CHAPTER 12

Network Operating Systems

The Network+ exam expects you to know how to

- 3.1 Identify the basic capabilities (for example: client support, interoperability, authentication, file and print services, application support and security) of the following server operating systems to access network resources: UNIX/Linux/Mac OS X Server, NetWare, Windows, AppleShare IP (Internet Protocol)
- 3.2 Identify the basic capabilities needed for client workstations to connect to and use network resources (for example: media, network protocols, and peer and server services)
- 3.4 Given a remote connectivity scenario [comprising] a protocol, an authentication scheme, and physical connectivity, configure the connection. Includes connection to the following servers: UNIX/Linux/MAC OS X Server, NetWare, Windows, AppleShare IP

To achieve these goals, you must be able to

- Define the concepts of resource-, server-, and organization-based network models and place any operating system into the proper model
- Describe in detail how different operating systems perform networking
- Configure a Windows client to connect to any version of a Windows server.

Fifteen years ago, operating systems and network operating systems were two very different things. Back then, operating systems (like the old DOS and the first versions of Windows) were stand-alone, designed only for running applications—word processors, games, spreadsheets, and so forth. Operating systems didn't come with any built-in networking software. If you wanted to make one of these old operating systems run on a network, you had to install third-party networking programs. At the same time, if you wanted to make a server system for all your little DOS and Windows computers to connect to, you had to buy special (and usually expensive) *network operating system (NOS)* software—a special operating system designed from the ground up to act as a server in a network. Manufacturers packaged the operating system and network operating system versions of their software differently.

Today the old line between an operating system and a network operating system no longer exists. With one glaring exception, every operating system today comes complete with all the networking software needed to enable any system to share resources and

access shared resources. Even though operating systems and network operating systems are one in the same, different operating systems perform networking in very different ways. For example, Windows 98 SE shares a folder very differently than Windows Server 2003—there is simply no way a tech can support these operating systems without a deep appreciation of those differences!

Let's transform the definition of the term network operating system into something that works for today's operating systems. The phrase *network operating system* refers to the network functions built into a particular operating system. For example, Windows XP is an operating system, but how Windows XP accesses another system's shared resources on a network is a function of Windows XP's network operating system components.

The amount of security provided for users and data is the single greatest issue that differentiates one network operating system from another. The word *security* encompasses a number of critical issues as you'll see in this chapter, such as how, or even if, users can log in. This chapter begins by carefully defining client and server and providing some to appreciate how different operating systems use networking security. This section defines terms such as user accounts, groups, domains, and other important terms and show that every operating system in existence fits into one of three groupings that I call models. Once you have a grasp on these three models, we then turn to the most common operating systems in use today: Windows 9*x*/Me, Windows NT/2000/XP/2003, Linux, Novell NetWare, and Macintosh. Last, we go through the process of creating a network of Windows servers and Windows clients to appreciate some of issues that come into play when building a network.

Historical/Conceptual

Categorizing Operating Systems

All network operating systems share the same fundamental goal: to enable users, the human beings who sit at the computers, to get work done by sharing resources. The routes to that goal vary, of course, depending on the nature of the work. Some network operating systems simply enable users to share folders and printers, while others supply users with access to one or more of literally hundreds of sophisticated shared resources such as web servers, e-mail servers, and DHCP servers. Before choosing the right network operating system for your network, you must define the types of resources you want to share, which systems will do the sharing and the level of security you require. Understanding how the different operating systems fulfill those goals helps facilitate this decision making process.

Before we define network operating system models, it's important to clarify the difference between a client and a server. Chapter 2 defines a client and a server as software programs. In order for a computer to share a resource it must run some form of serving software and a system that wants to access that shared resource must run a client program. While this is absolutely correct, it contradicts other meanings of server. For many people, the term "server" refers to a great, big, heavy-duty computer, hidden in some equipment room, using a powerful CPU, lots of RAM, and stacks of hard drives, as shown in Figure 12-1. If this is a server, how can a server simply be a program as described earlier?

Figure 12-1
A typical server

The answer is that both definitions of server are accurate. Consider a web server. I have an old Windows 98 SE computer that runs a web server software called Microsoft Personal Web Server (PWS). Figure 12-2 shows the PWS Personal Web Manager screen running on that system. I use this system in my home network to keep a calendar that everyone in my family uses to keep track of the many activities taking place in our lives—a handy tool for a very busy family.

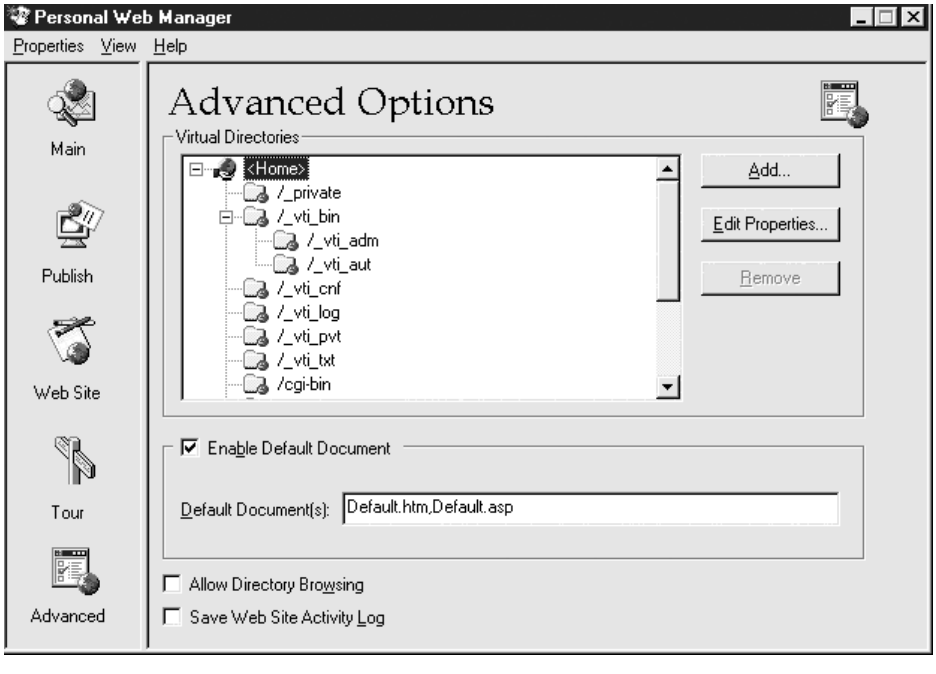


Figure 12-2 PWS running on an old Windows 98 system

This web server works perfectly for my little family, but if lots of people tried to access this web site, that little Windows 98 system's going to get pretty busy, isn't it? In fact, it would slow to a crawl. This is true for any computer running server software. Additionally, if my little server's hard drive crashes, no one will notice outside my family. Imagine the server that runs my company's web site—if that server's hard drive failed, many, many people would notice!

Server-class systems tend to have lots of RAM and powerful CPUs to support heavy use. They have big, redundant, hard drive arrays to keep the system running in case a drive dies. It's perfectly acceptable to call those big, powerful computers *servers* as long as you appreciate that a computer doesn't have to be big to be a server.

NOTE There are cases where serving software demands server-class hardware. Some serving programs, such as the popular Microsoft Exchange Server, have substantial hardware requirements—my little Windows 98 box couldn't hope to run the latest version of Exchange!

While we're on the subject of servers, here's a quick question: If a server is any computer running serving software, can one computer run more than one serving program? Absolutely! In fact, that's the common way to use serving programs. My Windows 2003 server, for example, runs about 13 different serving programs at the same time.

Now that we've clarified the term "server," let's move to "client." As previously defined, a client is a program that is used to access resources shared by serving program. The term client is also used to define a computer whose main job is to access other system's shared resources, a computer that people sit at and use every day—the ones that run applications like Microsoft Word or a web browser. My office system, the one I use to write books, check e-mail and surf the 'Net, is a powerful Athlon 64 with gigabytes of RAM and hundreds of gigabytes of hard drive capacity, running Windows XP Professional. This machine has plenty of serving software installed, but I rarely use it. The vast majority of the time this computer only accesses shared resources from other computers (folders, printers, e-mail and the Web). Even though this powerful computer does a bit of serving—I share a single folder that someone might access once a week or so—its main job is to run applications I need to get my job done. This computer is a client computer. The term *workstation* is also used to define client computers. In general a workstation is a more powerful client system. The terms client and workstation are interchangeable from a networking standpoint.

Can a client system act as a server? Absolutely yes! All modern operating systems provide some form of serving software to enable a system share folders or files. Can a server system also act as a client? In most cases, yes—although you'll see one exception when I show you Novell NetWare in a moment. Let's use my old Windows 98 system as an example. It's acting as a web server but I can still fire up a web browser on the PC and I access other servers' files or printers while the web server runs happily in the background. In most cases, one system can be both a client and a server.

Be careful with the terms client and server. Remember that either term may refer to either a physical system or to a program—and make sure you know the difference when you use these terms!

Test Specific

Why is it so important to understand this concept of server and client? It comes down to recognizing how different operating systems work. Every brand of operating system has very different ways of determining which systems can act as servers and which as clients. If you don't understand the differences in how Novell NetWare handles servers and clients as compared to Microsoft Windows XP, you could find yourself making a major mess by asking your NOS to do something it isn't designed to do!

Client/Server vs. Peer-to-Peer

Networking folks traditionally use the terms *client/server* and *peer-to-peer* to categorize network operating systems. Coined almost 20 years ago, these terms no longer work as an accurate tool for grouping today's complex and powerful operating systems. Even though they no longer do a good job in categorizing, their presence in the common networking vernacular (plus the fact these two terms are on the Network+ exam) motivate us to understand these terms. Let's look at how these two terms define the functionality of different network operating systems.

Client/Server

The earliest network operating systems used a *client/server* model. In that model, certain systems act as dedicated servers. Dedicated servers are called dedicated because that's all they do. You cannot go up to a dedicated server and run Word or Solitaire. Dedicated servers run powerful server network operating systems that offer up files, folders, web pages, and so on to the network's client systems. Client systems on a client/server network never function as servers. One client system can't access shared resources on another client system. Servers serve and clients access, and never the twain shall meet in client/server land! The classic example of this type of network operating system is the popular and powerful *Novell NetWare*. Figure 12-3 shows a typical client/server network. As far as the clients are concerned, the only system on the network is the server system. The clients cannot see each other nor can they share data with each other directly. They must save the data on the server so other systems can access it.

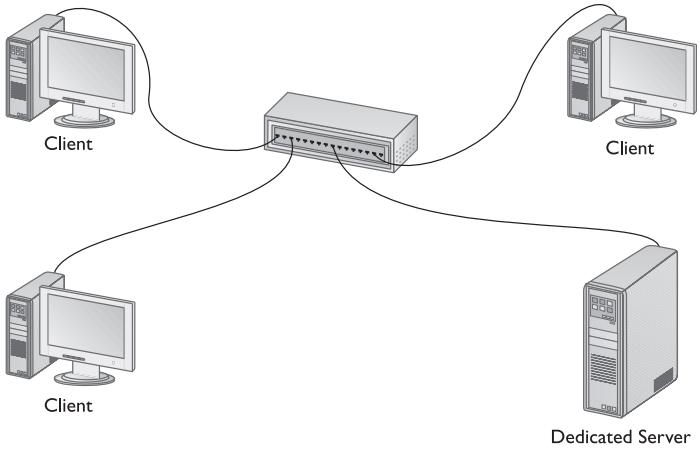


Figure 12-3 In a pure client/server network, the clients cannot access each other directly.

Novell NetWare servers are true dedicated servers. You cannot go up to a Novell NetWare server and write yourself a resume; there is no Windows, there are no user applications. The only thing Novell NetWare servers know how to do is share their own resources, but they share those resources extremely well! Novell NetWare's operating system is totally different from Windows. It requires you to learn an entirely different set of installation, configuration, and administration commands. Figure 12-4 shows a screen from Novell NetWare. Don't let the passing resemblance to Windows fool you—it is a completely different operating system!

TIP Fortunately, the Network+ exam does not expect you to know how to install, configure, or administer a NetWare server, or any other high-end NOS for that matter. Good thing, too, because if they did, this book would be about 5,000 pages!

Peer-to-Peer

In a *peer-to-peer* network operating system, any system can act as a server or a client or both, depending on how you decide to configure it (see Figure 12-5). PCs on peer-to-peer networks frequently act as both clients and servers. One of the most common examples of a peer-to-peer network is the venerable Windows 9*x* series of operating systems.

At first glance, it would seem that peer-to-peer is the way to go—why create a network that doesn't allow the clients to see each other? Wouldn't it make more sense to give users the freedom to allow their systems to both share and access any resource? Good questions! Let's answer them by going back in time to around 1983.

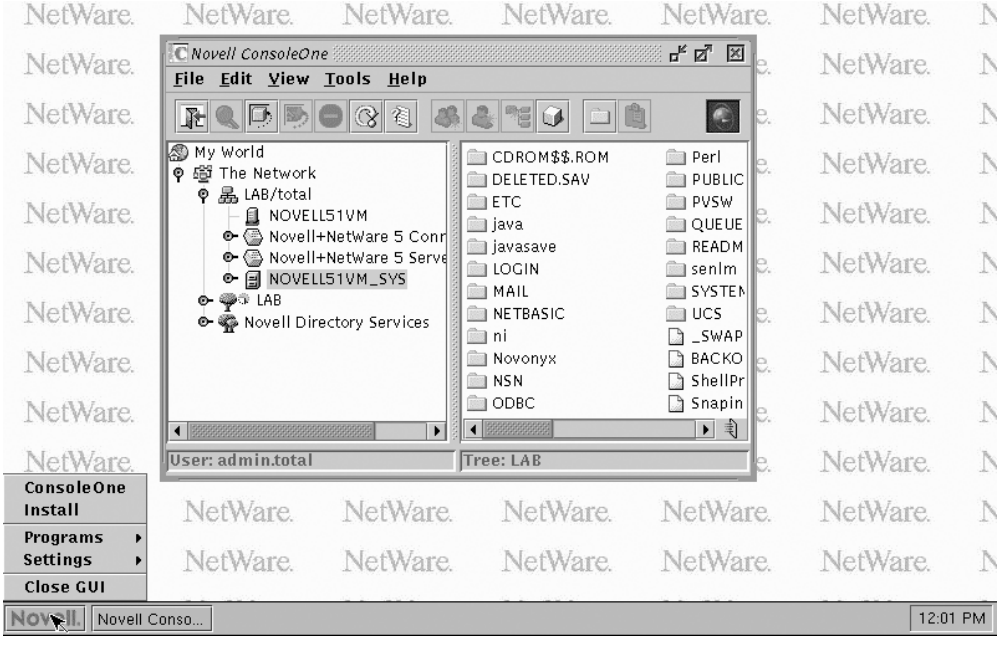


Figure 12-4 Novell NetWare—this isn't Windows!

