

Protecting Your Network

The Network+ Certification exam expects you to know how to

- 2.17 Identify the following security protocols and describe their purpose and function: IPSec, SSL
- 3.5 Identify the purpose, benefits, and characteristics of using a firewall
- 3.6 Identify the purpose, benefits, and characteristics of using a proxy service
- 3.7 Given a connectivity scenario, determine the impact on network functionality of a particular security implementation (for example, port blocking/filtering, authentication, encryption)
- 3.8 Identify the main characteristics of VLANs (Virtual Local Area Networks)
- 3.9 Identify the main characteristics and purpose of extranets and intranets

To achieve these goals, you must be able to

- Define the various types of network threats and how they are caused
- Explain how firewalls, NAT, port filtering, packet filtering, encryption, and authentication protect a network from threats
- Explain how to implement these levels of protection on different types of networks

The very nature of networking makes networks vulnerable to a dizzying array of threats. By definition, a network must allow for multiple users to access serving systems, but at the same time we must protect the network from harm. Who are the people causing this harm?

The news may be full of tales about *hackers* and other malicious people with nothing better to do than lurk around the Internet and trash the peace-loving systems of good folks like us, but in reality hackers are only one of many serious *network threats*. You will learn how to protect your networks from hackers, but first I want you to appreciate that the average network faces plenty of threats from the folks who are *authorized* to use it! Users with good intentions are far more likely to cause you trouble than any hacker. So, the first order of business is to stop and think about the types of threats that face the average network. After we define the threats, we can discuss the many tools and methods used to protect our precious networks from intentional harm.



NOTE Be aware that in some circles, the term “hacker” describes folks who love the challenge of overcoming obstacles and perceived limitations—and that’s a positive thing! To distinguish these good hackers from the bad guys that we hear so much about, folks who consider themselves good hackers have coined the term “cracker” to describe the bad guys who abuse computer systems. Sadly, the mainstream press continues to use the term “hacker” instead of “cracker,” and we will follow that common usage going forward.

Historical/Conceptual

Defining Network Threats

What is a threat? What makes something bad for our network? In my opinion, anything that prevents users from accessing the resources they need to get work done is a threat. Clearly that includes the evil hacker who reformats the server’s hard drive, but it also includes things like bad configurations, screwed up permissions, viruses, and unintentional corruption of data by users. To make the security task more manageable, I like to sort these possibilities into two groups: internal threats and external threats.

Internal Threats

Internal threats are all the things our own users do to networks to keep them from sharing resources properly. Internal threats may not be as intriguing as external threats, but they are far more likely to bring a network to its knees, and they’re the ones we need to be most vigilant to prevent. Here are the most common internal threats:

- Unauthorized access
- Data destruction
- Administrative access
- System crash/hardware failure
- Virus

Let’s look at each one of these threats in turn.

Unauthorized Access

The most common of all network threats, unauthorized access, occurs when a user accesses resources in an unauthorized way. The unauthorized access itself does no actual damage to data; the person is usually just accessing data in a way that he or she shouldn’t—such as reading employee personnel files or notes from the last board of directors meeting. Not all unauthorized access is malicious—usually this problem arises when users who are randomly poking around in the network discover that they can access resources in a fashion the administrators did not intend. Once a user has unauthorized access to a

resource, they might just see more than they should; or worse, it can lead to data destruction. Our job is to protect these users from themselves.

Data Destruction

An extension of unauthorized access, accidental data destruction means more than just intentionally or accidentally erasing or corrupting data. Consider the case where users are authorized to access certain data, but what they do to that data goes beyond what they are authorized to do. A good example is the person who legitimately accesses a Microsoft Access product database to modify the product descriptions, only to discover he can change the prices of the products, too. This type of threat is particularly dangerous where users are not clearly informed about the extent to which they are authorized to make changes. A fellow tech once told me about a user who managed to mangle an important database due to someone giving them incorrect access. When confronted, the user said: “If I wasn’t allowed to change it, the system wouldn’t let me do it!” Many users believe that systems are configured in a paternalistic way that wouldn’t allow them to do anything inappropriate. As a result, users will often assume they’re authorized to make any changes they believe are necessary when working on a piece of data they know they’re authorized to access.

Administrative Access

Throughout this book you’ve seen that every *network operating system* (NOS) is packed with administrative tools and functionality. We need these tools to get all kinds of work done, but by the same token we need to work hard to keep these capabilities out of the reach of those who don’t need them. Clearly giving regular users Administrator/Supervisor/root access is a bad idea, but far more subtle problems can arise. I once gave a user Manage Documents permission for a busy laser printer in a Windows 2003 network. She quickly realized she could pause other users’ print jobs and send her print jobs to the beginning of the print queue—nice for her but not so nice for her coworkers. Protecting administrative programs and functions from access and abuse by users is a real challenge, and one that requires an extensive knowledge of the NOS and of users’ motivations.

System Crash/Hardware Failure

Like any technology, computers can and will fail—usually when you can least afford for it to happen. Hard drives crash, servers lock up, the power fails—it’s all part of the joy of working in the networking business. We need to create redundancy in areas prone to failure (like installing backup power in case of electrical failure) and perform those all-important data backups. Chapter 19, “The Perfect Server,” goes into detail about these and other issues involved in creating a stable and reliable server.

Virus

Networks are without a doubt the fastest and most efficient vehicles for transferring computer viruses among systems. News reports focus attention on the many virus attacks from the Internet, but a huge number of viruses still come from users who bring in programs on floppy disks, writeable CDs, and USB drives. We could treat viruses as an exter-

nal threat as well, but instead of repeating myself, I'm going to cover internal and external issues of computer virus protection in Chapter 19, "The Perfect Server," including both the various methods of virus infection, and what you need to do to prevent virus infection of your networked systems.

External Threats

External threats come in two different forms. First, an outsider can manipulate your people to gain access to your network, a process called *social engineering*. Second, a hacker at a remote location can use technical exploits of your network to gain access. The mechanics of gaining access differs dramatically between the two threats, but both result in the same problems for you. Let's take a look.

Social Engineering

The vast majority of attacks against your network come under the heading of *social engineering*—the process of using or manipulating people inside the networking environment to gain access to that network from the outside. The term "social engineering" covers the many ways humans can use other humans to gain unauthorized information. This unauthorized information may be a network login, a credit card number, company customer data—almost anything you might imagine that one person or organization may not want a person outside of that organization to access.

Social engineering attacks aren't hacking—at least in the classic sense of the word—although the goals are the same. Social engineering is where people attack an organization through the people in the organization or physically access the organization to get the information they need. Here are a few of the more classic types of social engineering attacks.



NOTE It's common for these attacks to be used together, so if you discover one of them being used against your organization, it's a good idea to look for others.

Infiltration Hackers can physically enter your building under the guise of someone who might have legitimate reason for being there, such as cleaning personnel, repair technicians, or messengers. They then snoop around desks, looking for whatever they can. They might talk with people inside the organization, gathering names, office numbers, department names—little things in and of themselves, but powerful tools when combined later with other social engineering attacks.

Telephone Scams *Telephone scams* are probably the most common social engineering attacks. In this case the attacker makes a phone call to someone in the organization to gain information. The attacker attempts to come across as someone in the organization and uses this to get the desired information. Probably one of the most famous of all of these scams is the "I forgot my user name and password" scam. In this gambit, the attacker first learns the account name of a legitimate person in the organization, usually

using the infiltration method. The attacker then calls someone in the organization, usually the help desk, in an attempt to gather information, in this case a password.

Hacker: "Hi, this is John Anderson in accounting. I forgot my password. Can you reset it please?"

Help Desk: "Sure, what's your user name?"

Hacker: "j_w_Anderson"

Help Desk: "OK, I reset it to e34rd3"

Certainly telephone scams aren't limited to attempts to get network access. There are documented telephone scams against organizations aimed at getting cash, blackmail material, or other valuables.

Dumpster Diving *Dumpster diving* is the generic term for anytime a hacker goes through your refuse, looking for information. The amount of sensitive information that makes it into any organization's trash bin boggles the mind! Years ago, I worked with an IT security guru who gave me and a few other IT people a tour of our office's trash. In one 20-minute tour of the personal wastebaskets of one office area, we had enough information to access the network easily, as well as to seriously embarrass more than a few people. When it comes to getting information, trash is the place to look!

Physical Theft I once had a fellow network geek challenge me to try to bring down his newly installed network. He had just installed a powerful and expensive *firewall* router and was convinced that I couldn't get to a test server he added to his network just for me to try to access. After a few attempts to hack in over the Internet, I saw that I wasn't going to get anywhere that way. So I jumped in my car and drove to his office, having first outfitted myself in a techy looking jumpsuit and an ancient ID badge I just happened to have in my sock drawer. I smiled sweetly at the receptionist, and walked right by my friend's office (I noticed he was smugly monitoring incoming IP traffic using some neat packet-sniffing program) to his new server. I quickly pulled the wires out of the back of his precious server, picked it up, and walked out the door. The receptionist was too busy trying to figure out why her e-mail wasn't working to notice me as I whisked by her carrying the 65-pound server box. I stopped in the hall and called him from my cell phone.

Me (cheerily): "Dude, I got all your data!"

Him (not cheerily): "You rebooted my server! How did you do it?"

Me (smiling): "I didn't reboot it—go over and look at it!"

Him (really mad now): "YOU <EXPLETIVE> THIEF! YOU STOLE MY SERVER!"

Me (cordially): "Why, yes. Yes, I did. Give me two days to hack your password in the comfort of my home, and I'll see everything! Bye!"

I immediately walked back in and handed him back the test server. It was fun. The moral here is simple—never forget that the best network software security measures can be rendered useless if you fail to physically protect your systems!

Hacking

Ah, here's the part I know you want to talk about—those infamous network threats from outside, the lawless hacker working out of his basement somewhere on another continent, using satellite uplinks to punch into networks using sophisticated Internet worms and other arcane geek weapons. Given Hollywood's influence from popular high-tech movies, many people assume that hacking is a sexy, exciting business, full of suspense and beautiful people. I hate to break this to those of you inclined to such a view, but the world of hackers is a pathetic sideshow of punk kids, Internet newbies, and a few otherwise normal folks with some extra networking knowledge who for one reason or another find a motivation to try to get into areas of public and private networks where they have no business. Hacking isn't sexy—it's a felony.

The secret to preventing hacking is to understand the motivations of hackers. I divide hackers into four groups, each with different motivations: inspectors, interceptors, controllers, and flooders.

