through some other mechanism.
- Don't run NIS on the gate machine. Do not import or export NIS files, especially the **alias** and **passwd** files.
- Turn on full logging on the gate machine. Read the logs regularly. Set the **syslog.conf** file so that the gate logs to an internal machine as well as a hardcopy device, if possible.
- Mount as many disks as possible read-only. This prevents a cracker from modifying the files on those disks. Some directories, notably **/usr/spool/uucp**, **/usr/adm**, and **˜ftp/pub**, will need to be writable. You can place all of these directories on a single partition and use symbolic links so that they appear in the appropriate place.
- Turn on process and file quotas, if available.
- Use some form of smart-card or key-based access for the **root** user. Otherwise, don't allow anyone to log in as **root** on the machine or to **su** — set things up so users can **rlogin** to **root** only from designated accounts (other **root** accounts, for instance) from internal machines.
- Make the gate computer "equivalent" to no other machine. Remove the files **/etc/hosts.equiv** and **/etc/hosts/ipd**.
- Disable all unneeded network services.

When you configure your gate machine, remember that every service and program that can be run presents a threat to the security of your entire protected network. The purpose of the gate is to restrict access to your network, not to serve as a computing platform. Therefore, remove everything that's not essential to the network services.

Be certain to monitor your gate on a regular, scheduled basis. If you just set it up and forget about it, it may take you weeks or longer to discover a break-in.

Even if you follow all of these rules and closely monitor your gate, it may still be possible for a group of very persistent and clever crackers to break through to your machines. If they do, it's unlikely that the cause will be accidental. They will have to work hard at it, and you will likely find evidence of the break-in soon after it occurs. The steps we've outlined will probably discourage the random or curious cracker, as well as many more serious intruders, and that is really your goal. ■

*Simson Garfinkel is a senior editor at NeXTWorld magazine, on leave from the doctoral program at MIT's Media Lab. Gene Spafford is on the faculty of the Department of Computer Sciences at Purdue University and is associated with Purdue's Software Engineering Research Center (SERC).*