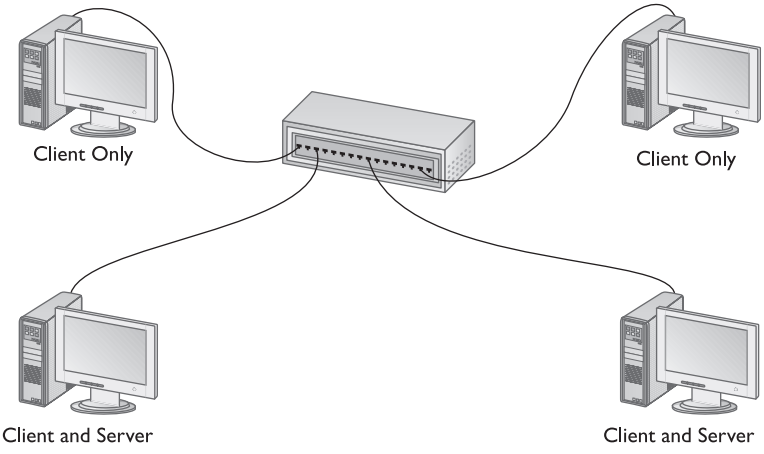


Figure 12-5 In a pure peer-to-peer network, the clients may all act as servers.

Back in the early 1980s, networking didn't really exist in the PC world. These were the days of the first processors, CPUs like the Intel 8088 and 80286. The demands of running a PC worked those poor CPUs to death. Then folks started to get the bright idea of adding networking to these systems. This was great in concept, but adding networking also meant adding lots of extra software, and networking software had to run continuously in the background while other activities were taking place. Those early systems had a very limited amount of processing power with which to tackle such networking challenges. Oh, and did I mention this was back in the days of DOS, the simplistic, single-tasking operating system of your forefathers? All the same, folks were determined to make 8088 systems running DOS handle networking. Clearly, this was going to take some doing!

The answer came in the form of client/server networking. Novell NetWare was invented back in the DOS days. Novell knew good and well that these little PCs didn't have the power to handle both networking and application software, so they put all the functionality in the server software and added the least possible amount of software to the client PCs in the form of special NetWare client software. A NetWare client was nothing more than a DOS—or eventually Windows—system with a little bit of extra software added so the client knew how to access the server's shared resources.

By keeping the server functionality separate from the client systems, the Novell folks made very powerful, dedicated servers without overwhelming the clients with tons of software. NetWare servers had (and still have) tremendous power and great security because the only thing they do is run serving software. In the early days of networking, client/server was king!

In time, CPU power advanced beyond those early CPUs, so Microsoft came up with a new answer to the resource sharing question: peer-to-peer networking. Although peer-to-peer networking appeared in the mid-1980s, it didn't really become popular until the introduction of Microsoft Windows for Workgroups in the early 1990s. Early versions of peer-to-peer network operating systems didn't have nearly the strength of Novell's client/server NOS, but they worked fairly well for small networks. They couldn't have the same security, reliability, and speed as NetWare running on the same hardware, for

the simple reason that every system in a peer-to-peer network had to provide both server and client networking support, and still let users do things like run word processors.

TIP The client/server model means dedicated servers with strong security. Clients can only see the server. In the peer-to-peer model, any system can be a client, server, or both, but at the cost of lower security and additional demands on the system resources of each peer.

For years, we divided all the operating systems into either the client/server or the peer-to-peer camps. The ability to pigeonhole all operating systems into these two network types made us happy and content. All was well with the networking world until Microsoft (who else?) came out with Windows NT in the early 1990s. Windows NT totally messed up the lovely division of networks into client/server and peer-to-peer, because an NT (and Windows 2000/2003 and Windows XP) system can be part of both a client/server network and a peer-to-peer network at the same time! A system running Windows NT gave you all the power and security of a dedicated server, while enabling that system to act as a client as well. Okay, it's actually not quite that simple, and I'll go into more detail in the next section, but the main point stands: NT messed up the entire client/server vs. peer-to-peer concept.

In my opinion, the terms client/server and peer-to-peer are no longer useful ways to organize different types of operating systems. Unfortunately, the terms peer-to-peer and client/server are still tossed around by network folks like dice at a craps table. So how do we manifest the concepts of client/server and peer-to-peer networking in a world that has outgrown these categories? The secret is in security.

Security

Network security involves protecting a network's users from their two greatest enemies: "bad guys" and themselves. When most people think about security issues, they immediately visualize some evil hacker attempting to break into a network and steal company secrets. For many organizations, especially those connected to the Internet, such threats are no joke. Network security, however, must also include controlling how users access the shared resources on their own network. *User-proofing* a network—preventing users from accidentally destroying data or granting access to unauthorized individuals—is a key part of network security.

NOTE What you are about to learn is not on the Network+ exam, nor is it part of common network vernacular. It is my way of understanding networks, an idea I developed with my good friends Brian and Libby Schwarz a few years ago. Even though this concept is not directly on the Network+ exam, it will help you understand networking.

How do we secure our network shares? Well, that begs the question, "What aspects of the shared resource need to be secured?" Think about this for a second: If you're sharing a folder, what exactly do you want to protect? You could just stop anyone from doing

anything to that folder—that's certainly secure—but you can slice the issue a lot more finely than just blocking everyone and everything. For example, you could set up security so users could read the files in a particular folder but not delete them. Or, slicing things even more finely, you could set it up so some users could edit files but not delete them. It's this fine level of detailed control that really makes a network powerful.

These security issues aren't limited to shared folders. Security comes into play with any type of resource you want to share. Every time you access a web site, you run headlong into security. I can set my web server up so some visitors can only view web pages, others can edit certain pages, and a very few others can do anything they want, including delete the entire site if necessary. I can secure my printers so that some people can print to a printer while others can not only print but also configure the printer remotely. The level of control that users can exercise on resources is called permissions or *rights*, depending on the brand of NOS you use. In the next chapter, I'll spend plenty of time discussing permissions and rights, types of protected resources, and how to share and secure them, but for now, what I want to discuss is the different approaches to how an NOS handles all this security—the security models.

Security Models

Odds are good you've heard of terms like user accounts, passwords, groups, domains, and the like. These terms are critical to understanding how a network secures resources. Don't worry if you don't understand any or all of them right now—I cover them all in detail in this chapter and the next. Whether you know these terms or not, do know this: each NOS uses these tools in different ways. This is what my security models concept is all about—it separates the different network operating systems by the way they secure the network's resources.

My scheme divides networks into three different security models, based on which part of the NOS handles the security: Resource, Server, and Organization. Think about this—some part of the NOS must keep track of who can do what on the network. Somewhere in the network, some system—or many systems—must store information that defines what resources are shared and how they are to be shared. Some part of the NOS must check this information whenever a client tries to access a shared resource to make sure that person is allowed to do whatever they are trying to do with that resource. My security models model identifies three parts of the NOS that do the dirty job of handling security. As I describe my three security models, I'll pause to define things like user accounts and groups. Let's get busy learning about the most basic type: resource-based security.

TIP Even though the Network+ exam doesn't discuss my security models, this section of the chapter is crammed with critical definitions you need to understand for the exam.

Resource-Based Security Model

The simplest network operating systems use what I call resource-based security. In *resource-based network operating systems*, the individual resources themselves store the information about who can access the resource and what they can do (see Figure 12-6).

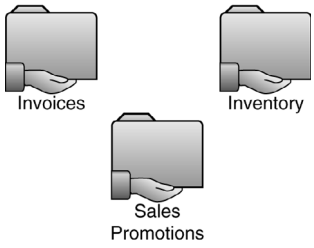


Figure 12-6 In resource-based network operating systems, the resources themselves store the security information.

This information is usually stored within some data structure that is part of the actual shared resource, although it can also be stored in some arbitrary part of the NOS itself. The important thing to understand is that there's no central storage facility for such information—each resource is in charge of its own security storage. The most common example of a resource-based NOS is the Microsoft Windows 9*x* series of operating systems. Most of what the traditional model calls peer-to-peer network operating systems belongs in my resource-based security model.

Storing security information within individual resources is a simple security solution, but one that can handle only simple security issues. As an example, let's take a look at my Windows 98 system. If I want to share a folder called C:\MHTechEDPersonnelRecords, I alternate-click (right-click) that folder and select Sharing. I click the Shared As radio button to see the folder's sharing properties (see Figure 12-7).

Note that I can choose from a whopping three levels of sharing. I can set up the folder so anyone who uses it gets full access to do anything they want; I can set it up so everyone gets only Read access, or I can set a password on the resource to control full vs. read-only access. These are your first examples of permissions/rights! Microsoft calls these network permissions.

While resource-based sharing works perfectly well as far as it goes, it has some serious limitations. First, unless I use a password, I have to give everyone who accesses this

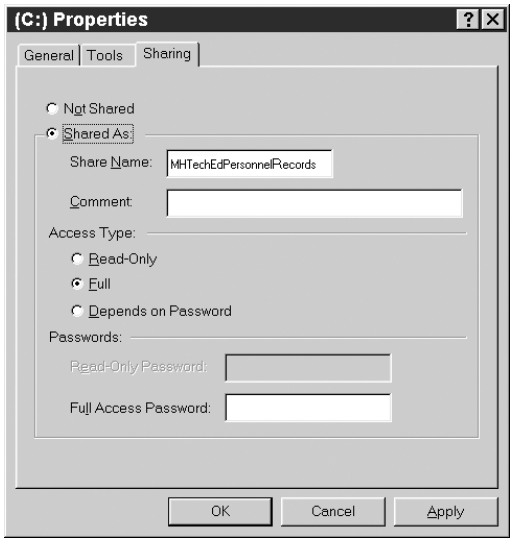


Figure 12-7 The Sharing folder in Windows 98

folder the same level of access. What if I want some people to have full access and others just Read access? I can use a password, of course, but consider the problem from an administrator's standpoint: everyone with the same level of access has the same password. Suppose I want to change just one person's access—I have to change the password and then give the new password to everyone who needs access to that folder. This is not only wildly annoying, it's just asking for problems, because you're trusting those who know this one common password not to tell anyone else.

NOTE In a resource-based network operating system, each resource keeps track of its own permissions.

Now imagine there are 30 or 40 more shared folders and printers you need to protect this way—every one of those shared resources will get its own password! At this rate, a single system could use 60 to 70 different passwords! No problem, you say, if you make them all the same. Well, you could, but then how would you give different users different levels of access? But if you must have different passwords for different users of different resources, how can one lonely admin keep track of them all? Write it all down on a piece of paper? Make a spreadsheet? Let's face it, resource-based security may be fine for simple networks, but this is just not going to hack it in a more complex network.

Server-Based Security Model

A *server-based network* employs a central database on each server to track who gets what level of access to the resources on that server. Most folks give Novell the nod for inventing this security type since it first appeared in the early versions of NetWare. Since Novell had its own operating system, Novell could design every part of the dedicated server specifically to optimize its ability to handle sharing, including the file system. NetWare's file system is nothing like the old FAT16 or FAT32 file systems: Novell invented NetWare from the ground up to share folders. By creating its own file system, Novell could add resource sharing directly to the file system.

To access a shared resource on a NetWare server, you must have a user account. A *user account* contains lists of user rights that tell the network what the user can and cannot do on the network, including file system rights that determine which shared resources the user can access. Each user account also has a *password*. A person who wants to access the shared resources on the server must go through a process called logging on to the server. Figure 12-8 shows a classic example of how a person on a Windows system logs onto a NetWare server via the Novell Client for Windows. As soon as the person logs on to a NetWare server, all of the access privileges for every shared resource on the server are set for the duration of that session.

Server-based security makes life a lot easier from an administrative standpoint. Still, in a large organization, assigning specific rights to each user individually makes for an excessive workload for the network administrator. The solution: organize users with similar needs into *groups*. For example, Alice, the administrator of the network, assigns Greg, Bobby, and Peter's user accounts to the ACCOUNTING group and Jan's user

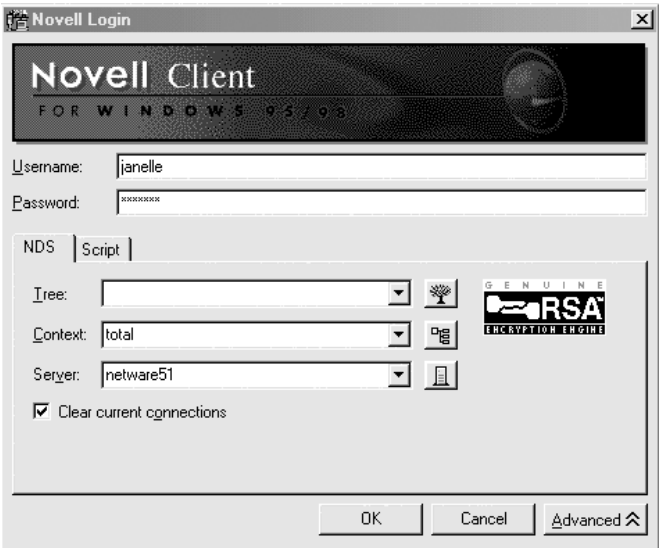


Figure 12-8 The NetWare login screen

account to the SALES group. Alice then assigns the ACCOUNTING group permission to access the accounting database and any other appropriate resources. By virtue of their membership in ACCOUNTING, Greg, Bobby, and Peter's user accounts have access to the ACCOUNTING group's resources, without Alice having to touch the individual accounts. If the company hires more accountants, Alice simply creates new user accounts and adds them to the ACCOUNTING group. Alice creates groups for whatever different work specialties her company employs, and then assigns user accounts to the appropriate groups. In large organizations with hundreds of employees who have similar needs, the time and effort saved becomes significant.