Inspector

An *inspector* is a person who wants to poke around on your serving systems like a regular user. This person looks for vulnerabilities in your Internet access, permissions, passwords, and other methods to gain access to your network. The inspector's motivation ranges from the casual—a person who notices open doors into your network—to the serious—hackers looking for specific data. This is the type of hacker most of us visualize when we think of hacking.

Interceptor

An *interceptor* doesn't try to hack into systems. This person just monitors your network traffic looking for intercept information. Once an interceptor finds the traffic he wants, he may read or redirect the traffic for a number of nefarious purposes, such as the classic "man in the middle" attack. (In this type of attack, the hacker changes intercepted data to make it appear that he is one of the people who's supposed to be in on the conversation.) The interceptor is often collecting passwords for later invasion of a network.

Controller

A *controller* wants to take and keep control of one particular aspect of your system. One of the controller's favorite gambits is taking control of SMTP servers and using them for other purposes—usually spamming. Other popular targets are FTP and web servers. Possibly the most nefarious of all controller type attacks are known as *zombie attacks*. To launch a *zombie attack*, a hacker infects a large number of systems with a Trojan horse of some type. The bad guy then uses these systems to perform a large-scale attack on other systems, making it difficult if not impossible to trace the attack to the bad guy.

Flooder

Flooding attacks, more commonly called *denial of service (DoS) attacks*, are the work of hackers whose only interest is in bringing a network to its knees. This is accomplished by

flooding the network with so many requests that it becomes overwhelmed and ceases functioning. These attacks are most commonly performed on web sites and mail servers, but virtually any part of a network can be attacked via some DoS method. The zombie attack I mentioned earlier is a common type of flooding.

Test Specific

Protecting from Internal Threats

The vast majority of protective strategies related to internal threats are based on policies rather than technology. Even the smallest network will have a number of user accounts and groups scattered about with different levels of rights/permissions. Every time you give a user access to a resource, you create potential loopholes that can leave your network vulnerable to unauthorized accesses, data destruction, and other administrative nightmares. To protect your network from internal threats, you need to implement the right controls over passwords, user accounts, permissions, and policies. Let's start with probably the most abused of all these areas: passwords.

Passwords

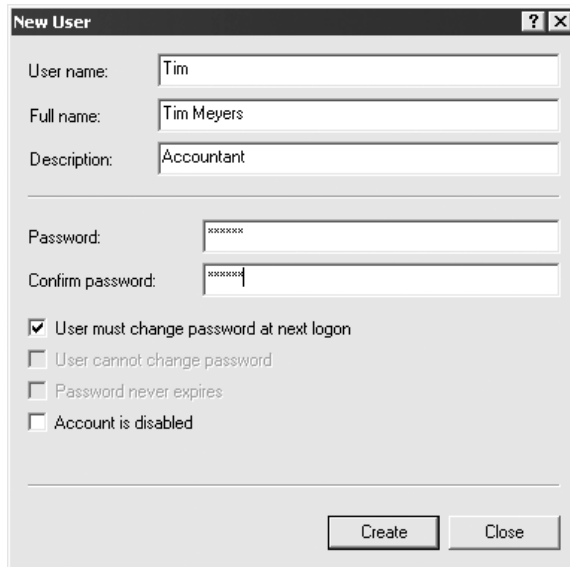
Passwords are the ultimate key to protecting your network. A user account with a valid password will get you into any system. Even if the user account only has limited permissions, you still have a security breach. Remember: for a hacker, just getting into the network is half the battle.

Protect your passwords. Never give out passwords over the phone. If a user loses a password, an administrator should reset the password to a complex combination of letters and numbers, and then allow the user to change the password to something they want. All of the stronger network operating systems have this capability. Windows 2000 Server, for example, provides a setting called *User must change password at next logon*, as shown in Figure 17-1.

Make your users choose good passwords. I once attended a network security seminar, and the speaker had everyone stand up. She then began to ask questions about our passwords—if we responded positively to the question we were to sit down. She began to ask questions like “Do you use the name of your spouse as a password?” and “Do you use your pet's name?”

By the time she was done asking about 15 questions, only 6 people out of some 300 were still standing! The reality is that most of us choose passwords that are amazingly easy to hack. Make sure you use strong passwords: at least six to eight characters in length, including letters, numbers, and punctuation symbols.

Figure 17-1
Windows 2000
Server's User
must change
password at next
logon setting



TIP Using non-alphanumeric characters makes any password much more difficult to crack for two reasons. First, adding non-alphanumeric characters forces the hacker to consider many more possible characters than just letters and numbers. Second, most password crackers use combinations of common words and numbers to try to hack a password. Because non-alphanumeric characters don't fit into common words or numbers, including a character such as an exclamation point will defeat these common-word hacks. Not all serving systems let you use characters such as @, \$, %, or \, however, so you need to experiment to see if a particular server will accept them.

Once you've forced your users to choose strong passwords, you should make them change passwords at regular intervals. While this concept sounds good on paper, and for the Network+ exam you should remember that regular password changing is a good idea, in the real world it is a hard policy to maintain. For starters, users tend to forget passwords when they change a lot. One way to remember passwords if your organization forces you to change them is to use a numbering system. I worked at a company that required me to change my password at the beginning of each month, so I did something simple. I took a root password—let's say it was "m3y3rs5"—and simply added a number to the end representing the current month. So when June rolled around, for example, I would change my password to "m3y3rs56." It worked pretty well!

No matter how well your password implantation goes, using passwords always creates administrative problems. First, users forget passwords and someone (usually you) have to access their account and reset their passwords. Second, users will write passwords down, giving hackers an easy way into the network if those bits of paper fall into the wrong hands. If you've got the cash, there are two alternatives to passwords: smart devices and biometrics.

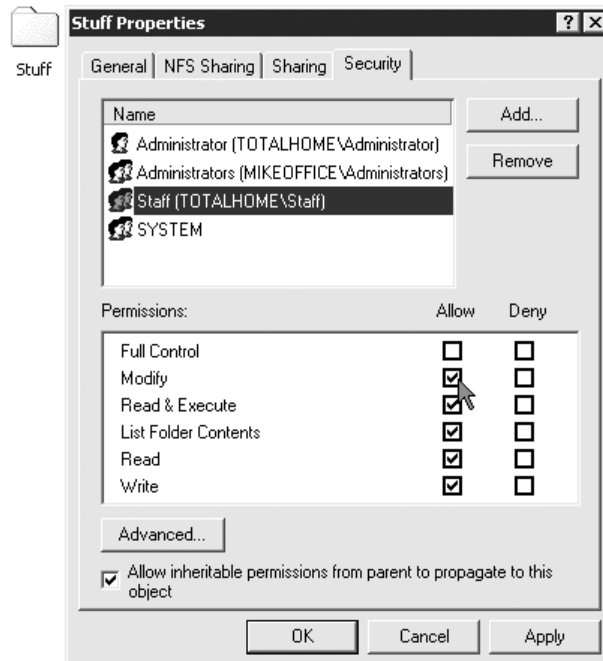
Smart devices are credit cards, USB keys, or other small devices that you insert into your PC in lieu of entering a password. They work extremely well and are incredibly difficult to bypass. They do have the downside in that you might lose them.

If you want to go seriously space-age, then biometrics are the way to go. *Biometric devices* scan fingerprints, retinas, or even the sound of the user's voice to provide a fool-proof replacement for both passwords and smart devices. Biometrics have been around for quite a while, but were relegated to extremely high-security networks due to their high cost (thousand of dollars per device). That price has dropped substantially, making biometrics worthy of consideration for some networks.

User Account Control

Access to user accounts should be restricted to the assigned individuals, and those accounts should have permission to access only the resources they need, no more. Tight control of user accounts is critical to preventing unauthorized access. Disabling unused accounts is an important part of this strategy, but good user account control goes far deeper than that. One of your best tools for user account control is groups. Instead of giving permissions/rights to individual user accounts, give them to groups; this makes keeping track of the permissions assigned to individual user accounts much easier. Figure 17-2 shows me giving permissions to a group for a folder in Windows 2000. Once a group is created and its permissions set, you can then add user accounts to that group as needed. Any user account that becomes a member of a group automatically gets the permissions assigned to that group. Figure 17-3 shows me adding a user to a newly created group in the same Windows 2000 system.