In most instances, a user account's rights are cumulative, that is, a user receives the sum total of the rights granted to his individual user account and the rights granted to any of the groups to which he belongs. Greg, for example, belongs to both the MANAGERS and ACCOUNTANTS groups. Suppose Alice sets up a shared folder on a server and assigns the MANAGERS group the right to add files to the folder, the ACCOUNTANTS group the right to read (but not alter or delete) files in that folder, and Greg (as an individual) the right to modify files that already exist in that folder. To see what Greg can do, add up the rights: Greg can add files (MANAGERS), read files (ACCOUNTANTS), and modify files (Greg) in that directory because of his cumulative individual and group rights.

NOTE In a server-based network, every server keeps its own list of user accounts, groups and permissions.

Server-based networks work great unless a network has more than one server. To use a server-based network with multiple servers, you must first have a user account on every server you want to use, and then you must log onto each one before you can use its resources. If your network only has a few servers, this isn't too much of a hassle for the user or the admin, but when the network has many servers, you've got an administrative nightmare once again.

Organization-Based Security Model

In an *organization-based network*, a single database acts as the logon point for all the shared resources of the network. This single source—I like to think of it as a database—stores at the minimum all of the user accounts and groups for the entire network. This database may reside on one computer, it may reside on one computer with one or more computers acting as a backup, or multiple computers might share complete copies of the database and constantly update each other through a process called *replication*. When a user logs on, that user's rights/permissions are checked against this database. A single logon defines the user's rights for every shared resourced on the network (see Figure 12-9). Different brands of network operating systems call this database by different names, but all of them work basically the same way.

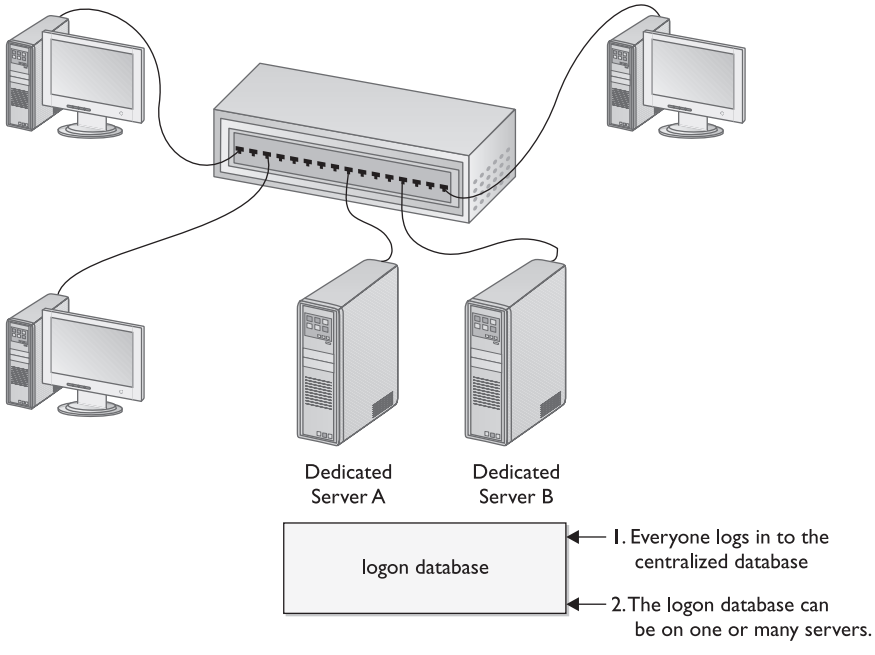


Figure 12-9 Everyone authenticates through the logon database

Organization-based security model networks simplify network administration by replacing multiple logins to individual servers with a single login that works for all the servers on the network. All of the modern operating systems use some form of organization-based security model. In fact, both Microsoft and Novell now use an even more advanced type of database called a directory. A directory goes beyond just providing authentication for the user accounts—a directory literally maps out the entire network. A good directory implementation describes every system, every printer, every user, and every group on its network, providing a central repository of all that is the network in one big database.

Mixing Models

As you will soon see, every operating system fits nicely into one of these three security models. Some operating systems may fit into one or another model, depending on how they are configured. Do understand, however, that these models can and do work happily together in one physical network. For example, networked Windows 98 systems that operate in a resource-based mode when communicating amongst themselves can also communicate in a server-based mode with a NetWare server on the same physical network. Microsoft did an amazing job enabling Windows systems to act as clients in networks running multiple brands and models of servers. Better yet (but confusing to new techs), all this complexity is hidden from the user. Figure 12-10 shows a screenshot of the My Network Places on a Windows XP system. There is a NetWare server, a Windows Server 2003, a Linux server, and a number of other Windows 2000 and XP systems. The NetWare and Linux servers look no different than a Windows system. If it weren't for the computer names, you wouldn't know one from the other.

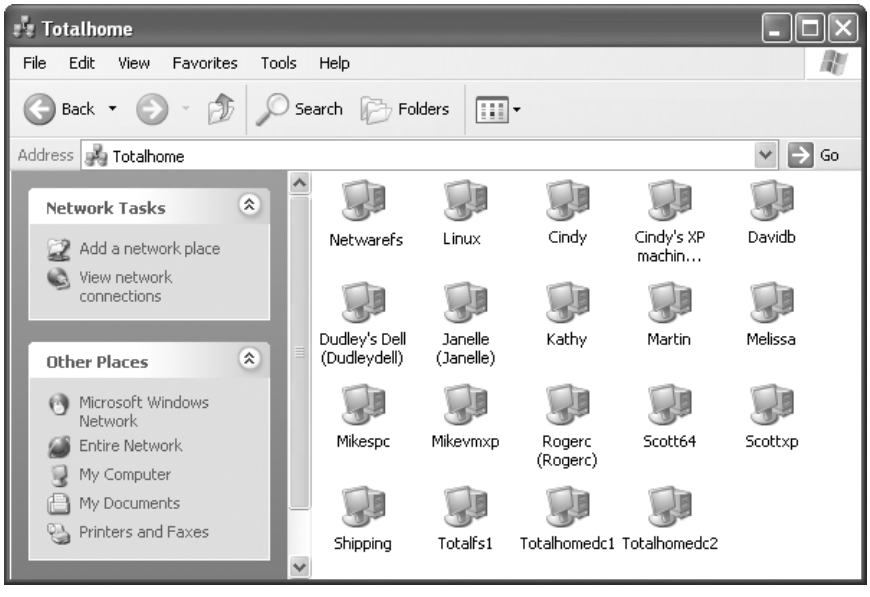


Figure 12-10 My Network Places

Client/Server and Peer-to-Peer Today

Okay, Mike, you say client/server and peer-to-peer no longer mean anything, but anyone who's into networking at all hears these terms now more than ever. So what do client/server and peer-to-peer mean in the context of today's networks?

Client/server and peer-to-peer have taken on new or updated definitions, and refer more to applications than to network operating systems. Consider e-mail for a moment. Most of us easily accept that for e-mail to work, you need an e-mail client like Microsoft Outlook Express. But you also need an e-mail server program like Microsoft Exchange to handle the e-mail requests from your e-mail client. Outlook Express is a *dedicated client*—you cannot use Outlook Express as a mail serving program. Likewise, you cannot use Microsoft Exchange as an e-mail client. Exchange is a *dedicated server* program.

Peer-to-peer applications act as both client and server. The best examples of these applications are the now infamous file-sharing programs based on special TCP/IP protocols. The applications, with names like LimeWire, BearShare, and Kazaa, act as both clients and servers, enabling a user both to share files and access shared files. Figure 12-11 shows one such program, Kazaa Lite, in the process of simultaneously uploading and downloading files.

The Dangers of Peer-to-Peer File Sharing

Peer-to-peer file sharing programs represent a creative use of TCP/IP protocols with powerful and useful implications and applications. On the other hand, the proven potential for abuse with these programs has made headline news around the world. So what's the big deal?

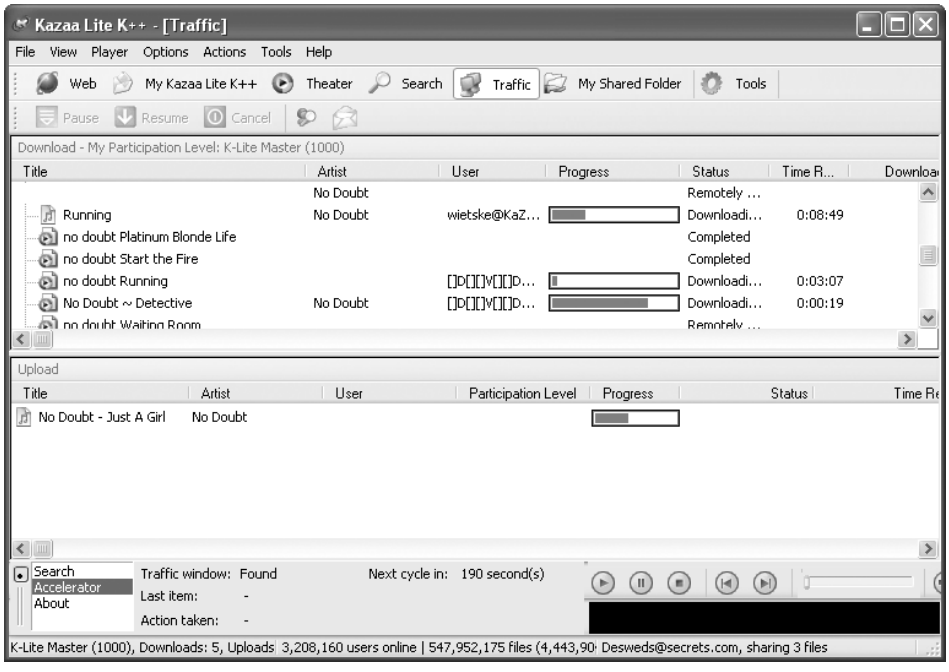


Figure 12-11 Kazaa Lite in action

On the plus side, peer-to-peer file sharing programs enable decentralized storage and distribution of many files. By storing on many serving systems, the loss of one or more of those systems means little in terms of the safety of the data being served. To bring a new system on line and get all the documents it needs for the user to be productive becomes a very simple operation.

On the negative side, many people have used peer-to-peer file sharing programs to flaunt intellectual property laws and steal substantial amounts of commercial music, videos, movies, and more. During the heyday of Napster and Kazaa, the theft was so commonplace that even otherwise upstanding citizens—who wouldn't dream of stealing a music CD from a record store—blithely stole hundreds of dollars of music every day!

To combat the loss of revenue, anti-piracy groups attacked in two ways. Some artists such as Madonna released bogus tracks onto the distributed networks. A search for a popular Madonna song, for example, will turn up a likely file; but when you click on it to play, you get Madonna calling you a dirty pirate (almost that plainly)! Second, record companies have gone after casual pirates through the legal system in the United States, suing people for revenue loss.

As a final word of caution, a lot of virus-infected files have made it into the distributed computing networks. This creates a dangerous situation for even legitimate uses and users of these networks. Use them at your own risk!

The Major Network Operating Systems

Microsoft, Novell, Apple and UNIX all provide strong network operating system solutions that address the goals of networking, including access to shared resources and security. Microsoft Windows dominates the client market (with some niche clients using Macintosh and Linux). Microsoft, Novell, Apple and UNIX compete for the server NOS market. In this section, I'll cover all the different variations of these network operating systems, and discuss a few of the more important aspects of each.

Microsoft Windows

Microsoft competes for NOS market share with two distinct Windows product lines that I'll call: the *Windows 9x family* and the *Windows NT family*. The Windows 9*x* family includes Windows 95, 98, 98 SE, and Me. The Windows NT family includes Windows NT, Windows 2000, Windows XP, and Windows Server 2003. Windows 9*x* functions as a flexible desktop operating system, capable of connecting to virtually any type of server. Windows NT, 2000, XP, and 2003 in contrast, can function both as powerful client systems and as full-featured server network operating systems.

Windows 9x

Microsoft Windows 9*x* systems provide basic file and print sharing functions, but little security by themselves. A network tech can configure a Windows 9*x* system as a client, or as both a client and a server. When operating as a server, however, Windows 9*x* uses a share-level security model, making it significantly less secure than more sophisticated server operating systems like Windows NT, 2000, 2003, Novell NetWare, and UNIX. Overall, Windows 9*x* has very weak security. Neither passwords nor user accounts provide much, if any, security in a pure Windows 9*x* network.

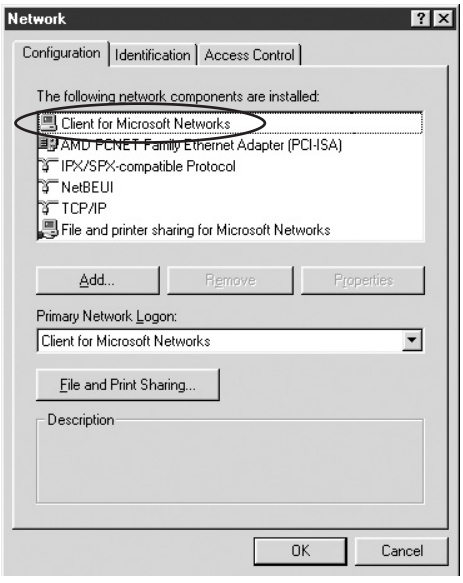


Figure 12-12 Client for Microsoft Networks is installed.

A network composed of only Windows 9*x* systems will always use NetBIOS, either over NetBEUI or over TCP/IP. NetBIOS will manifest itself with the Client for Microsoft Networks, as shown in Figure 12-12. You don't have to worry about installing it, however—Windows installs Client for Microsoft Networks automatically when it detects a modem or a NIC.

Speaking of automatic installation, one of the more interesting aspects between different versions of Windows 9*x* comes in the default clients and protocols that Microsoft installs. Windows 95 installed support for NetWare networks and NetBEUI by default as shown in Figure 12-13. TCP/IP had to be installed manually.

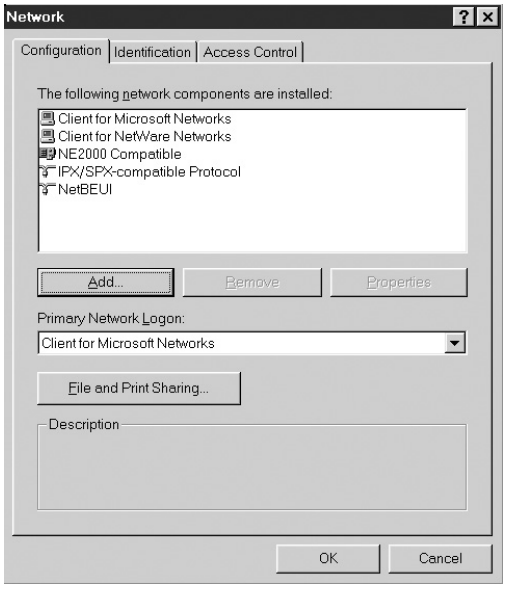


Figure 12-13 Windows 95 default Network Properties

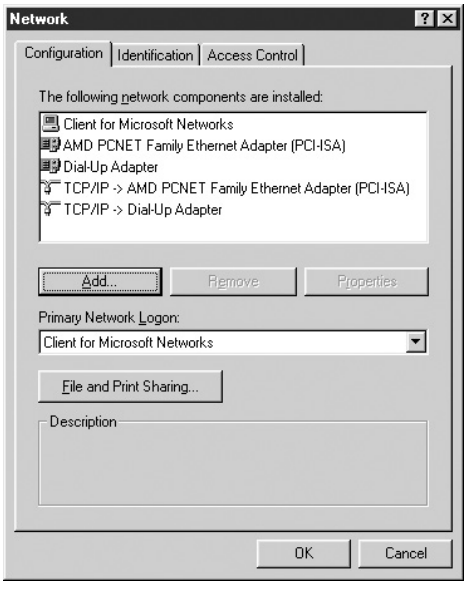


Figure 12-14 Windows 98 default Network Properties

Starting with Windows 98, Microsoft stopped installing NetBEUI and NetWare support and instead went to only the Client for Microsoft Networks and TCP/IP, as shown in Figure 12-14. All versions of Windows 9*x* still support IPX/SPX and NetBEUI, but the protocols require manual installation.

Windows 9*x* systems receive their NetBIOS names at installation, but these can be changed in the Identification Tab of the Network Control Panel applet. Figure 12-15 shows this tab.

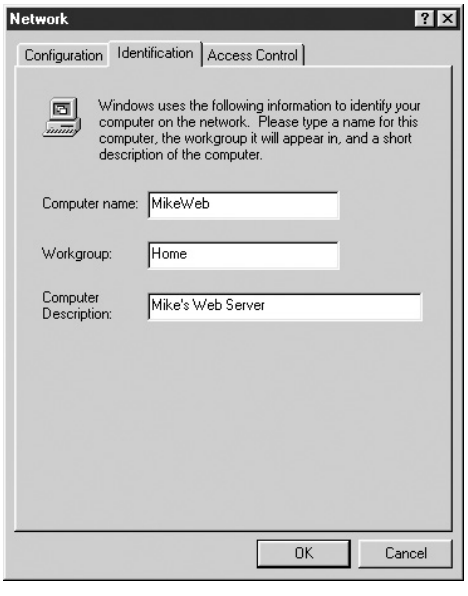


Figure 12-15 Windows 98 Identification Tab

Note the Workgroup setting in Figure 12-15. Windows 9*x* systems can be grouped into what are called *workgroups*. These workgroups have little purpose other than providing a way to organize slightly more complex networks. When you set up a Windows system, you give it a workgroup to join. Putting systems in workgroups makes it easier for other systems to find them in Network Neighborhood.

Again, workgroups don't really do anything other than organize. There's no security aspect to them that would stop an unauthorized user from accessing a workgroup or control what a user might do in a workgroup. Workgroups are actually a throwback to the early days of Microsoft networking, and have been replaced in more advanced Microsoft network operating systems with the much more powerful organization-based feature called a domain. Workgroups are still popular in Windows networks using a resource or server-based security model.

NOTE All later versions of Windows support workgroups.

One very interesting issue commonly seen in pure Windows 9*x* environments is a little phenomena called a *browser election*. Because Windows 9*x* systems rely exclusively on NetBIOS, a single computer in the workgroup must be the keeper of all of the NetBIOS names. This computer is called the Browse Master or Master Browse Server. Any computer running NetBIOS can become the Browse Master; the process used by the computers in the network to determine the Browse Master is called a *browser election*. Browser elections take place whenever any computer cannot detect a Browse Master on the network. All NetBIOS computers announce their name on the network every 12 minutes, so any time any one of those computers cannot get a response from a Browse Master, a browser election takes place. A browser election slows down a Windows 9*x* network, sometimes quite dramatically, so you should reduce the occurrence of browser elections whenever possible.

The best way to remove this issue is to use a WINS server (see Chapter 14), but WINS only works with NetBIOS over TCP/IP. Another way to reduce browser elections, no matter what protocol NetBIOS is running on top of, is to go into the properties of the File and Printer Sharing for Microsoft Networks service. This service must be installed on a Windows 9*x* system for it to share folders and printers. Normally this is a service that is loaded and forgotten, but there is a handy setting in its properties that can substantially reduce browser elections. Figure 12-16 shows the File and printer sharing for Microsoft Networks Properties dialog box for a Windows 98 computer.

The File and printer sharing for Microsoft Networks Properties dialog box offers two settings: Browse Master—the one that's relevant here—and LM Announce. The Browse Master setting determines if this machine will attempt to become a Browse Master during an election. Turn this to Disabled for all the systems in your Windows 9*x* network except one—and on that machine set it to Enabled. The one machine with Browse Master set to Enabled will always be the Browse Master, eliminating any future browser elections. Be warned: this is a bit risky. Make sure that your designated Browse Master is always on the network. If that one machine ever goes off the network your computers won't be able to

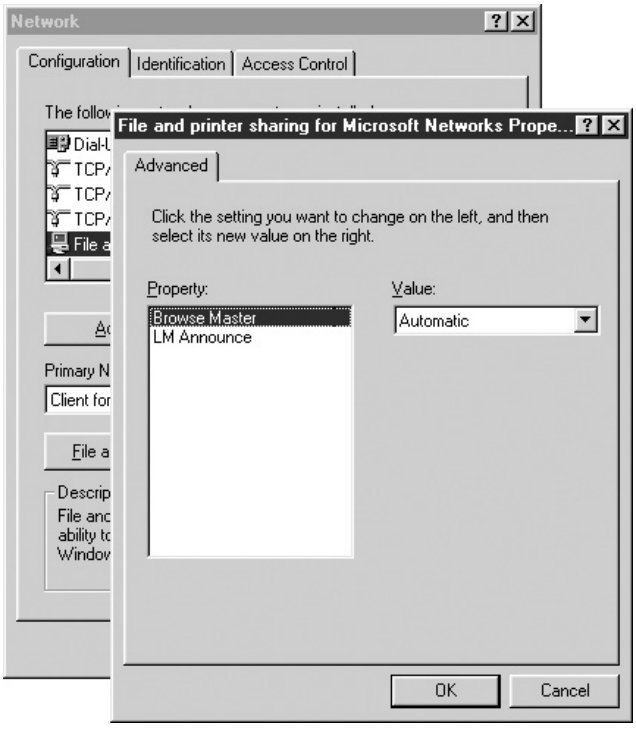


Figure 12-16 File and printer sharing for Microsoft Networks Properties

browse the network. If you want to play it safe, turn the Browse Master setting to Enabled on a second machine.

LM Announce is a virtually useless setting designed to enable a modern Windows system to work with any system running a very old form of Microsoft networking called LAN Manager. Unless you've got some ancient DOS machines in the corner, turn this setting to No.

NOTE The LM Announce setting can sometimes bite modern networkers when they least expect it. Ever have a Windows 9*x* system that never seems to shut down completely when you run Start | Shut Down? It's often due to the LM Announce setting set to Yes. Turn it to No and see if that fixes the problem!

The Windows 9*x* resource-based security model and complete dependence on NetBIOS makes it unacceptable for any but the smallest networks. Yet, a Windows 9*x* network, especially when running NetBEUI was so easy to set up that anyone could install a perfectly acceptable small network. Microsoft's ongoing desire to make the user's life as easy as possible had created a monster. NetBIOS made networking available to the world and Microsoft had to support it, even up to today's latest versions of Windows.

Windows 9*x* may not make a very robust network solution alone, but all versions of Windows 9*x* do an excellent job acting as clients in more advanced network operating

systems running Windows NT, 2000, Server 2003, NetWare and UNIX. As we investigate these more powerful network operating systems, we'll return to Windows 9*x* to see what you need to do to make a Windows 9*x* system work with these network operating systems.

The Windows 9*x* line of products may have ceased production with the introduction of Windows XP, but given that the installed base of Windows 9*x* systems was in excess of 180 million copies late in 2001, you can rest assured that Windows 9*x* will continue to be an operating system you need to understand for years to come.

TIP You can technically still purchase older versions of Windows. But don't bother asking your local computer builder for a new system with Windows 98! These purchases are done through *channels*—that's Microspeak for calling Microsoft directly and begging.

Windows NT

When Microsoft developed Windows NT in the early 1990s, they chose to make two very different versions: Windows NT Workstation and Windows NT Server. Windows NT Workstation was marketed as the high-end desktop operating system and contained a number of underpinnings to give NT Workstation incredible network support. Windows NT Server had all the power of Windows NT Workstation, plus Microsoft added a number of server tools not found on NT Workstation such as DNS, WINS, and DHCP servers, as well as support for an organizational-based security model called a domain.

Windows NT Workstation

Windows NT Workstation offered the same user interface as Windows 95 but with greatly enhanced security and stability as compared to the weak Windows 9*x* security. First, Windows NT Workstation used a server-based security network model. If a user wanted to access anything on a Windows NT Workstation system, he or she had to have a user account and a password for that system. This was true whether you logged on at the machine or if you wanted to access a shared resource from another system. Windows NT workstation also used a new file system called NT File system (NTFS). NTFS gave tremendous control on how users and groups could use shared folders and files within shared folders. With NTFS, you could define permissions such as Modify (Change a file or the contents of a folder), List Contents (Define whether users or groups could see a file or the contents of a folder), and Read (Define whether users could open a file).

NOTE There were three versions of Windows NT: 3.1, 3.5 and 4.0. This section discusses the only popular version, NT 4.0. The 3.1 and 3.5 versions were very early and have disappeared from the market. When the Network+ exam talks about Windows NT, it means version 4.0.

Windows NT provides native support for NetBEUI and TCP/IP, as well as strong security by using robust user accounts. You cannot log onto a Windows NT system without a valid user account. Every Windows NT (as well as every Windows 2000, XP and 2003) operating system comes with a special "super user account" called *Administrator*. Anyone

who logs in using the administrator account of a Windows NT, 2000, 2003 or XP system has complete and total control over the entire system. Clearly, very few people should ever have access to the administrator account!

NT User Accounts

Back in the days of Windows NT Workstation, creating a user account or a group meant a trip to the NT User Manager. Figure 12-17 shows a screen from User Manager in a newly installed NT Workstation system. Note the two preinstalled accounts: Administrator and Guest. The Guest account, which first appeared in Windows NT and continues in every version of Windows since, is a very basic account with very limited permissions.

At the bottom of the User Manager dialog box are the Windows built-in groups. Windows NT came with six built-in groups. Windows 2000, XP, and 2003 have seven such groups. In any Windows OS, you cannot delete these built-in groups.

- **Administrators** Any account that is a member of the Administrators group has complete administrator privileges. It is common for the primary user of a Windows 2000 or XP system to have his or her account in the Administrators group.
- **Power Users** Power users are almost as powerful as administrators, but they cannot install new devices or access other users' files or folders unless the files or folders specifically provide them access.

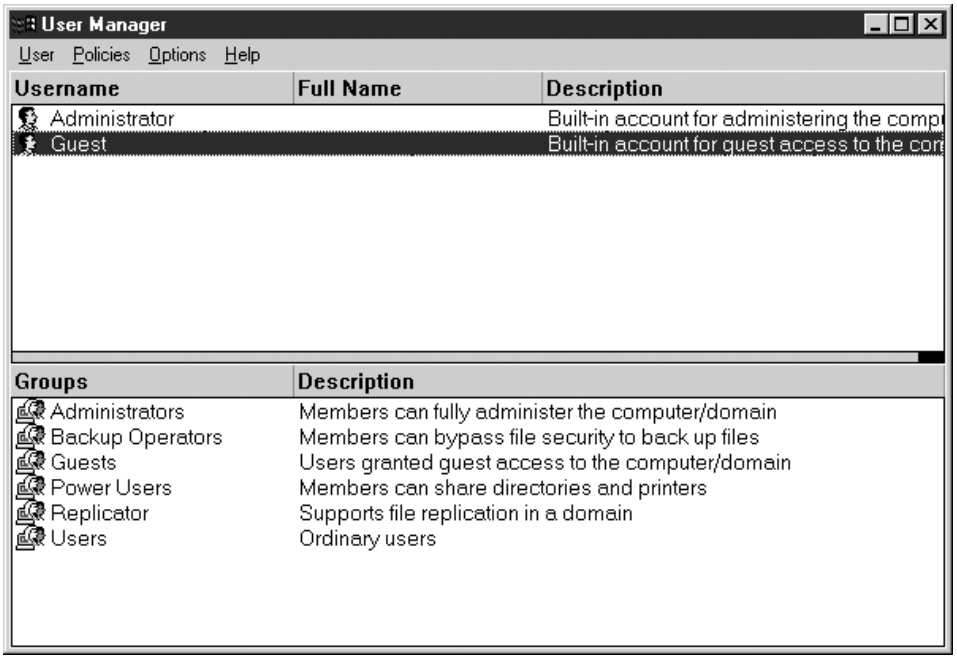


Figure 12-17 User Manager in Windows NT Workstation

- **Users** Users cannot edit the Registry or access critical system files. They can create groups, but can manage only those groups they create.
- **Backup Operators** Backup operators have the same rights as users, but they can run backup programs that access any file or folder—for backup purposes only.
- **Replicator** Members of the Replicator group can replicate files and folders in a domain.
- **Guests** Someone who does not have an account on the system can log on using the Guest account if the system has been set up to enable that feature. This group is used in certain network situations.
- **Everyone** This account (which wasn't in Windows NT but appears in 2000, XP, and 2003) applies to any user that can log onto the system. You cannot edit this group.

Windows NT Workstation worked beautifully in a network environment except for one little nasty—every Windows NT Workstation system ran a server-based security model. This meant that if you wanted to access another Windows NT Workstation system, you had to have a user account on that other system. The accounts for each NT Workstation system are known as the *local user accounts*. Calling a user account a local account wasn't obvious in NT's User Manager for accounts, but NT Workstation also had local groups, which were far more obvious. Figure 12-18 shows the creation of a local group. Note how NT Workstation called the local groups "local groups," but only called the local users "users"—a strange aspect of NT that was corrected in later versions of Windows.