Figure 17-2
Giving a group permissions for a folder in Windows 2000



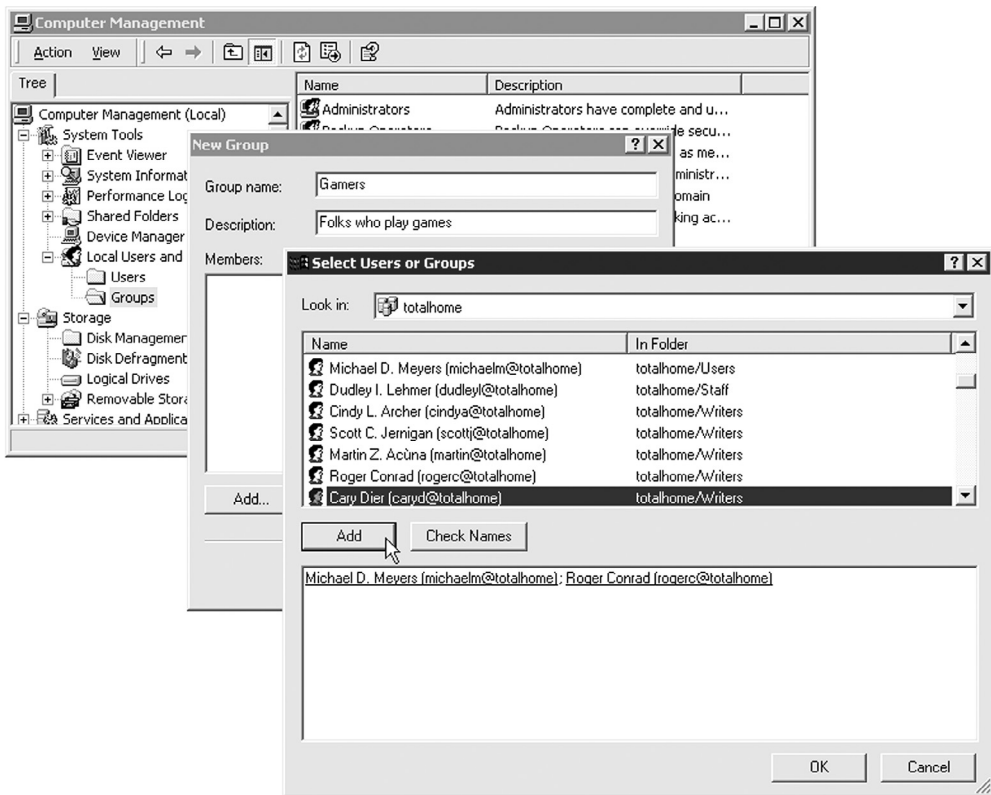


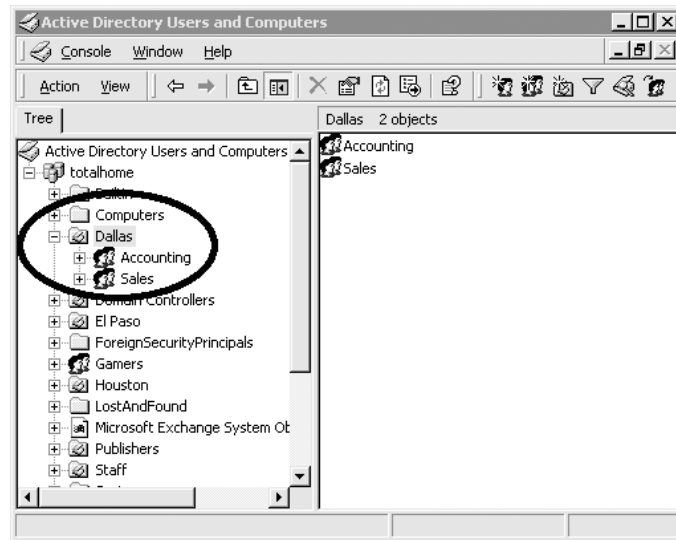
Figure 17-3 Adding a user to a newly created group in Windows 2000

Groups are a great way to get increased complexity without increasing the administrative burden on network administrators, because all network operating systems combine permissions. When a user is a member of more than one group, which permissions does he have with respect to any particular resource? In all network operating systems, the permissions of the groups are *combined*, and the result is what we call the *effective permissions* the user has to access the resource. Let's use an example from Windows 2000. If Timmy is a member of the Sales group, which has List Folder Contents permission to a folder, and he is also a member of the Managers group, which has Read and Execute permissions to the same folder, Timmy will have both List Folder Contents *and* Read and Execute permissions to that folder.

Another great tool for organizing user accounts in network operating systems using organization-based security is the *organizational unit (OU)*. Organizational-based network operating systems like NetWare 4.x/5.x/6.x and Windows 2000 Server/Windows Server 2003 store the entire network structure—computers, groups, printers, users, shared resources—as one big directory tree. This is great for administration, but having

all your groups in one big directory tree can become unwieldy when networks grow past a certain size. Large organizations tend to be geographically dispersed and organizationally complex. For example, most large companies don't just have *an* accounting organization, they have *many* accounting organizations serving different locations and different organizations. Organizational units are a tool to help network administrators group the groups. An OU usually does not get rights or permissions; it is only a storage area for users and groups. Figure 17-4 shows the Dallas OU, containing the Sales and Accounting groups, on a Windows 2000 Server system.

Figure 17-4
Dallas
organizational
unit containing
Sales and
Accounting
groups



Both Windows and NetWare network operating systems provide powerful applications that enable you to see and manipulate various parts of the network tree. Figure 17-5 shows me using the NetWare 5.x NWADMIN application to add users to the Accounting group in the directory tree. Figure 17-6 shows the same activity in the equivalent Windows 2000 Server tool, called Active Directory Users and Computers. These are pretty similar interfaces for such different network operating systems.

Watch out for *default* user accounts and groups—they can become secret backdoors to your network! All network operating systems have a default Everyone group and it can easily be used to sneak into shared resources. This Everyone group, as its name implies, literally includes anyone who connects to that resource. Windows 2000 gives full control to the Everyone group by default, while NetWare gives the Everyone group no access—make sure you know this when working with these operating systems!

All of the default groups—Everyone, Guest, Users—define broad groups of users. Never use them unless you intend to permit all those folks to access a resource. If you use one of the default groups, remember to configure them with the proper rights/permissions to prevent users from doing things you don't want them to do with a shared resource!

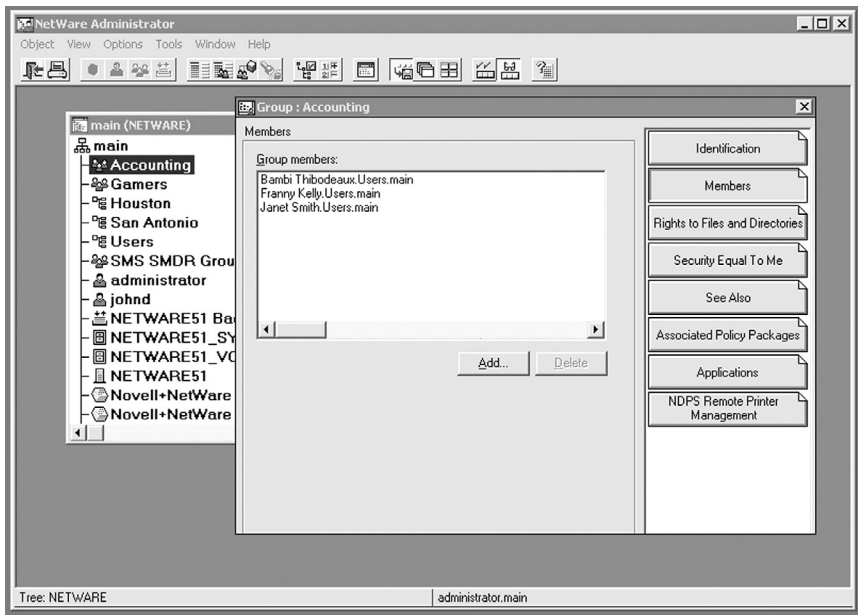


Figure 17-5 Adding users to a group in NetWare 5.x NWADMIN

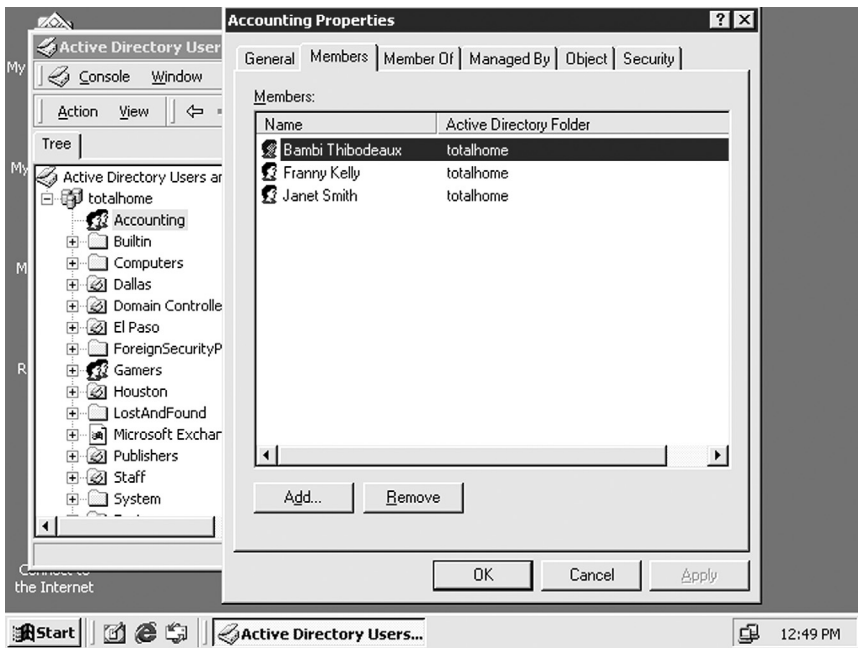


Figure 17-6 Adding users to a group in Windows 2000 Server's Active Directory Users and Computers

All of these groups and organizational units only do one thing for you: They let you keep track of your user accounts. That way you know they are only available for those who need them, and they only access the resources you want them to use. Before we move on, let me add one more tool to your kit: diligence. Managing user accounts is a thankless and difficult task, but one that you must stay on top of if you want to keep your network secure. Most organizations integrate the creation, disabling/enabling, and deletion of user accounts with the work of their human resources folks. Whenever a person joins, quits, or moves, the network admin is always one of the first to know!

Careful Use of Permissions

I have to admit that I gave most of this part away in the previous section when I discussed groups. The administration of rights/permissions can become incredibly complex even with judicious use of groups and organizational units. You now know what happens when a user account has multiple sets of rights/permissions to the same resource, but what happens if the user has one set of rights to a folder, and a different set of rights to one of its subfolders? This brings up a phenomenon called *inheritance*. We won't get into the many ways different network operating systems handle inherited permissions. Lucky for you, Network+ doesn't expect you to understand all the nuances of combined or inherited permissions—just be aware that they exist. However, those who go on to get more advanced certifications such as the CNE or MCSE will become extremely familiar with the many complex permutations of permissions.

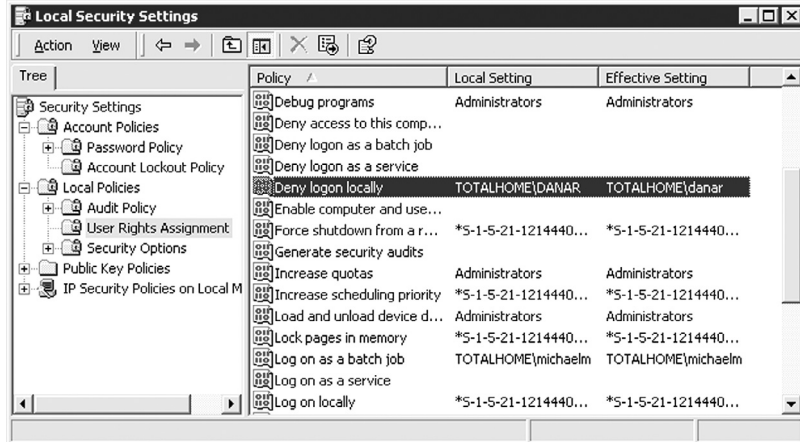
Policies

While rights/permissions control how users access shared resources, there are a number of other functions it would be useful to control that are outside the scope of resources. For example, do you want users to be able to access a command prompt at their Windows system? Do you want users to be able to install software? Would you like to control what systems or what time of day a user can log in? All network operating systems provide you with some capability to control these and literally hundreds of other security parameters, under what both Windows and NetWare call *policies*. I like to think of policies as permissions for activities as opposed to true permissions, which control access to resources. The actual process of performing and using policies varies not only from NOS to NOS, but even among different versions of a single NOS. In concept, however, they all work the same way.