By default, all Windows systems use the login name and password you use when you first start a system to try to access network resources. Let's say you have a Windows NT Workstation system and you used the username "Betsy" with the password "b3tsy232" when you logged in. If you then try to access another Windows NT Workstation system via Network Neighborhood, you'll be prompted for a login name and password for that system, unless that other system just happens to have an account with the username "Betsy" with the password "b3tsy232!"

Figure 12-18 Creating a local group in Windows NT

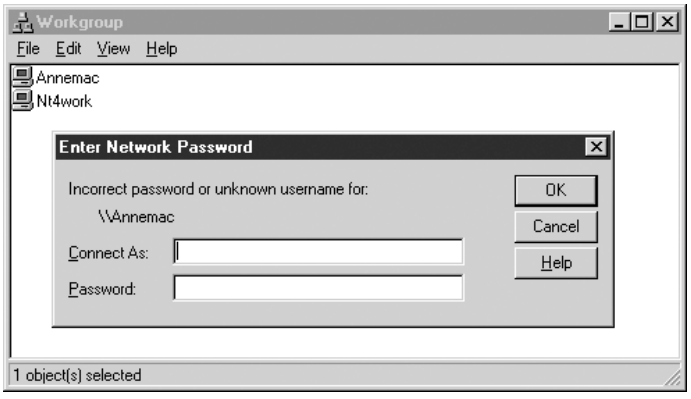


Figure 12-19 Prompting for a logon on another system

Note in Figure 12-19 that the Windows NT Workstation computer is part of a workgroup called Workgroup. A Windows NT (or 2000/2003 or XP) computer that's on a network must either be part of a workgroup or part of something far more powerful—a domain.

Windows NT Server

Windows NT Server has the ability to transform a group of individual Windows computers, each with its own local users and groups, into a organization-based model called a *domain*.

A *domain* functions like a workgroup, but has all the security centralized on a single server. In Windows NT Server, the system that held that central spot was called a primary domain controller (PDC). Any Windows NT network could have only a single PDC, but you could add one or more backup domain controller (BDC) computers also running Windows NT Server to provide some redundancy in case the PDC went down.

During the installation of Windows NT, you were prompted for what Microsoft called the *role* of the machine. A Windows NT Server system could just join a workgroup and handle its own local users and groups; it could create a new domain and act as the PDC of the domain; it could join an existing domain and act as a BDC; or it could join a domain, but not act a PDC or BDC. Once the role of the server was defined, it could not easily be changed.

The creation of the Windows NT domain concept made for a bit of a problem when you had both Windows NT Server and Windows NT Workstation systems in the same domain. Remember, each Windows NT Workstation system has its own local users and groups. But when you created a domain, the PDC now had its own set of user names and accounts that were for the entire domain. This created a situation that still exists even today: dual sets of user accounts and groups. The users and groups created on the NT Server PDC were called *global users and groups*.

Figure 12-20 shows the problem: each computer has its own local users and groups, while the entire domain has its domain users and groups. Microsoft skirted this issue by creating a dual logon. In Windows NT—and every other version of Windows—you may either log onto the domain or log on locally.

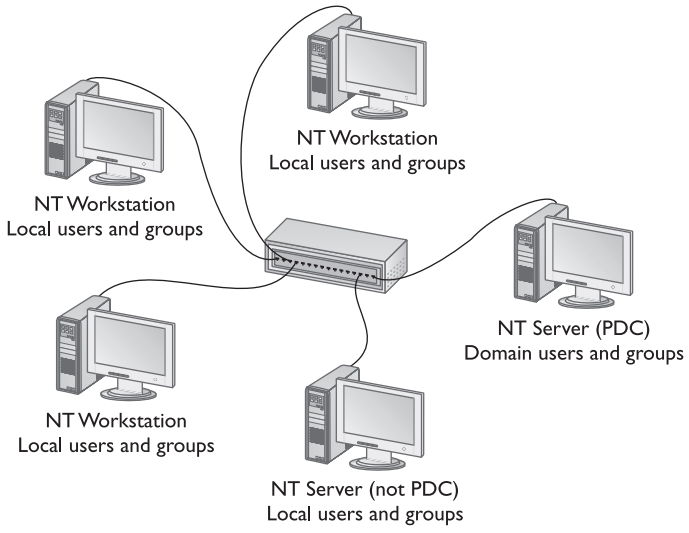


Figure 12-20 Local and Domain users in the same network

Figure 12-21 shows the logon screen of a Windows NT Workstation computer named NT4WORK before it joins a domain. Figure 12-22 shows the same system's logon screen after it joins a domain called TOTALHOME. Note that if the system is part of a domain, you are given the choice to log onto the domain or to just log on locally to the system.

One interesting point to note in Figure 12-22 is that both the local login and the domain login choice are listed under *Domain*. Later versions of Windows would fix this!

Creating global users and groups in Windows NT Server required a different utility, called User Manager for Domains. User Manager for Domains was only found on Windows NT Server systems running as a PDC or BDC—if you wanted to create a global user

Figure 12-21 NT4WORK before domain

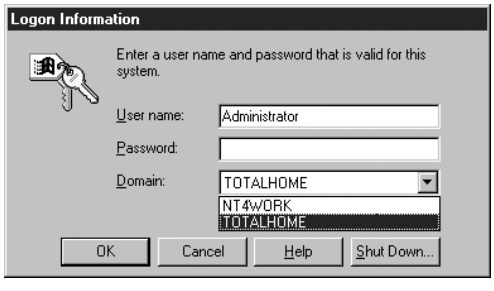


Figure 12-22 NT4WORK after joining the domain TOTALHOME

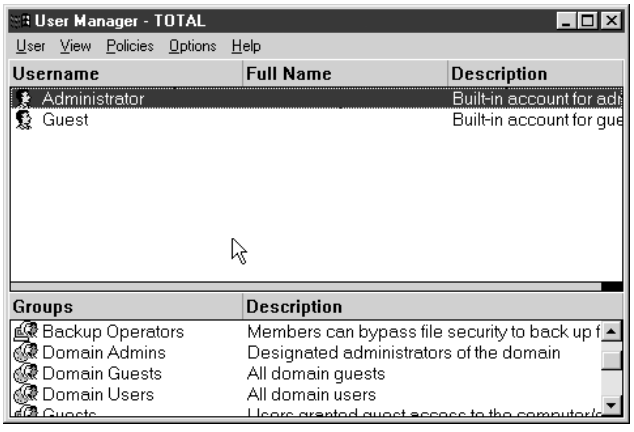


Figure 12-23 User Manager for Domains

or group, you had to sit in front of a Windows NT PDC or BDC to create these accounts. Figure 12-23 shows User Manager for Domains.

At first glance, User Manager for Domains looks identical to User Manager in Windows NT Workstation. If you look down at the default groups, you'll notice that there are two sets of default accounts: one for the domain and one for the local computer. That's right—even the Windows NT Server systems that were part of the domain had local users and groups! The domain groups have a slightly different icon than the local groups. Windows NT has three built-in domain groups.

- **Domain Admins** Any account that is a member of this group has complete administrator privileges to the entire domain.
- **Domain Guests** Accounts assigned to this group are similar to local guest group accounts, but they span the entire domain.
- **Domain Users** This group includes all users who are part of the domain.

This dual-groups-and-users idea is still with us today in the latest versions of Windows. If you use a Windows domain with Windows 2000 Server or with Windows Server 2003, you'll always deal with global and local users and groups. To simplify working in this fashion, Microsoft long ago came up with some rules that are now in common use.

- Don't make local groups for users in a domain-based network. All users on the domain should have only global user accounts. Do not create local users or groups for people.
- Global users go into global groups. Make your groups (like ACCOUNTING or DALLAS) global groups.

Working with NT

Configuring a network on an existing Windows NT (Workstation or Server) PC meant a trip to the Network applet in the Control Panel. The Network applet enabled you to configure your network name, the domain or workgroup to which you wished to join the system, and all protocol and NIC settings (Figure 12-24).

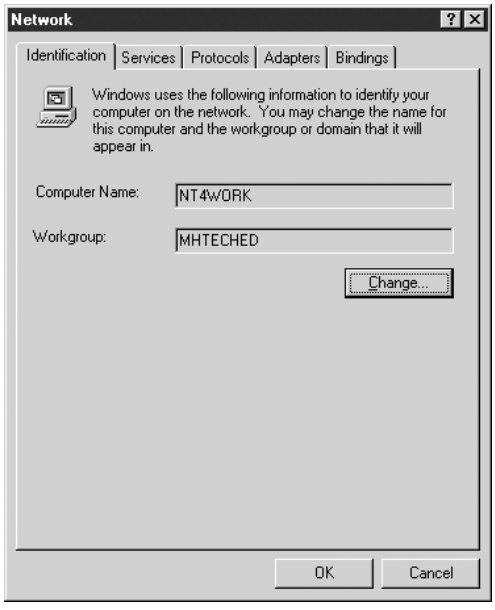


Figure 12-24
NT Network applet dialog box

To change a computer's name, domain or workgroup membership, you clicked the Change button on the Identification tab (Figure 12-25). To add a computer to a domain, you also had to use the administrator account.

Adding, editing, or deleting a protocol required a trip to the Protocols tab. Figure 12-26 shows the TCP/IP Properties dialog box. Note that this copy of Windows NT also has the NetBEUI and IPX/SPX protocols installed.

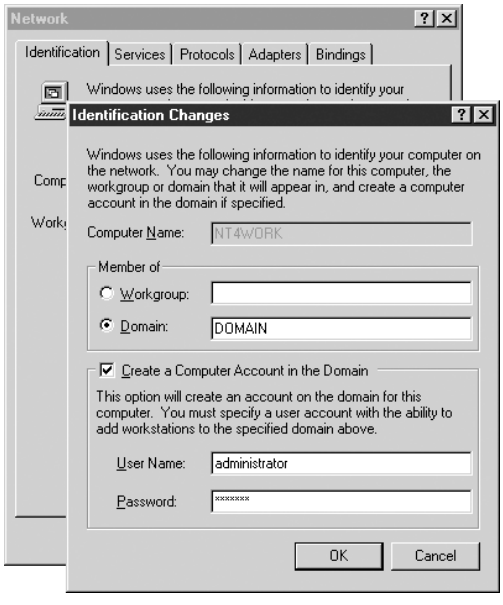


Figure 12-25
Changing Identification in NT

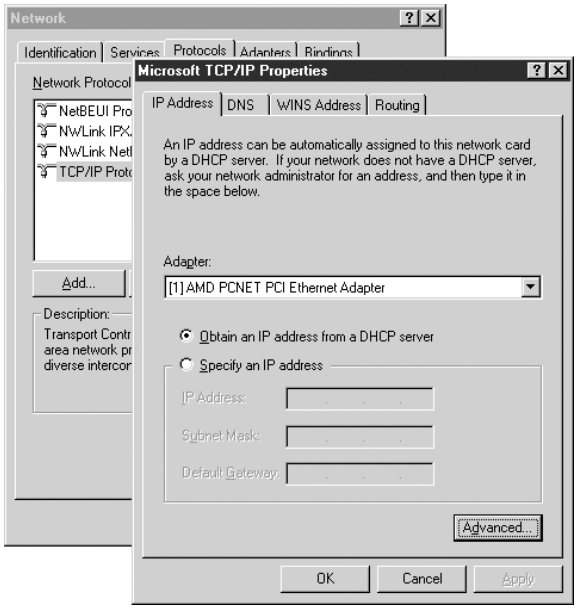


Figure 12-26 Changing IP settings in NT

NOTE Windows NT does not have Device Manager! If you wanted to install, configure, or delete a NIC you had to go to the Network applet!

Windows 2000

Microsoft improved on the Windows NT family with the Windows 2000 generation of operating systems. Like NT, 2000 came out in a desktop and a server version, called Windows 2000 Professional and Windows 2000 Server. Windows 2000 combined the Windows 98 user interface with the underlying power of Windows NT. Windows 2000 was virtually identical to Windows NT in terms of networking, security and users, but had the more up-to-date features of Windows 98, like better driver support, Device Manager and Plug and Play. Windows 2000 only supported TCP/IP natively, although through extra configuration it supported NetBEUI, IPX/SPX, and AppleTalk.

TIP Windows 2000 Server came out in three different versions: Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 Datacenter Server. For the specifics on the differences among these versions, head over to Microsoft's Windows 2000 site at http://www.microsoft.com/windows2000/default.asp.

Some of the biggest differences between 2000 and NT come in the networking arena. Microsoft, having watched the Internet grow around NT's NetBIOS-centric networking and tired of trying to come up with one method after another of keeping NetBIOS working in a world that was moving towards DNS is droves, totally redesigned their domains. The old NT NetBIOS domain names gave way to domains based on DNS names. Even in

a small network that's not part of the Internet, Windows 2000 domain names now have the dotted DNS naming scheme, such as *server.totalhome*. The last, and probably the biggest change in Windows networking came about with the introduction of a new superdomain called Active Directory. We discuss Active Directory shortly.

Windows 2000 Professional

Microsoft never liked being in the two operating system business, but by running the Windows 9*x* line alongside the Windows NT line of operating systems, that's exactly what they did. Microsoft didn't have much choice—NT had heavy hardware requirements and wasn't backwardly compatible with a number of older Windows programs. Windows 9*x* was very much backwardly compatible but was also showing its age in terms of outdated file systems (FAT), virtually non-existent security, and reliance on 16-bit code. Windows 9*x* needed replacing, but not at the cost of too many older systems.

Windows 2000 Professional was Microsoft's first attempt at replacing the old Windows 9*x* systems. Unlike NT, Windows 2000's use of Plug and Play, excellent hardware support, and an improved user interface made it a good replacement for many systems that used to run Windows 9*x*. Windows 2000 Professional still had heavy hardware requirements and couldn't totally displace Windows 9*x*. It would take Windows XP before Microsoft could officially declare Windows 9*x* obsolete.

For all of the improvements of Windows 2000, the OS retains many of the problems inherent to Windows networking. All Windows 2000 Professional systems still have local users. A group of Windows 2000 Professional computers will use a server-based organization model. If you log onto one Windows 2000 professional system and want to access another Windows 2000 Professional system via My Network Places, you'll need a separate local account on that system, just like we did in the Windows NT days.

The big change in networking comes when you add a Windows 2000 Server to your network and set up Active Directory!

Windows 2000 Server and Active Directory

Directory services are centralized storage areas for information about a network's resources, including users, applications, files, and printers. Directory services applications enable network administrators to centrally manage and share information about their networks' users and resources, and to centralize network security authority. Not until Windows 2000 did Microsoft finally create an NOS with directory services. Windows 2000's directory services are called Active Directory. All of the domain functions of Windows NT still work—they've just been incorporated into Active Directory. Just as in the NT days, a computer must be a member of either a workgroup or a domain. A single Active Directory consists of one or more domains. If you want a computer in the Active Directory, it will by default be in a domain.

NOTE A single Active Directory consists of one or more domains.

Windows 2000 Server dumps the idea of PDCs and BDCs. Instead, all of the domain controllers (DCs) are equal. If you create a user on one DC, it will automatically replicate

the new user information to all of the other DCs in the Active Directory. Active Directory domains are true DNS domains. In fact, all of the DNS data is built into the Active Directory itself.

When you install a Windows 2000 Server system, you eventually reach a screen that prompts you for the function that this server will perform. It will either be a domain controller, a member server (part of the domain but not a domain controller) or, if for some reason you didn't want to join a domain, a stand-alone server.

One of the most obvious places that Windows 2000 Server differs from Windows 2000 Professional is in the Administrative Tools. Windows 2000 Server includes every type of serving software necessary to run a Windows network, including DNS, WINS, and DHCP Servers. Figure 12-27 shows the standard Administrative Tools in Windows 2000 Professional. Compare that to a fairly typical set of Administrative Tools in Windows 2000 Server in Figure 12-28.

Figure 12-27 Administrative Tools in Windows 2000 Professional

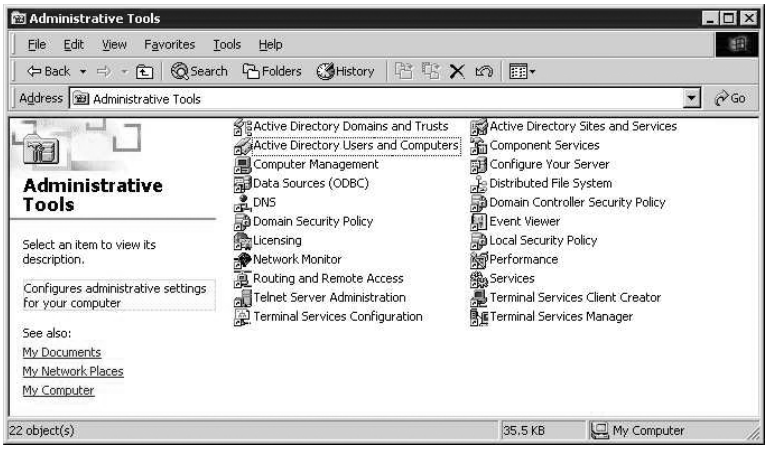


Figure 12-28 Administrative Tools in Windows 2000 server

There's no way to see the entire Active Directory, but there are some applications on Windows 2000 Server that give you a glimpse of parts of the Active Directory. Figure 12-29 shows the Active Directory Users and Computers utility. This handy program does many jobs, but it's most commonly used to create domain-level users and groups.

Note the name totalhome.local in Figure 12-26. Totalhome.local is a true DNS name—so why doesn't it end with ".com" or ".net" as we might expect? The ".local" shows that this domain is not open to the Internet. Of course any system on this domain may access the Internet, but none of these machines may act as an Internet web server, FTP server, or any other type of Internet server unless special security steps are taken.

Windows 2000 Server supports TCP/IP natively, but through extra configuration can support NetBEUI, IPX/SPX, and AppleTalk.

Working with 2000

One handy improvement in Windows 2000 over Windows NT is the consolidation of most all of the utilities you need into one handy tool called Computer Management. You can access this tool via your Control Panel, but most techs just alternate-click the My Computer icon and select the Manage menu option (Figure 12-30).

Computer Management is your one-stop shop for creating local user accounts and groups, accessing Device Manager, locating shared resources, and disk management. If you need to make any changes to your NICs, protocols, or network services, however, you still need to fire up your Network applet in the Control Panel. Windows 2000 calls this applet Network and Dial-up Connections (Figure 12-31).

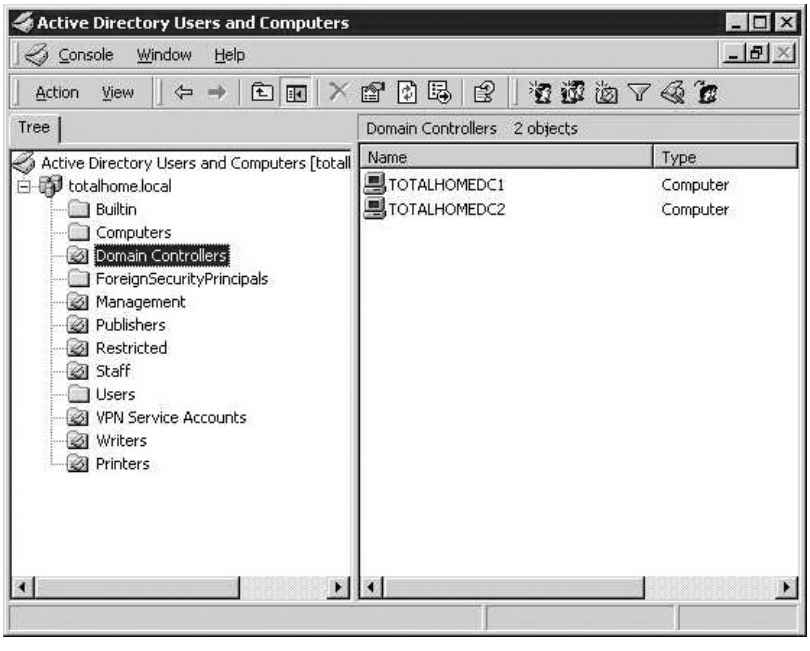


Figure 12-29 Active Directory users and groups

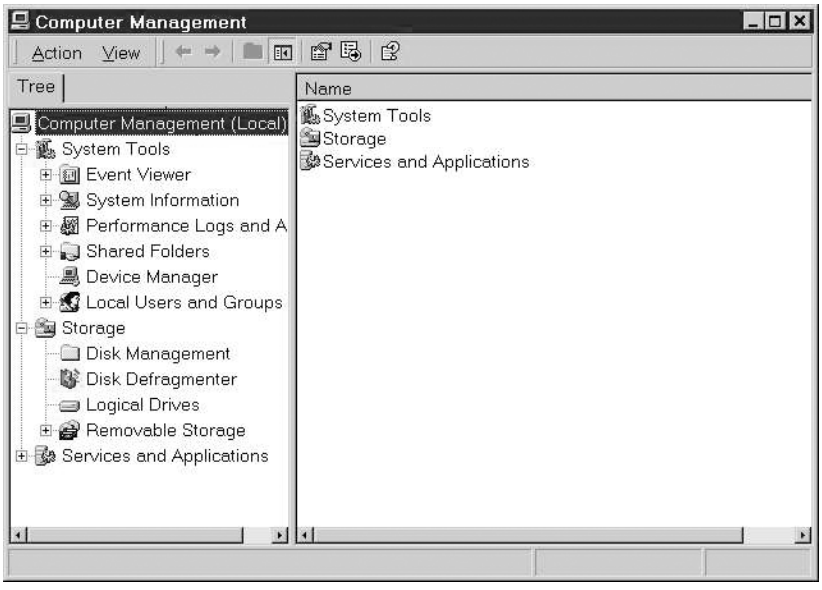


Figure 12-30 Computer Management in Windows 2000 Professional

Note that the computer displayed in Figure 12-31 has three connections: the 100BaseT is my main wired network connection; the Emergency Dialup is a backup dialup I can use if my main network goes down; and the Bluetooth enables this machine to connect to any Bluetooth devices. The number of devices you see on a system is simply a matter of the number of devices in that system.

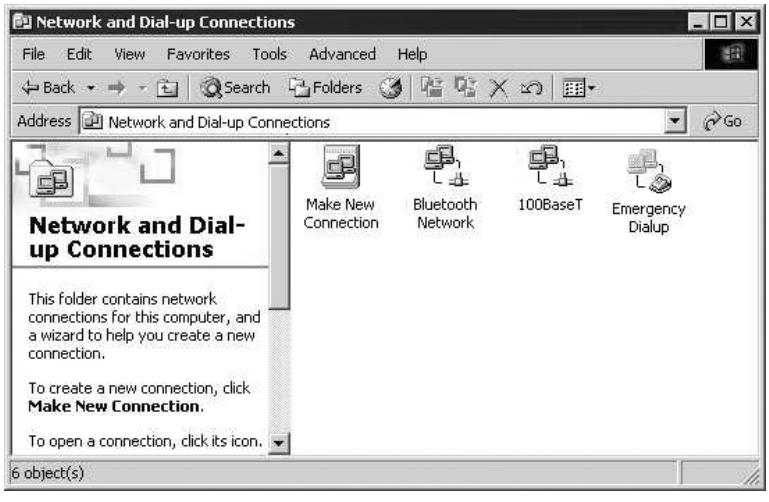


Figure 12-31 Network and Dial-up Connections

NOTE You can also access Network and Dialup Connections directly from the Start menu by selecting Settings | Network and Dialup connections.

Changing the name of your computer or changing the workgroup or domain membership is also very different in Windows 2000 compared to Windows 9*x* or Windows NT. Making any of these changes requires you to alternate-click on My Computer and select Properties to open your System Properties dialog box (alternatively you can select the System Control Panel applet). Click the Network Identification tab to see the current network settings, as shown in Figure 12-32.

You'll see two buttons: Network ID and Properties. Each of these buttons does the same job—change the computer name as well as the workgroup or domain membership. The Network ID button starts a handy wizard to walk you through the steps while the Properties button just brings up a dialog box to make the changes without a wizard (Figure 12-33).

Don't try changing the name of a Windows 2000 Server running as a domain controller this way! If you go over to your Network Identification screen on one of those machines, you'll see something like Figure 12-34, showing you that you cannot change the name of the Server system. Windows 2000 domain controllers are well named—they control the domains—so changing their names or domain membership takes a rather involved process. Note that this restriction applies only to 2000. Windows Server 2003 enables you to change the name of domain controllers.

In general, working with Windows 2000 in terms of network configuration is fairly straightforward—as long as you remember where to go to make those changes! The other nice part about knowing how to configure network settings in Windows 2000 is that it makes it easy to configure network settings in Windows XP—it's almost exactly the same.

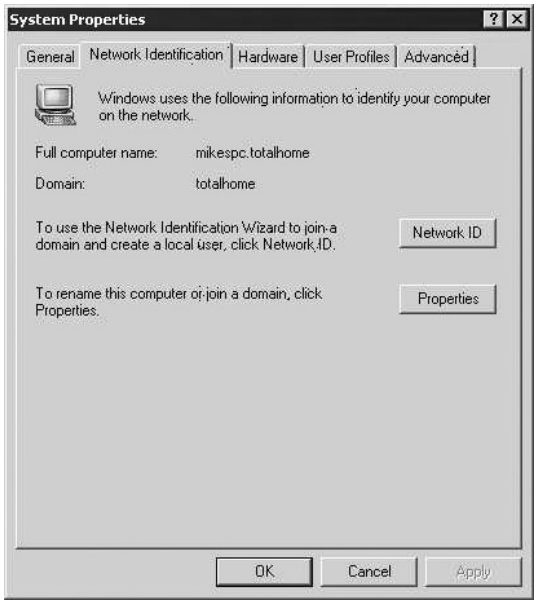


Figure 12-32 Network Identification tab in Windows 2000 Professional

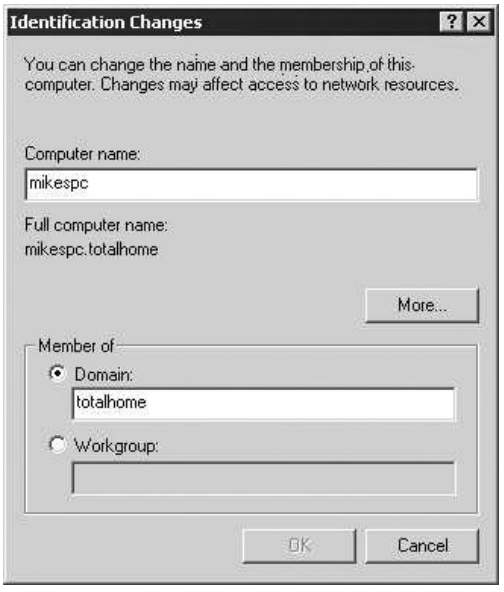


Figure 12-33 Network Identification changes

Windows XP

Microsoft touts Windows XP as the unifying operating system, bringing together the power of NT/2000 with the backward compatibility of Windows 9*x*. This claim might be open to argument. Windows XP, underneath its slick user interface, slightly improved tools for backward compatibility with older programs, and a number of built-in tools like a CD burner and support for .zip files, is little more than a spiffed-up version of Windows 2000 Professional.

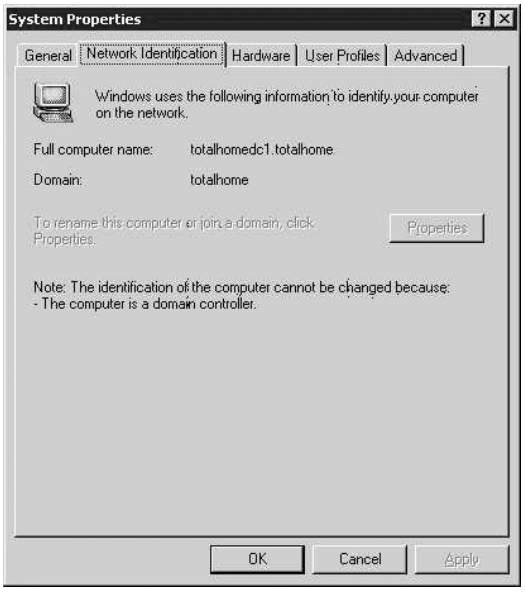


Figure 12-34 Network Identification on Windows 2000 domain controller

Windows XP has no Server version like we see with Windows NT and Windows 2000. There are however, two versions of Windows XP—Windows XP Home and Windows XP Professional, but these are both user versions. Windows XP Professional is designed to work in domain environments and has all the power and security of Windows 2000 Professional. There are no differences between Windows XP Professional and Windows 2000 Professional in terms of where you go to make any network configuration changes. To add, edit, or remove protocols and services, you go to your Network Connections Control Panel applet. Each network connection in Windows XP manifests as a separate icon, just as in Windows 2000. If you want to change the name of a system or change its workgroup or domain membership, you go to System properties. If you know how to configure a network in Windows 2000, then you know how to configure a network in Windows XP Professional, although a few names may be changed along the way. Let's say you want both a Windows 2000 and an XP Professional PC to join the MHTECHED.LOCAL domain. In both cases you select the System Control Panel applet. In Windows 2000 you click the Network Identification tab. In Windows XP you click the Computer Name tab (Figure 12-35).