A policy is usually applied to a user account, a computer, a group, or an OU—again this depends on the make and model of NOS. Let's use the example of a network composed of Windows 2000 Professional systems with a Windows 2000 Server system. Every Windows 2000 system has its own local policies program, which enables policies to be placed on that system only. Figure 17-7 shows the tool we use to set local policies on an individual system, called *Local Security Settings*, being used to deny the user account Danar the capability to log on locally.

Local policies work great for individual systems, but they can be a pain to configure if you want to apply the same settings to more than one PC on your network. If you want

Figure 17-7
Local Security
Settings



to apply policy settings *en masse*, then you need to step up to Windows Active Directory domain-based *Group Policy*. Using Group Policy, you can exercise deity-like—Microsoft prefers to use the term *granular*—control over your network clients.

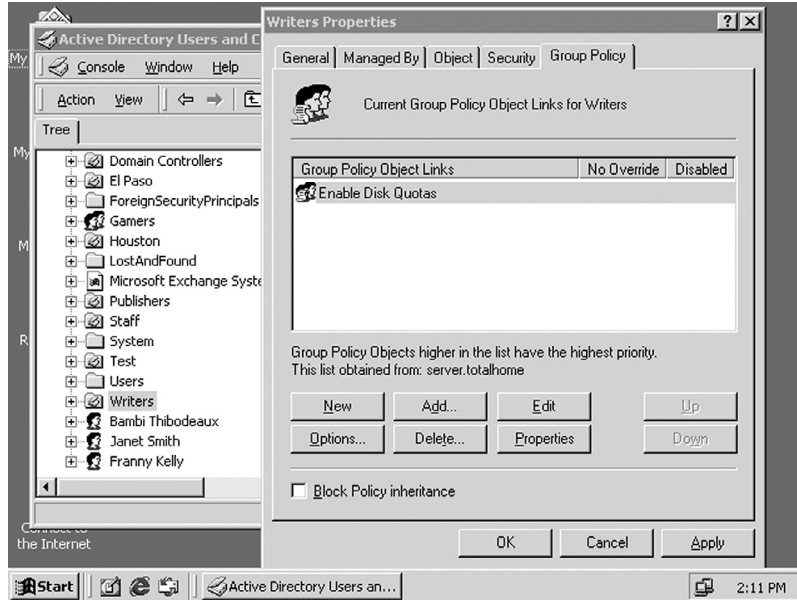
Want to set default wallpaper for every PC in your domain? Group Policy can do that. Want to make certain tools inaccessible to everyone except authorized users? Group Policy can do that too. Want to control access to the Internet, redirect home folders, run scripts, deploy software, or just remind folks that unauthorized access to the network will get them nowhere fast? Group Policy is the answer.

In a Windows 2000 Server or Server 2003 AD domain environment, you apply Group Policy settings to your network in bundles called a *Group Policy Objects (GPOs)*. These GPOs can be linked to the entire domain, to OUs, or to *sites*, which are units that represent PCs (usually Windows 2000 Server or Server 2003 domain controllers) connected to logical IP subnets. For example, Figure 17-8 shows me using the Active Directory Users and Computers console to apply to the Writers OU in my domain a GPO that enables disk quota management. By applying this Group Policy setting, I can set limits to the amount of disk space that members of the Writers OU are allowed to use. *That* should keep my editor from storing too many of his Irish folk music MP3s on the server!

That's just one simple example of the types of settings you can configure using Group Policy. There are literally hundreds of "tweaks" you can apply through Group Policy, from the great to the small, but don't worry too much about familiarizing yourself with each and every one. Group Policy settings are a big topic in the Microsoft Certified Systems Administrator (MCSA) and Microsoft Certified Systems Engineer (MCSE) certification tracks, but for the purposes of the CompTIA Network+ exam, you simply have to be comfortable with the concept behind Group Policy.

For many years NetWare didn't do a lot with policies. NetWare was content to add some of their own policies to the Windows policy list—a pretty smart way to handle things, given that NetWare wasn't that interested in the goings-on of client systems. Later versions of NetWare began to include a number of tools that made policies more important to NetWare networks, and Novell created a tool called ZENworks. ZENworks has its

Figure 17-8
Applying a GPO
using Active
Directory Users
and Computers



own set of policies to address issues that Windows policies do not cover. This tool does much more than just create NetWare policies; for example, ZENworks is Novell's tool for network-based software installation.

Linux doesn't provide a single application that you open to set up policies, like Windows does. In fact, Linux doesn't even use the name policies. Instead, Linux relies on individual applications to set up policies for whatever they're doing. This is in keeping with the Linux paradigm of having lots of little programs that do one thing well, as opposed to the Windows paradigm of having one program try to be all things for all applications. For the Network+ exam, you can safely say that Linux does not have policies. Certainly not in the Microsoft sense of the word!

Although I could never name every possible policy you can enable on a Windows system, here's a list of some of those more commonly used:

- **Prevent Registry Edits** If you try to edit the Registry, you get a failure message.
- **Prevent Access to the Command Prompt** This policy keeps users from getting to the command prompt by turning off the **Run** command and the MS-DOS Prompt shortcut.
- **Log on Locally** This policy defines who may log on to the system locally.
- **Shut Down System** This policy defines who may shut down the system.
- **Minimum Password Length** This policy forces a minimum password length.
- **Account Lockout Threshold** This policy sets the maximum number of logon attempts a person can make before they are locked out of the account.

- **Disable Windows Installer** This policy prevents users from installing software.
- **Printer Browsing** This policy enables users to browse for printers on the network, as opposed to using only assigned printers.



NOTE One of the big improvements of Windows Server 2003 over Windows 2000 Server is in the use of policies. Server 2003 has many new policies that provide more powerful control over users.

While the Network+ exam doesn't expect you to know how to implement policies on any type of network, you are expected to understand that policies exist, especially on Windows networks, and that they can do amazing things in terms of controlling what users can do on their systems. If you ever try to get to a command prompt on a Windows system, only to discover the **Run** command is grayed out, blame it on a policy, not the computer!

Protecting a Network from External Threats

So far, I've stressed that internal threats are far more likely to cause network failures than external threats, but in no way am I suggesting you should take external threats lightly. Hacking has reached epidemic proportions as the Internet has expanded beyond the wildest fantasies of network pioneers, and easy access to hacking tools and information has made virtually any 13-year-old with a modem and time on his hands a serious threat to your network.

Securing networks from external threats is an ever-evolving competition between hackers and security people to find vulnerabilities in networking software and hardware. It can be a horse race—hackers finding and exploiting network vulnerabilities, neck and neck with security experts creating fixes. Newly discovered vulnerabilities always make the news, but the vast majority of intrusions are not due to a hacker discovering a new vulnerability and using it. In almost all cases, hackers take advantage of well-known vulnerabilities that network administrators have simply failed to fix. These well-known vulnerabilities are what we'll concentrate on in this section.

Physical Protection

Most techs consider installing firewalls and instituting policies a critical step in securing your network. No doubt these issues are important, but you can't consider a network secure unless you provide some physical protection to your network. I separate physical protection into two different areas: protection of servers and protection of clients.

Server protection is easy. Lock up your servers to prevent physical access by any unauthorized person. Large organizations have special server rooms, complete with card-key locks and tracking of anyone who enters or exits. Smaller organizations will at least have a locked closet. While you're locking up your servers, don't forget about any network

switches! Hackers can access networks by plugging into a switch, so don't leave any switches available to them.

Physical server protection doesn't stop with a locked door. One of the most common mistakes made by techs is walking away from a server while still logged in. Always log off your server when not in use! As a backup, add a password-protected screen saver.

It's difficult to lock up all of your client systems, but you should have your users performing some physical security. First, all users should use screensaver passwords. Hackers will take advantage of unattended systems to get access to networks. Second, make users aware of the potential for dumpster diving and make paper shredders available. Last, tell users to mind their work areas. It's amazing how many users leave passwords available. I can go into any office, open a few desk drawers, and will invariably find little yellow sticky notes with user names and passwords. If users must write down passwords, tell them to put them in locked drawers!

Firewalls

I always fear the moment when technical terms move beyond the technical people and start to find use in the nontechnical world. The moment any technical term becomes part of the common vernacular, you can bet that its true meaning will become obscured, because without a technical background people are reduced to simplistic descriptions of what is invariably a far more complex idea. I submit the term *firewall* as a perfect example of this phenomenon. Most people with some level of computer knowledge think of a firewall as some sort of thing-a-ma-bob that protects an internal network from unauthorized access to and from the Internet at large. That type of definition might work for your VP as you explain why you need to get a firewall, but as techs, we need a deeper understanding.

Firewalls protect networks using a number of methods, such as hiding IP addresses and blocking TCP/IP ports. Any device that uses any or all of the techniques discussed next is by definition a firewall. Let's look at the protection methods, and then turn to implementation in the last section of this chapter.

Hiding the IPs

The first and most common technique for protecting a network is to hide the real IP addresses of the internal network systems from the Internet. If a hacker gets a real IP address, he can then begin to probe that system, looking for vulnerabilities. If you can prevent a hacker from getting an IP address to probe, you've stopped most hacking techniques cold. You already know how to hide IP addresses: either via a *Network Address Translation (NAT)* or a proxy server. Choosing between a NAT and a proxy server requires some analysis, because each has its advantages and disadvantages. NATs only translate IP addresses. This means a NAT has no interest in the TCP port or the information and it can work fairly quickly. A proxy server can change the port numbers as well as hide the IP addresses; this adds an extra level of security but at the cost of slower throughput because this involves more work by the system. For this reason, many networks only use NATs.

One problem we run into when discussing proxy servers and NAT is that proxy-serving programs also provide address translation—but using a completely different method than a NAT. At first you may say, “Well, if a proxy server changes the port number as well as the IP address, isn’t a proxy server always better?” A proxy server certainly provides more protection than a router that only performs NAT, but it comes with a significant overhead cost. Proxy servers tend to slow down network access—plus, if you ever change your proxy server’s IP address, you’ll have to go to every network-aware application on every system on your network and update the proxy server settings. For these reasons, most networks have abandoned proxy serving in favor of routers with NAT.