Compare this figure to Figure 12-33 (earlier in this chapter); they are virtually identical. If you click on the Change button in XP you'll see that it is virtually identical to Windows 2000. Figure 12-36 puts these two dialog boxes next to each other for comparison.

Windows XP Home Edition, as its name implies, is a greatly simplified version of XP designed for home and small office users that do not need the same complex security features found in Windows XP Professional. In fact, Windows XP Home Edition is crippled so that it *cannot* join a Windows domain. If you access the Network Identification properties on a Windows XP home system, you'll see that there is no mention of a domain (see Figure 12-37).

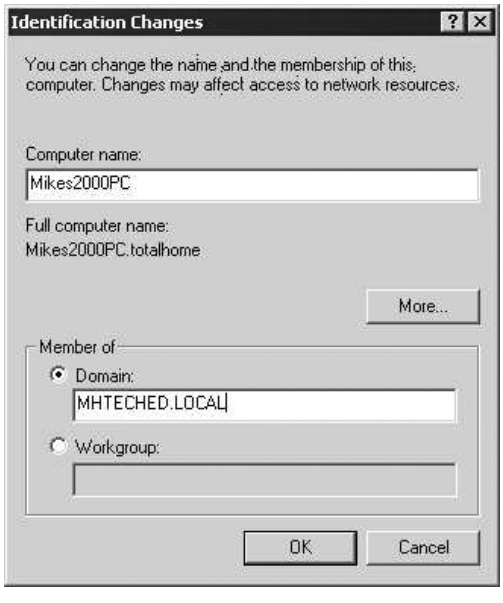


Figure 12-35 Computer Name tab in System applet

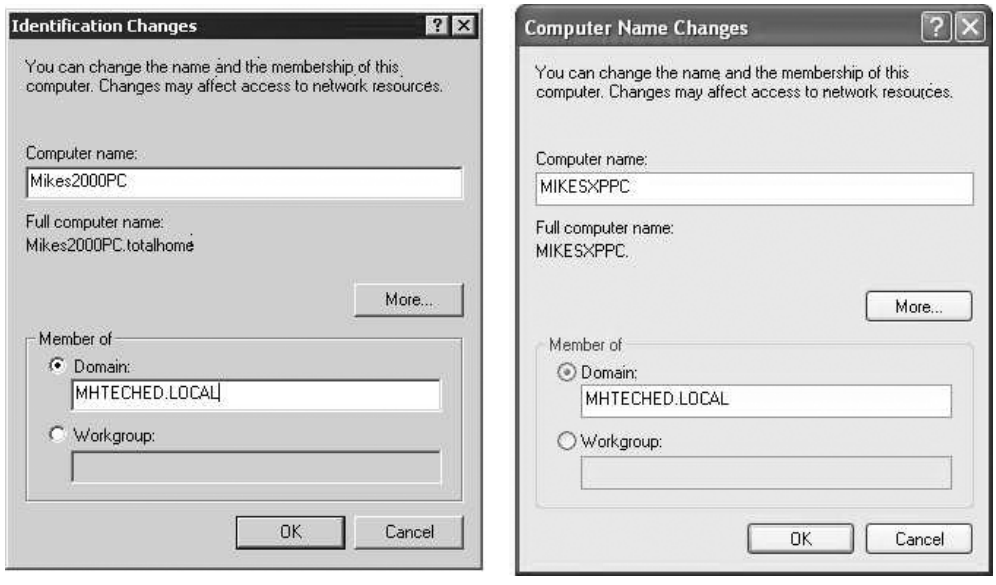
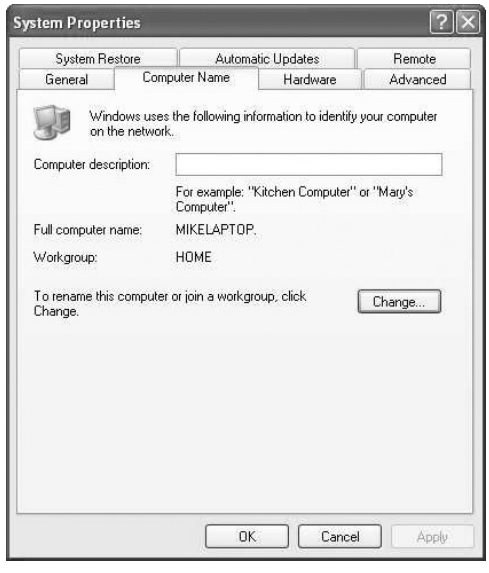


Figure 12-36 Networking's the same in Windows 2000 and Windows XP Professional.

Figure 12-37 No domain in XP Home!



Windows Server 2003

Windows Server 2003 (note that we do not say Windows 2003 Server!) is Microsoft's current server version. Windows Server 2003 is virtually identical to Windows 2000 Server. With the exception of a few changes to the interface and some rather handy utili-

ties, only advanced network technicians would notice the difference between these two network operating systems. Describing the differences is completely outside the scope of this book. Windows Server 2003 uses the same Active Directory, domain naming, services, and interfaces used in Windows 2000 Server.

User Profiles

All versions of Windows support the use of *user profiles*, which enhance both the usability and security of a network. A user profile is a collection of settings that corresponds to a specific user account and follows the user to any computer she uses on the network. User profiles enable users to customize their working environments. The server checks the user profiles to determine each user's wallpaper, desktop layout, and other environment preferences. Each time the user logs onto the network, the client system retrieves the profile and displays the OS accordingly. Here's an example.

Roger, Chris, and Cindy work different shifts and share the same Windows XP computer. When each of them logs onto the computer at the beginning of their respective shift, Windows XP loads the appropriate configuration from their profile. If the profiles exist on the local hard drive, they only affect that computer. But a savvy network administrator will store the profiles on a network server, enabling the profiles to follow the users regardless of where they sit. When Roger transfers to the day shift, he can use a different computer and still enjoy all of his customized settings. As much as Roger, Chris, and Cindy enjoy the benefits of user profiles, Martin, the network admin, likes them even more. Martin can use profiles to place restrictions on how Roger, Chris, and Cindy use their computers. When their boss, Dudley, complains that employees spend too much time playing Unreal Tournament 2004, Martin edits their profiles so they cannot run the Unreal Tournament program anymore. Martin can also restrict their use in other ways, to prevent them from doing the following:

- Running other programs
- Changing their desktop icons and wallpaper
- Loading new programs

User profiles offer a consistent look and feel to the end user, and control to the network administrator.

TIP A profile is a set of configuration settings specific to an individual user. Profiles can be stored locally or on a server. Administrators can use profiles to place restrictions on what users can do with their computers.

Novell NetWare

The continued use of older versions testifies to the power and stability of Novell NetWare. Many organizations upgrade their client software, but continue to use their existing NetWare 3.*x* and 4.*x* servers, following those ancient words of wisdom: "If it ain't broke, don't fix it!" Network techs should familiarize themselves with three significant versions of NetWare: NetWare 3.*x*, NetWare 4.*x*, and NetWare 5.*x*.

TIP The Network+ exam assumes all NetWare Networks use IPX/SPX unless specifically stated otherwise.

NetWare 3.x and the Bindery

NetWare 3.*x* offers solid file and print sharing capabilities using the IPX/SPX protocol suite, but lacks a centralized security database. Each NetWare 3.*x* server maintains its own security database, called the Bindery. When a user logs in, the NetWare server compares the username and password to its Bindery database and then determines which resources it will share with the user. NetWare 3.*x* works best in networks that require only a single server, because each server maintains its own independent Bindery database (see Figure 12-38). A user accessing resources on three different servers must have three separate user accounts and passwords. NetWare 3.*x*'s reliance on IPX/SPX also limits its use, as more and more networks move to TCP/IP as the protocol of choice.

TIP Although it is possible to add TCP/IP support to a NetWare 3.*x* server, NetWare 3.*x* servers running TCP/IP rarely occur in the wild. For the purposes of the Network+ exam, assume that all NetWare 3.*x* servers use IPX/SPX as their sole networking protocol.

NetWare 4.x and NDS

NetWare 4.*x* built on the success of NetWare 3.*x* by adding two key features: Novell Directory Services (NDS) and TCP/IP encapsulation. The NDS feature organizes all user and resource information in a database referred to as the NDS tree. The NDS tree acts as

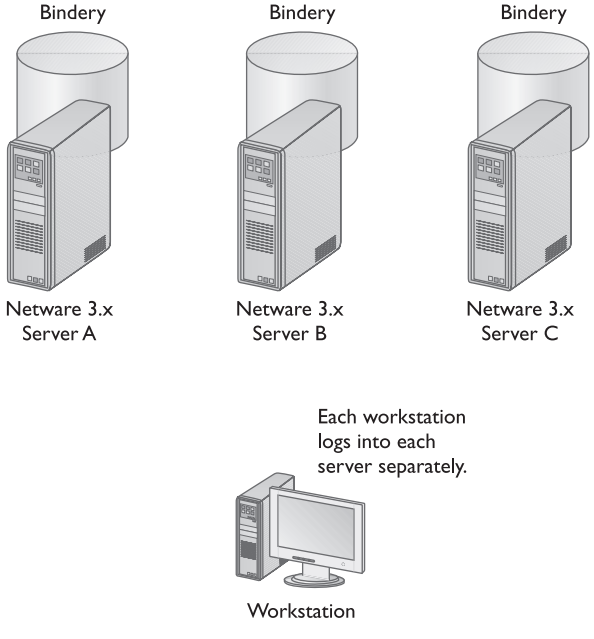


Figure 12-38 NetWare 3.*x* servers maintain separate Bindery databases

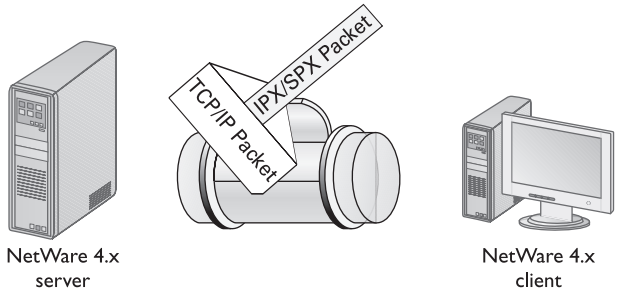


Figure 12-39 Encapsulation enables NetWare 4.*x* to use TCP/IP.

a centralized security database, enabling users who log onto the directory to access all of their resources anywhere on the network. NDS has been around for quite a while and precedes Windows Active Directory by many years. NetWare 4.*x* also supports TCP/IP, enabling NetWare servers and clients to place IPX packets inside of TCP/IP packets, a process known as encapsulation (see Figure 12-39). Although NetWare's basic design assumes the use of IPX/SPX, encapsulation enables NetWare to use TCP/IP without a massive redesign. Unfortunately, encapsulation hurts performance by adding an additional layer of protocol information to each packet.

TIP Both NDS and Active Directory are based on a directory standard called X.500.

NetWare 5.*x*/6.*x*

NetWare 5.*x* and 6.*x* run TCP/IP natively, removing the need for TCP/IP encapsulation. Having native TCP/IP means that NetWare no longer needs to use IPX/SPX at all (although it can for backward compatibility). Because NetWare now speaks TCP/IP natively, it performs far more efficiently than NetWare 4.*x* when using TCP/IP.

For the Network+ exam, familiarize yourself with the protocols and security databases used by each version of NetWare, as shown in Table 12-1.

Novell calls its version of the Windows NT/2000/XP administrator account—an account that provides total and complete access to the system—the supervisor or admin account, depending on the version of NetWare. Make sure only a few administrators have access to the supervisor/admin account!

NetWare Version	Security Database	Protocol(s)
NetWare 3.*x*	Bindery	IPX/SPX
NetWare 4.*x*	NDS	IPX/SPX or TCP/IP
NetWare 5.*x*	NDS	IPX/SPX or TCP/IP
NetWare 6.*x*	NDS	IPX/SPX or TCP/IP

Table 12-1 NetWare Security Databases and Protocols

UNIX and Linux

As the importance of the Internet continues to grow, the *UNIX* operating system, long a mainstay of university and scientific computing, is becoming more important for the average network tech in the trenches. Originally, the Internet consisted of a few UNIX-based systems at a handful of universities spread around the world. The basic Internet protocols, like FTP, HTTP, DNS, and ARP, actually originated in the world of UNIX and were only later ported to other operating systems. UNIX comes in many versions, but they all share certain features. The flexibility of UNIX and the rise of open source variants like Linux and Free BSD make UNIX a network operating system that network techs ignore at their own peril.

Many Flavors

The wide variety of UNIX versions, commonly referred to in geek-land as *flavors*, arose because Bell Labs made UNIX available to universities, and allowed the universities to modify the operating system to meet their own needs. This freedom to adapt the operating system encouraged innovation, leading to the development of critical technologies such as TCP/IP-based networking, but it also resulted in many flavors of UNIX possessing significant differences. Today, major variations include Sun's Solaris, IBM's AIX UNIX, Hewlett-Packard's HP UNIX, and BSD. While all versions of UNIX share a similar look and feel, a program written for one flavor often requires significant revision before it can run on another. Fortunately, the typical network tech can safely leave the variations among UNIX flavors to the programmers. From the network tech's point of view, all versions of UNIX are more alike than different.

TIP The Network+ exam does not cover the differences between versions of UNIX/Linux.

Web Applications

Although it faces increasing competition from the Windows NT and NetWare families, UNIX remains the server of choice for providing Internet-based services such as web browsing and e-mail. The protocols used for Internet-based services mostly originated in UNIX versions, and many organizations that use NetWare or Windows for their file and print sharing needs still rely on UNIX for their Internet services.

Printing

For many years, the UNIX/Linux people used the protocol set LPR/LPD to handle printing chores. Clients used the line printer request (LPR) portion to submit a print job to a print server. The server ran the line printer daemon (LPD) protocol to handle those submissions.