TIP Most Windows clients now have such powerful caches that a proxy server’s caching capability doesn’t provide substantial improvement in web page access.

Now you know another reason why most routers have built-in NATs. Not only do NATs reduce the need for true IANA-supplied public IP addresses, but they also do a great job protecting networks from hackers (see Figure 17-9).

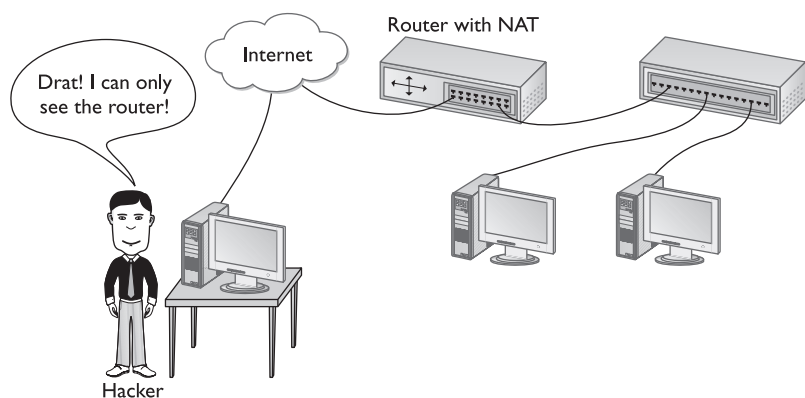
Port Filtering

The second most common firewall tool is *port filtering*, also called *port blocking*. Hackers will often try less commonly used port numbers to get into a network. Port filtering simply means preventing the passage of any TCP or UDP packets through any ports other than the ones prescribed by the system administrator. Port filtering is effective, but it requires some serious configuration to work properly. The question is always, “Which ports do I allow into the network?” No one has problems with the well-known ports like 80 (HTTP), 20/21 (FTP), 25 (SMTP), and 110 (POP), but there are a large number of lesser-known ports that networks often want opened.

I recently installed port filtering on my personal firewall and everything worked great—until I decided to play the popular game Half-Life on the Internet. I simply could not connect to the Internet servers, until I discovered that Half-Life required TCP ports

Figure 17-9

Hacker stopped cold by NAT!



27010 and 27015 to work over the Internet. After reconfiguring my port filter I was able to play Half-Life, but when I tried to talk to my friends using Microsoft NetMeeting, I couldn't log on to a NetMeeting server! Want to guess where the problem lay? Yup, I needed to open ports 389, 522, 1503, 1720, and 1731! How did I figure this out? I didn't know which ports to open, but I suspected that my problem was in the port arena so I fired up my web browser (thank goodness that worked!) and went to the Microsoft NetMeeting web site, which told me which ports I needed to open. This constant opening and closing of ports is one of the prices you pay for the protection of port filtering, but it sure stops hackers if they can't use strange ports to gain access!

Most routers that provide port blocking manifest it in one of two ways. The first way is to have port filtering close *all* ports until you open them explicitly. The other port filtering method is to leave all ports open unless you explicitly close them. The gotcha here is that most types of IP sessions require *dynamic port* usage. For example, when my system makes a query for a web page on HTTP port 80, the web server and my system establish a session using a *different* port to send the web pages to my system. Figure 17-10 shows the results of running NETSTAT with the `-n` switch while I have a number of web pages open—note the TCP ports used for the incoming web pages (the Local Address column). Dynamic ports can cause some problems for older (much older) port filtering systems, but almost all of today's port filtering systems are aware of this issue and handle it automatically.

Port filters have many different interfaces. On my little gateway router, the port filtering uses a pretty, web-based interface shown in Figure 17-11. Linux systems use either IPTABLES or NETFILTER for their firewall work. Like most Linux tools, these programs are rather dull to look at directly and require substantial skill manipulating text files to do your filtering chores. Most Linux distributions come with handy graphical tools, however, to make the job much easier. Figure 17-12 shows the firewall configuration screen from the popular YaST utility, found on the SUSE Linux distribution.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\>netstat -n

Active Connections

  Proto  Local Address          Foreign Address         State
  TCP    192.168.4.10:1707      207.46.144.86:80       ESTABLISHED
  TCP    192.168.4.10:1710      207.46.238.24:80       ESTABLISHED
  TCP    192.168.4.10:1711      207.46.144.86:80       ESTABLISHED
  TCP    192.168.4.10:1712      207.46.144.86:80       ESTABLISHED
  TCP    192.168.4.10:1713      207.46.144.86:80       ESTABLISHED
  TCP    192.168.4.10:1741      216.239.33.100:80      CLOSE_WAIT

C:\>
```

Figure 17-10 The `netstat -n` command showing HTTP connections

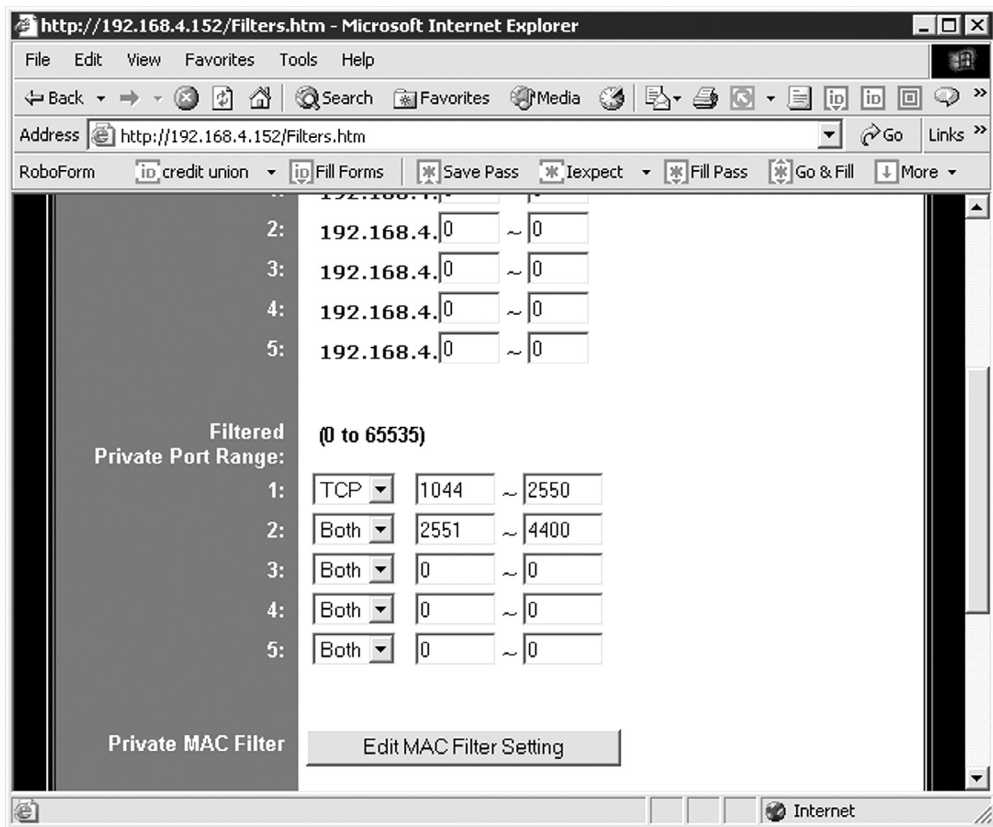


Figure 17-11 Web-based port-filtering interface

So, can one router have both a NAT and port filtering? You bet it can! Most gateway routers come with both—you just need to take the time to configure them and make them work!



TIP The Network+ exam expects you to know that NAT, proxy servers, and port filters are typical firewall functions!

Packet Filtering

Port filtering deals only with port numbers; it completely disregards IP addresses. If an IP packet comes in with a filtered port number, the packet is blocked, regardless of the IP address. *Packet filtering* works in the same way, except it only looks at the IP addresses. *Packet filters*, also known as *IP filters*, will block any incoming or outgoing packet from a

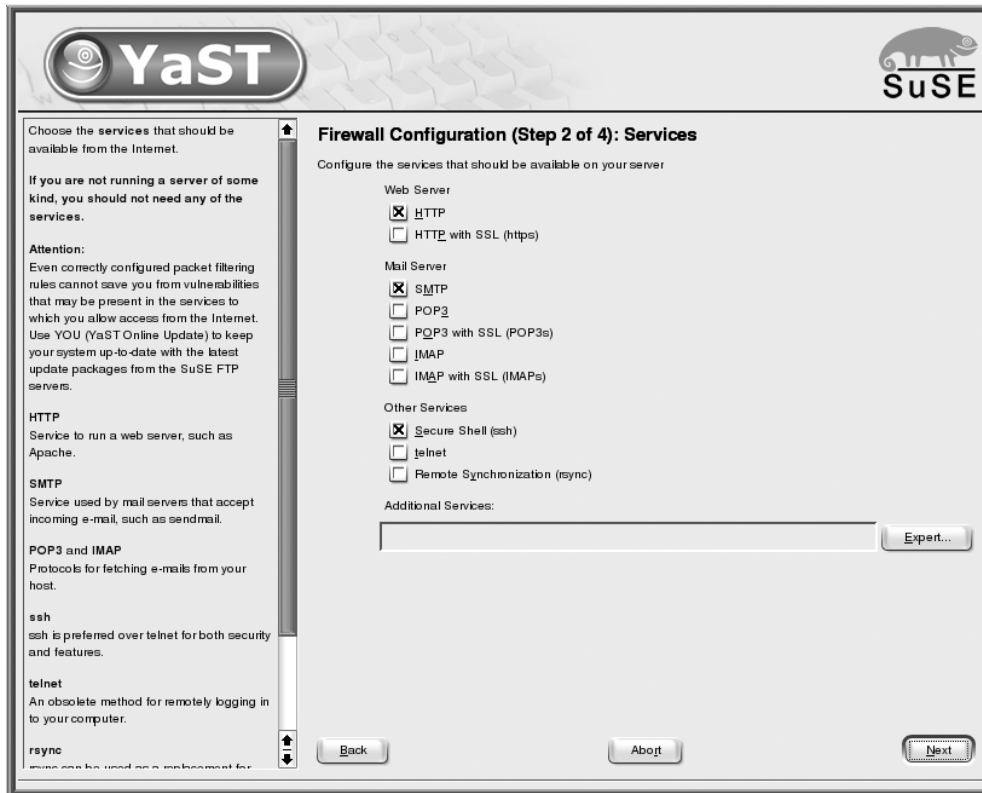


Figure 17-12 The YaST configuration program

particular IP address or range of IP addresses. Packet filters are far better at blocking outgoing IP addresses, because the network administrator knows and can specify the IP addresses of the internal systems. Blocking outgoing packets is a good way to prevent users on certain systems from accessing the Internet. Figure 17-13 shows a configuration page from a router designed to block different ranges of IP addresses and port numbers.

Encryption

Firewalls do a great job controlling traffic coming into or out of a network from the Internet, but they do nothing to stop interceptor hackers who monitor traffic on the public Internet looking for vulnerabilities. Once a packet is on the Internet itself, anyone with the right equipment can intercept and inspect it. Inspected packets are a cornucopia of passwords, account names, and other tidbits that hackers can use to intrude into your network. Because we can't stop hackers from inspecting these packets, we must turn to *encryption* to make them unreadable.

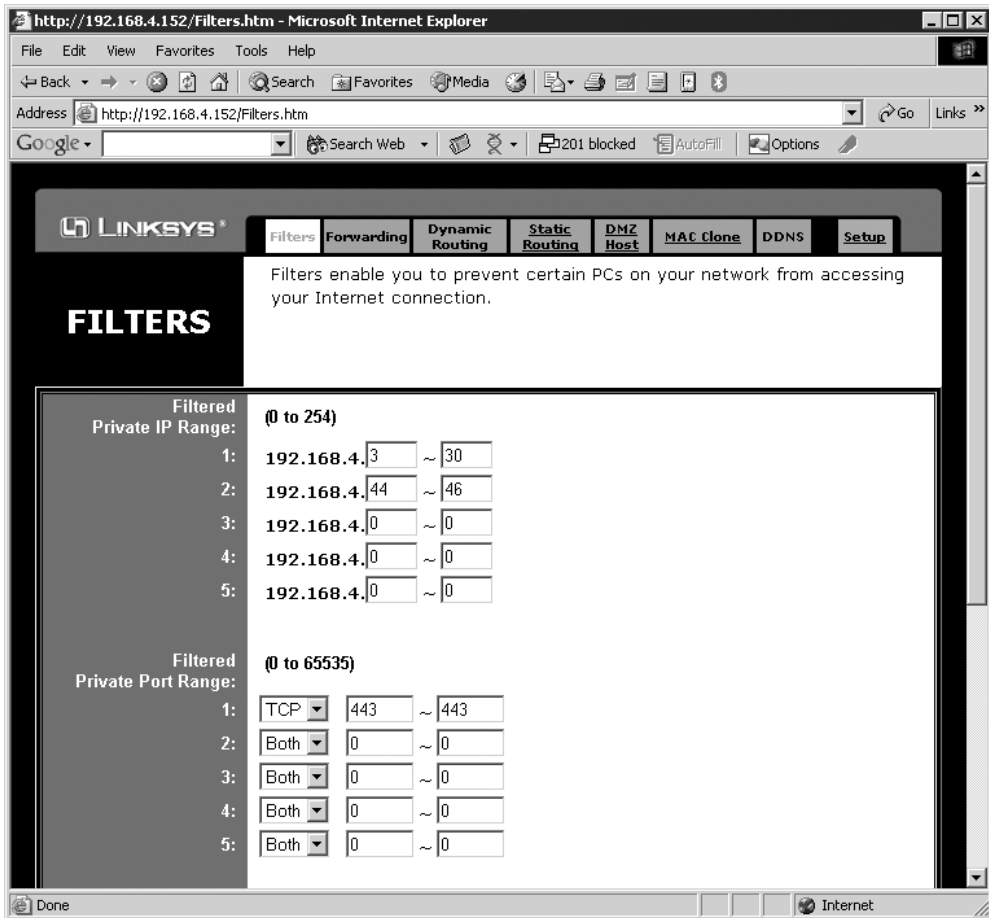


Figure 17-13 Blocking IP addresses

Network encryption occurs at many different levels and is in no way limited to Internet-based activities. Not only are there many levels of network encryption, but each encryption level provides multiple standards and options, making encryption one of the most complicated of all networking issues. You need to understand where encryption comes into play, what options are available, and what you can use to protect your network.

Authentication

Throughout this book, I've used examples where users type in user names and passwords to gain access to networks. But have you ever considered the process that takes place each time this *authentication* is requested? If you're thinking that when a user types in a user name and password, that information is sent to a server of some sort to be authenticated, you're right—but do you know how the user name and password get to the serving system? That's where encryption becomes important in authentication.

In a local network, encryption is usually handled by the NOS. Because NOS makers usually control software development of both the client and the server, they can create their own proprietary encryptions. However, in today's increasingly interconnected and diverse networking environment, there is a motivation to enable different network operating systems to authenticate any client system from any other NOS. Modern network operating systems like Windows NT/2000/XP/2003 and NetWare 4.x/5.x/6.x use standard authentication encryptions like MIT's *Kerberos*, enabling multiple brands of servers to authenticate multiple brands of clients. These LAN encryptions are usually transparent and work quite nicely even in mixed networks.

Unfortunately, this uniformity falls away as you begin to add remote access authentications. There are so many different remote access tools, based on UNIX/Linux, NetWare, and Windows serving programs, that most remote access systems have to support a variety of different authentication methods.

PAP Password Authentication Protocol (PAP) is the oldest and most basic form of authentication. It's also the least safe, because it sends all passwords in clear text. No NOS uses PAP for a client system's login, but almost all network operating systems that provide remote access service will support PAP for backward compatibility with a host of older programs (like Telnet) that only use PAP.

CHAP Challenge Handshake Authentication Protocol (CHAP) is the most common remote access protocol. CHAP has the serving system challenge the remote client. A *challenge* is where the host system asks the remote client some secret—usually a password—that the remote client must then respond with for the host to allow the connection.

MS-CHAP MS-CHAP is Microsoft's variation of the CHAP protocol. It uses a slightly more advanced encryption protocol.

Configuring Dial-Up Encryption

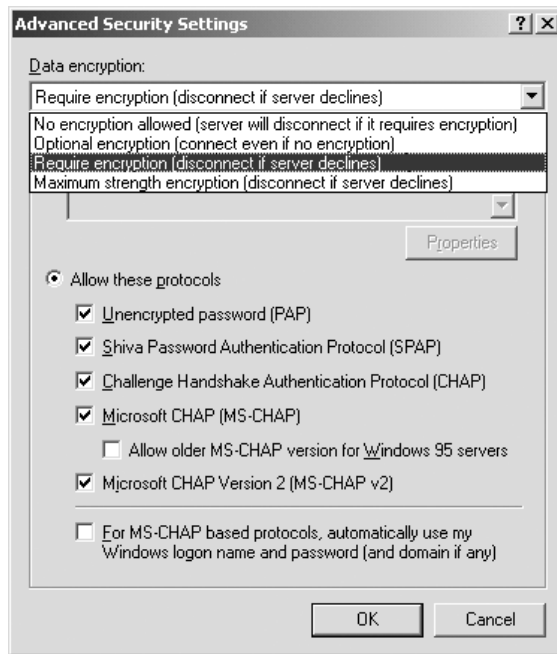
It's the server not the client that controls the choice of dial-up encryption. Microsoft clients can handle a broad selection of authentication encryption methods, including no authentication at all. On the rare occasion when you have to change your client's default encryption settings for a dial-up connection, you'll need to journey deep into the bowels of its properties. Figure 17-14 shows the Windows 2000 dialog box where you configure encryption, called Advanced Security Settings. The person who controls the server's configuration will tell you which encryption method to select here.

Data Encryption

Encryption methods don't stop at the authentication level. There are a number of ways to encrypt network *data* as well. The choice of encryption method is dictated to a large degree by the method used by the communicating systems to connect. Many networks consist of multiple networks linked together by some sort of private connection, usually some kind of telephone line like ISDN or T1. Microsoft's encryption method of choice for this type of network is called *IPSec* (derived from *IP security*). IPSec provides transparent encryption between the server and the client. IPSec will also work in VPNs, but other encryption methods are more commonly used in those situations.

Figure 17-14

Setting dial-up
encryption in the
Windows 2000
Advanced
Security Settings
dialog box



Application Encryption

When it comes to encryption, even TCP/IP applications can get into the swing of things. The most famous of all application encryptions is Netscape's *Secure Sockets Layer (SSL)* security protocol, which is used to create secure web sites. Microsoft incorporates SSL into its more far-reaching HTTPS (HTTP over SSL) protocol. These protocols make it possible to create the secure web sites we use to make purchases over the Internet. HTTPS web sites can be identified by the *HTTPS://* included in their URL (see Figure 17-15).

Public Keys and Certificates

Did you ever use one of those “secret decoder rings” when you were young? I thought secret decoder rings were a thing of the past, until I recently saw my daughter playing with one she got from a box of breakfast cereal. A secret decoder ring uses an encryption algorithm to exchange each letter of the alphabet for another, enabling you to turn readable text into a coded message or vice versa. For example, your decoder ring might exchange each letter in the alphabet for the letter three steps away—which would transform the statement “I HAVE A SECRET” into something like “F EXSB X PBZOBQ.” In this case, moving the letters of the alphabet three positions is the algorithm, and the secret decoder ring is the key we use to encrypt and decrypt.

Encryption in the world of electronic data works in much the same way. Incredibly complex algorithms use a special string of numbers and letters, known as a key, to encrypt and decrypt anything from Word documents to the data areas of IP packets. Given enough time, most people could break the simple, three-letter algorithm used in our

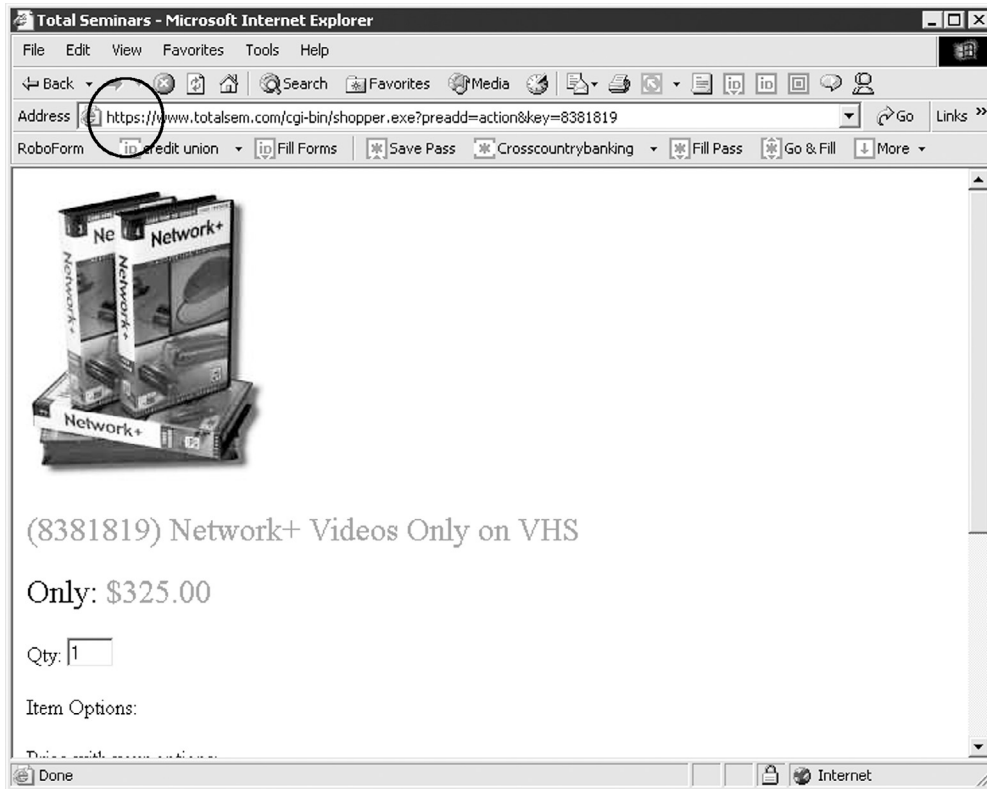


Figure 17-15 A secure web site

first example—we call this *weak encryption*. The best encryption algorithms used in computing are for all practical purposes impossible to crack, and are thus known as *strong encryption*.



TIP The length of the key is an indicator of the strength of the algorithm. Most encryption methods will mention the size of their key in bits. 128-bit is considered the safest practical key size.

Even the strongest encryption is easily broken if someone can get the key. Early encryption techniques used what is called *symmetric key*. Symmetric key means the same key is used both to encrypt and to decrypt. This leads to the obvious question: “How do you get the key to the other person without anyone else getting it?” Simply sending it over the network is risky—a hacker might intercept it. If the key is stolen from either system the encryption is also compromised.

To avoid this single-key issue, most strong encryption uses an *asymmetric key* methodology. The asymmetric approach uses two keys: a public key and a private key. The encryption algorithms are designed so that anything encrypted with the public key can

only be decrypted with the private key. You send out the public key to anyone you want to send you encrypted information. Since only the private key can decrypt data, stealing the public key is useless. Of course, if you want two-way encryption, each party must send the other its public key. We refer to this method of public and private keys simply as *public key* encryption.

Public key provides another big benefit beyond encryption: *digital signatures*. For certain types of transactions, you don't need encryption, but you *would* like to know that the data is actually coming from the person or source that you think is sending it. A digital signature is a string of characters created by running an algorithm on the private key and a special value of the data called a *hash*. The person receiving the signature then uses the public key to generate what's called a *digest* and compares the two values. If they are the same, you can be certain that they came from the person holding the private key!

Public key has one weak spot. Let's say you're about to go to a secure web site to buy some great textbooks. Part of SSL's security comes from the use of public keys. Secure web sites will send you a public key for that web site to handle the transaction. But then, how can you know this truly is the public key, and not a forged key placed on the web site by a hacker? This is the third interesting aspect of Public keys: digital certificates.

Digital certificates are public keys signed with the digital signature from a trusted third party called a certificate authority (CA). Web sites pay these CAs hundreds of dollars per year just for the CA to sign the web site's digital certificates. The predominate CA for secure web sites is Verisign (www.verisign.com). Certificates are interesting in that they are one of the few parts of the HTTPS protocol that you can actually see if you want. Go to any secure web site and look for the small lock icon that appears at the bottom of the web browser. Click the lock to see the certificate. Figure 17-16 shows a typical certificate.

Figure 17-16
Certificate details



VLAN

The best place for a hacker to access a system is right at the system itself. The second best place to access a system is by sitting at another computer on the same collision domain. When two computers sit on the same collision domain, the hacker has no concerns about firewalls and other tools that we use to protect our systems from outside threats. When you place a number of systems on a single switch, they will by design have total access to one another—they are after all on the same collision domain. In most cases, this is fine. We want all systems on a single collision domain to have easy access to each other. More secure environments or environments that want to cut down on broadcast traffic, however, may find themselves wanting to reorganize their networks into multiple collision domains.

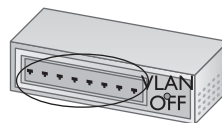
Chopping a single collision domain into multiple domains is usually a pricey and somewhat complex process. In most cases, a network on a single collision domain is probably already well-established. You have a single switch (or a few switches chained together) in a single equipment room with horizontal cable connections to each system. Breaking this network into multiple pieces requires you to add routers between the switches—and if you don't have multiple switches, you had better go buy some! The cost of routers and the sheer amount of configuration involved in this process motivated the networking industry in the mid-1990s to come up with an alternative method of organizing networks. That solution was called a *Virtual LAN (VLAN)*.

Simply put, a VLAN is a LAN that—using smart, VLAN-capable switches—can place some systems (or any systems, on the more expensive VLANs) on whatever collision domain you want. The simplest manifestation of VLAN comes in the form of small, eight-port switches; at the press of a button, these switches can split the single eight-port collision domain into two four-port domains (Figure 17-17).

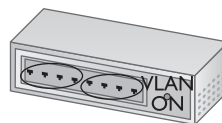


NOTE VLANs are great for security, but they're also a great solution for reducing network traffic and for administration of networks.

Figure 17-17
Simple VLAN



Single collision domain



Two collision domains

With a VLAN, it's as though you take a single switch and with the press of a button turn it into two separate switches! So how would computers on different sides of this VLAN communicate? Well, with this particular VLAN, they wouldn't—unless you could figure out a way to place a router in between the two collision domains! More complex VLAN boxes have built-in functions to enable the two collision domains to connect. Some have a built-in router; others use proprietary methods involving MAC addresses or some electronics that can tell one port on the VLAN switch from another.

The VLAN boxes that identify each port enable you to combine the ports into any grouping of collision domains. A 48-port VLAN box can create a collision domain, for example, with ports 2 and 46. These boxes provide excellent control over network splitting and configuration, although at a hefty price tag. Figure 17-18 shows the configuration screen for a high-end Cisco VLAN box. Note the number of VLANs on the left-hand side. In this case, two of the ports on slot 3, port 15 and port 25, are configured for VLAN-1.

What's a slot? Well, this VLAN box is so large that it uses slots: open spaces on the box that accept many different types of ports on modules called *blades*. You can insert a blade of switched ports into these slots, for example, and create a huge VLAN configuration!

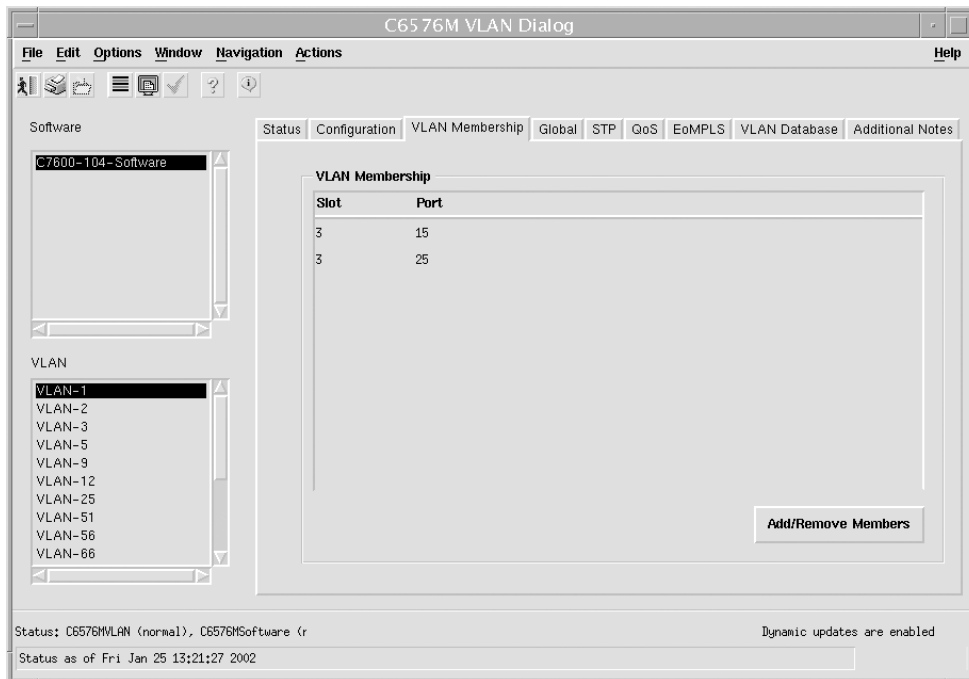


Figure 17-18 Complex VLAN configuration

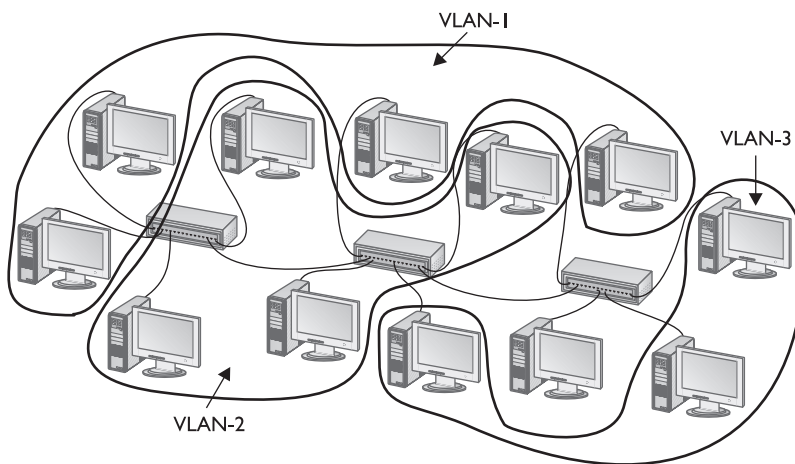


NOTE When VLAN switches get complex enough to configure individual ports, many of the folks who make these boxes don't like to call them switches because they do so much more than mere switches. Some manufacturers call them routers; others stick with switches. I've used the word "box" as a generic label.

The ultimate in VLANs comes when you have multiple VLAN boxes that work together to make complex collision domains that span physical boxes. Figure 17-19 shows just this type of VLAN setup. Note how even though there are three VLAN boxes, different systems connected to different boxes can all be members of the same collision domain. In this example, there are three collision domains: VLAN-1, VLAN-2, and VLAN-3.

Figure 17-19

Very complex
VLAN



VLANs are also handy in situations where you have a large number of systems with static IP information that you don't want to change, for example an ISP running hundreds of web servers. If a system must be physically moved, it's easy to reconfigure the VLAN to keep the system in its original network ID, enabling the system to keep all of its IP information.

Implementing External Network Security

Now that you understand how to protect your networks from external threats, let's take a look at a few common implementations of network protection. I've chosen three typical setups: a single home system connected to the Internet, a small office network, and a large organizational network.

Personal Connections

Back in the days of dial-up connections, the concept of protection from external threats wasn't very interesting. The concept of dial-up alone was more than enough protection

for most users. First, systems using dial-up connections were by definition only periodically on the Internet, making them tough for hackers to detect. Second, all dial-up connections use DHCP-assigned IP addresses, so even if a hacker could access a dial-up user during one session, that dial-up user would almost certainly have a different IP address the next time they accessed the Internet. As long as they have installed a good anti-virus program, dial-up users have nothing to fear from hackers.

The onset of high-speed, always-connected Internet links has changed the security picture completely. The user who dumps his or her dial-up connection for ADSL or a cable modem immediately becomes a prime target for hackers. Even though most ADSL and cable modems use DHCP links, the lease time for these addresses is more than long enough to give even the casual hacker all the time they need to poke around in the systems.

One of the first items on the agenda of Windows users with high-bandwidth connections is to turn off File and Print Sharing. Because NetBIOS can run over IP, sharing a folder or printer makes it available to anyone on the Internet unless your ISP helps you out by filtering NetBIOS traffic. Some hacker groups run port scanner programs looking for systems with File and Print Sharing enabled and post these IP addresses to public sites (no, I will not tell you where to find them!). When I first got my cable modem about two years ago, I absentmindedly clicked Network Neighborhood and discovered that four of my fellow cable users had their systems shared, and two of them were sharing printers! Being a good neighbor and not a hacker, I made sure they changed their erroneous ways!

Although you can buy a firewall system to place between your system and the Internet, most single users prefer to employ a personal software firewall program like BlackICE Defender or ZoneAlarm Pro (see Figure 17-20). These personal firewall programs are quite powerful, and have the added benefit of being easy to use—plus, many of them are free! These days, there's no excuse for an individual Internet user not to use a personal firewall.

Every version of Windows comes with the handy Internet connection sharing (ICS) but ICS alone doesn't provide any level of support other than NAT. Starting with Windows XP we now have *Internet Connection Firewall (ICF)*. ICF works with ICS to provide basic firewall protection for your network. ICF is often used without ICS to provide protection for single machines connected to the Internet. Figure 17-21 shows the Advanced tab of the NIC's properties dialog box to turn on ICF.

Figure 17-20
ZoneAlarm Pro

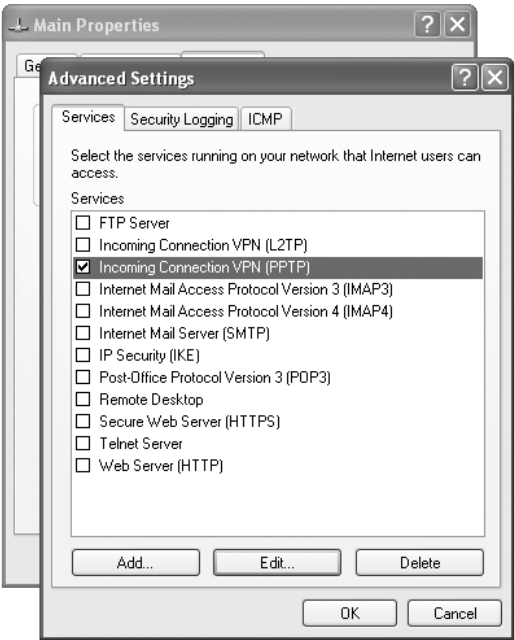


Figure 17-21
Advanced tab
for ICF



By default ICF blocks all incoming IP packets that attempt to initiate a session. This is great for networks that only use the Internet to browse the Web or grab e-mail, but will cause problems in circumstances where you want to provide any type of Internet server on your network. To open well-known TCP ports, click the Settings button (see Figure 17-22).

Figure 17-22
Opening a VPN
port in ICF



Products such as ZoneAlarm and ICF do a fine job protecting a single machine or a small network. But software firewalls run on your system, taking CPU processing away from your system. On an individual system this firewall overhead doesn't strain your system, but once you start to add more than three or four systems or if you need to add advanced functions like a VPN, you'll need a more robust solution. That's where SOHO connections come into play.

SOHO Connections

The typical small office/home office (SOHO) setup is a few networked systems sharing a single Internet connection. Solutions like ICS with ICF will work, but if you want reliability and speed you need a combination firewall/router. You have two choices here: you can drop two NICs in a system and make it a router (expensive, challenging to configure, and yet another system to maintain), or you can buy a SOHO firewall/gateway router like my little Linksys router. These routers are cheap, provide all the firewall functions you'll probably ever need, and require little maintenance. There are a number of great brands out there. Figure 17-23 shows the popular Cisco SOHO 70 series router.

These routers all do NAT with almost no setup. As your needs grow, these SOHO routers grow with you, enabling you to implement IP filtering, port blocking, and many other handy extras. Plus, as your network grows, you can use these same small routers to support separate DNS, DHCP, and WINS servers, although the configuration can become challenging.

Large Network Connections

Large networks need heavy-duty protection that not only protects from external threats, but does so without undue restriction on the overall throughput of the network. To do this, large networks will often use dedicated firewall boxes, which usually sit between the gateway router and the protected network. These firewalls are designed to filter IP traffic (including NAT and proxy functions), as well as to provide high-end tools to track and stop incoming threats. Some of the firewall systems even contain a rather interesting feature called a honey pot. A *honey pot* is a device (or a set of functions within a firewall) that creates a fake network, which seems attackable to a hacker. Instead of trying to access the real network, hackers are attracted to the honey pot, which does nothing more than record their actions and keep them away from the true network.

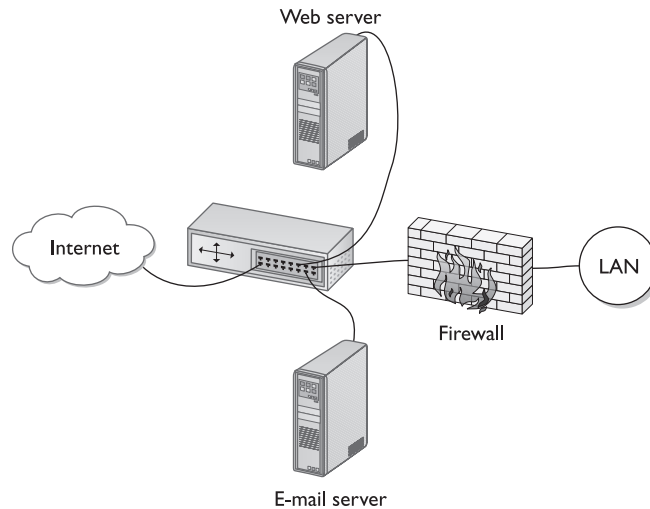
Once you start to add components to your network like web and e-mail servers, you're going to have to step up to a more serious network protection configuration. Be-

Figure 17-23
Cisco SOHO 70
series router



cause web and e-mail servers must have exposure to the Internet, you will need to create what we call a *demilitarized zone (DMZ)*. A DMZ is a lightly protected or unprotected network positioned between your firewall and the Internet. There are a number of ways to configure this; Figure 17-24 shows one classic example.

Figure 17-24
A DMZ
configuration



The private, protected network is called an *intranet*. Compare this term to the term *extranet* you learned in the last chapter and make sure you understand the difference!

Chapter Review

Questions

1. Which two encryption applications work together to make secure web sites for online purchases?
 - A. HTTP
 - B. VPN
 - C. HTTPS
 - D. SSL
2. Which three basic technologies are used with firewalls?
 - A. Proxy servers
 - B. Packet filtering
 - C. Dynamic routing
 - D. Network Address Translation (NAT)

3. What is the most common technique for protecting a network?
 - A. Port filtering
 - B. Hiding IP addresses
 - C. Packet filtering
 - D. Encryption
4. Most routers have built-in proxy servers.
 - A. True
 - B. False
5. Which two of the following methods can you use to hide IP addresses?
 - A. Static IP addresses
 - B. DHCP
 - C. Proxy server
 - D. NAT
6. Which of the following blocks IP packets using any port other than the ones prescribed by the system administrator?
 - A. Hiding IP addresses
 - B. Port filtering
 - C. Packet filtering
 - D. Encryption
7. Which of the following blocks any incoming or outgoing packets from a particular IP address or range of IP addresses?
 - A. Hiding IP addresses
 - B. Port filtering
 - C. Packet filtering
 - D. Encryption
8. Firewalls cannot stop which type of hacker?
 - A. Inspector
 - B. Interceptor
 - C. Controller
 - D. Flooder
9. Which method prevents hackers from reading packets intercepted on the Internet?
 - A. Hiding IP addresses
 - B. Port filtering

- C. Packet filtering
 - D. Encryption
10. Of the following choices, which two are encryption methods?
- A. L2TP
 - B. PPTP
 - C. VPN
 - D. SLIP

Answers

1. C, D. HTTP secure (HTTPS) and Secure Sockets Layer (SSL) are used to create secure web sites.
2. A, B, D. Proxy servers, packet filtering, and network address translation are the basic technologies used with firewalls.
3. B. Hiding IP addresses is the most common technique for protecting a network.
4. B. False. Most routers have built-in NATs, not built-in proxy servers.
5. C, D. Both a proxy server and a NAT can hide IP addresses.
6. B. Port filtering blocks IP packets using any ports other than the ones prescribed by the system administrator.
7. C. Packet filtering blocks any incoming or outgoing packets from a particular IP address or range of IP addresses.
8. B. Firewalls can do nothing to stop interceptor hackers who monitor traffic on the public Internet looking for vulnerabilities.
9. D. Encryption prevents hackers from reading packets intercepted on the Internet.
10. A, B. L2TP is a Cisco encryption method, and PPTP is a Microsoft encryption method.