The LPR/LPD printing system is quickly being replaced by the *Common UNIX Printing System (CUPS)*. CUPS addresses a number of limitations inherent to LPR/LPD and has made printing in UNIX and Linux far easier and flexible than in the past. The CUPS printing system is based on the Internet Printing Protocol (IPP) standard and includes

substantial improvements over LPR/LPD. CUPS supports any printer language, although its most commonly associated with the PostScript language. In fact, CUPS printer definition files all end with the PPD (Postscript Printer Definition) extension, even the ones that are used with non-PostScript printers. CUPS has built-in web-based support for printer connections and printer management and supports SAMBA and LPD printers. CUPS also supports most TCP/IP features such as encryption and proxies, and other features that LPD never knew how to handle.

The CUPS server program on UNIX /Linux systems is called—surprise—CUPS. On the client side you either run the CUPS service or use one of many different programs to access the CUPS server. On most UNIX/Linux distributions, CUPS is now hidden from users by some from of graphical printer configuration dialog box. For those die-hard command-line users who want to run CUPS from a command prompt, you use the **lp** or **lpr** command to send your CUPS print jobs to your CUPS server.

Open Source and Linux

If you haven't heard of Linux yet, you need to read the newspaper a little more often! Linus Torvalds, while a student, expressed his frustration over the high cost of most versions of UNIX by building his own. What makes this story special is that Torvalds licensed his UNIX clone, dubbed Linux, in a unique way. *Linux* is an *open source operating system*, distributed under the terms of the GNU General Public License (GPL), which means (among other things) that anyone who purchases a copy receives full access to its source code, the building blocks of the operating system. Free access to the source code gives software developers the power to modify the operating system to meet their needs. This has led to the rapid development of a wide variety of applications, including some of the most commonly used web and e-mail servers on the Internet. In most cases, both the Linux operating system and Linux applications are available for free download from the Internet, although vendors like SuSE and Caldera sell boxed versions, and charge for support services. For all intents and purposes, Linux is a full-featured clone of UNIX.

Does UNIX have a super account like Windows and NetWare? You bet it does! The all-powerful account in all versions of UNIX/Linux is called root. Again, giving someone the password to the root account gives them the ability to log onto a UNIX/Linux system with complete access to anything they want to do on that system. So give out the root password sparingly!

Mac OS

Apple Computer was one of the earliest adopters of network functions for its systems. In keeping with Apple's long-term attitude of "we can do it better," Apple implemented networking very differently from the other network operating systems. Adding to the confusion, over the years Apple has made a number of upgrades to the networking functions of the *Macintosh operating system*. All of these incremental changes make it difficult to give a brief overview of Macintosh networking without going way, way outside the scope of the Network+ exams. Instead, I'm going to concentrate on current Macintosh NOS functions, with a small nod to a few critical historical points that CompTIA wants you to know.

The key to the uniqueness of Macintosh networking in the early days is *AppleTalk*, Apple's do-it-all family of networking protocols. AppleTalk handles tasks ranging from Transport Layer packet creation to establishing sessions between systems to support for network applications. One can reasonably compare the functionality of AppleTalk with the territory covered by Microsoft's NetBIOS and NetBEUI (although any good Mac networking tech will probably cringe when I say that). Like NetBIOS, AppleTalk was designed primarily for file and printer sharing. Its naming conventions are very similar to what you see in NetBIOS, although AppleTalk supports very long system names. The practical limit for an AppleTalk name, however, is about 20 characters. Like NetBEUI, AppleTalk does not support routing and instead uses a NetBIOS-like broadcast function to enable systems to recognize each other. Macintosh systems also use a grouping function called *zones*. A zone works for the most part just like a Microsoft Workgroup. Zones do not provide any real network security and simply act as a tool for organization.

Not surprisingly, given its overwhelming popularity, all modern Macintosh systems implement TCP/IP, using a program called AppleShare IP. AppleShare IP's mission is to connect your Macintosh system to IP networks, including the Internet. Apple also makes Mac OS X Server, a full-blown server OS that includes tools to facilitate interconnectivity with Windows and Linux. It also greatly enhances network security, in particular by implementing groups and robust user accounts. So, networking in Macintosh involves two products: the basic networking functions of AppleShare IP, which are built into all Macintosh systems, and Mac Server.

Creating Servers and Clients

After choosing the best NOS for your network, you must install the operating system software on your networked systems. You will confront a number of critical issues at various steps during the installation process. While each operating system handles these issues differently, either you or the NOS must take care of each of them.

Network Interface

Every system on the network must have some device by which to access the network, called the *network interface*. In most cases, this will be a NIC or a modem. Fortunately, the world of Plug and Play (PnP) now predominates, and installing a NIC or modem has pretty much been reduced to plugging in the device and then kicking back while the NOS handles everything else for you. The only issue you need concern yourself with is making sure the device installed properly. That means knowing where in the operating system you go to check on this. In almost all versions of Windows, it means a trip to the good old Device Manager. Figure 12-40 shows a perfectly functioning NIC in the Windows 2000 Device Manager.

Other operating systems like Linux aren't nearly as pretty, but they're just as functional. Figure 12-41 shows someone running the **IFCONFIG** command and looking for eth0—the universal name for an Ethernet adapter in the UNIX/Linux world.

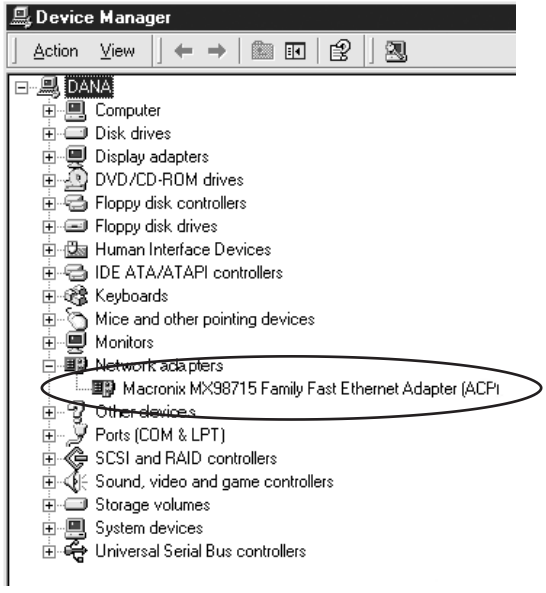


Figure 12-40 A functioning NIC in the Windows 2000 Device Manager

If a NIC isn't working correctly, you'll get some type of error information. Windows adds a pretty **X** or **?** to the graphic of the device, while Linux just gives you some text, but either way you can tell whether the NIC is working. Keep in mind that every operating system invariably provides more than one way to check the NIC. The two examples I gave aren't the only ways to check a NIC in either Windows or Linux—they're just the ones I use. Refer to Chapter 8 for more information on installing NICs.

Protocol

Every networked PC must run a software suite that enables the PC to communicate over the network, in other words, a *protocol*. Since everyone and their dog uses TCP/IP, you can bet that your NOS will invariably install TCP/IP as the default protocol, unless you're running something a tad older. This is a big deal in the Network+ exam's eyes—make sure you know the different protocols that install with the different network operating systems, including some of the older ones! I describe them all in this chapter.

Figure 12-41 Running IFCONFIG eth0 in Linux

```
[root@localhost /dev]# ifconfig eth0
eth0   Link encap:Ethernet HWaddr 00:40:F4:23:0C:51
       inet addr:192.168.4.19 Bcast:192.168.4.255 Mask:255.255.255.0
       UP BROADCAST RUNNING MTU:1500 Metric:1
       RX packets:14 errors:0 dropped:0 overruns:0 frame:0
       TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       Interrupt:10 Base address:0x1000

[root@localhost /dev]#
```

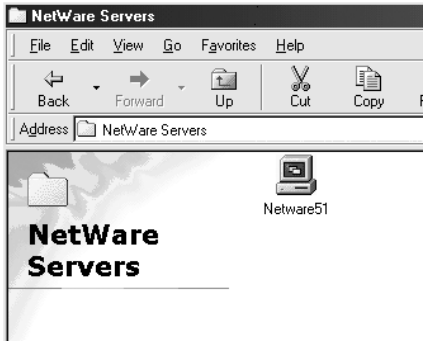


Figure 12-42 The Novell NetWare server displayed in My Network Places

Naming

Okay, this is a big one! In most networks you need to give every system, or at least every system that shares resources, a "friendly name," which is to say, one that isn't 192.168.43.2 or something else cryptic and hard to remember. You see this friendly name when you view shared network resources. In Windows, you can view shared resources using the ever-popular Network Neighborhood/My Network Places. (Novell NetWare also calls its application Network Neighborhood.) All network operating systems have some similar application. Figure 12-42 shows a Windows 2000 system's My Network Places displaying a Novell NetWare Server, called NETWARE51.

As I've said, organization-based networks name groups of computers for organizational purposes. Luckily, creating these group names is the realm of the folks who set up servers, an area Network+ doesn't expect you to know. However, if a network administrator tells you to set up a Windows 2000 client system, Network+ does expect you to understand organizational groups and group names, and it expects you to know how to assign a client to a group. Figure 12-43 shows five groups on my network. These groups are called domains because, well, that's what Microsoft decided to call its groups of computers. The domain names in this example are ones I created.

Different organization-based network operating systems use different names for these groups. Windows calls them domains. NetWare, meanwhile, calls them trees.

Server or Client

If you install Novell NetWare, it will automatically set up as a server. Other operating systems, such as Windows 98, automatically perform as clients. Windows 2000/XP/2003, Linux, and Macintosh computers run both as clients and servers. The trick is that some

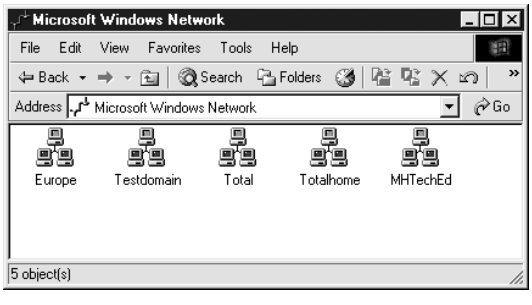


Figure 12-43 Domains in Mike's system

operating systems, particularly Windows 9*x* clients, when running in resource-based networks, require that you also set them up to act as servers. All this really means is you must turn on File and Print Sharing, so the Windows 9*x* systems can share.

Super User Accounts

All network operating systems require user accounts. Any NOS that requires this will come with a built-in, all-powerful user account that has total control of everything in the network. This account is called supervisor or admin in NetWare, Administrator in Windows NT/2000, and root in UNIX/Linux. When a system is first installed, you must set the password for this account. As you might imagine, this account's password is something you want only the most trusted people to know! Most operating systems require you to use this account, or an equivalent (you can usually make more accounts with the same power), to do most of the network administrative tasks. Most of the time the network admin herself will create user accounts; however, most network operating systems enable an administrator to delegate her power to create accounts to other user accounts—a nice way to delegate administrative work!

TIP As a security consideration, most network techs change the default name of the built-in administrator account. Why? Well, any potential hacker knows that there's an account called, for example, "Administrator" on a Windows server system. A valid user account name is half of the hack—now all they need to guess is the password! If you want to harden your server's security, you should also make them have to guess a valid user name. Some sneakier network techs even create a dummy account called "Administrator" that has no real power, except to act as an obstacle for hackers.

Groups

Most network operating systems have default groups. For example, Windows NT/2000 has a group called All Users. Anyone with a valid user account automatically becomes a part of this group. Your network admin will almost certainly have made such groups for your network.

Passwords

Passwords are now pretty common to all network operating systems, and the folks who give the Network+ exam want you to have a good general understanding of passwords.

Network security only works well when users keep their passwords secure. Passwords, for example, should never be written down where another user can find them, and users should never reveal their passwords to anyone, even the network administrator! In most cases, the administrator can reset a user's password without knowing the old one. Many users, however, remain unaware of this possibility and so fall prey to one of the oldest hacker tricks in the book: the fake tech support phone call. In a large organization, most users will not know every network support technician. A hacker can call up one of these hapless users and say, "This is Howie from tech support. We're upgrading the forward deflector array and we need your password so we can reset it when we're done." A shocking

number of users will simply give out their password when asked over the phone. Getting humans to think is never easy, but it's a vital part of network security!

Educating network users about the proper care and feeding of their passwords is a critical part of any network security plan. First, teach users to pick good passwords. A good password cannot be guessed easily. They should never be based on any information about the user that a bad guy can obtain easily. For example, if Herman lives at 1313 Mockingbird Lane, is married to Lily, and has a pet named Spot, he should never use the following passwords:

- mockingbird
- dribgnikcom (mockingbird spelled backwards)
- lily
- ylil (lily spelled backwards)
- spot
- tops (spot spelled backwards)

Ideally, a password should not be a real word at all. Hackers probing a network often run password-guessing utilities that try common dictionary words at random. Network administrators can reduce the effectiveness of such password-guessing programs by requiring that all passwords be longer than six to eight characters. Hackers have a more difficult task guessing longer passwords because there are so many more possible combinations. The most secure passwords contain a combination of letters and numbers. Users hate them because they are also the hardest to remember. The following list contains strong passwords:

- gr78brk3
- tnk23wqk
- bob0tw2&

A good network administrator should assume that, given enough time, some users' passwords will become public knowledge. To limit the impact of these exposed passwords, a careful network administrator sets passwords to expire periodically, usually every 30 days at the most. If a password becomes public knowledge, the gap in network security will automatically close when the user changes his password. One of the most frustrating aspects of implementing passwords is the stream of support calls from users who can't log onto the network. If I only had a dollar for every time a user left on the CAPS LOCK key, or just didn't type in the password correctly!

TIP A strong password should be at least eight characters, contain both letters and numbers, be changed on a regular schedule, and not be based on easily guessed information.

Most network operating systems also enable you to disable a user's account. A disabled user account is simply an account whose access has been disabled but that hasn't been removed from the system. Many network administrators will disable an account while a user is on extended leave or on temporary assignment. Of course, somebody is sure to hear about it when the user comes back and can't log on because their account is disabled! While the Network+ exam isn't interested in whether you know how to enable and disable user accounts for your particular NOS, you do need to know that they can be disabled.

As a person supporting networks, you must have a basic understanding of the different makes and models of network operating systems available today. Become familiar with the many variations of Novell, Microsoft, and Linux/UNIX products, and be sure you can explain the differences between client/server and peer-to-peer networking.

Chapter Review

Questions

1. Your network consists of a Novell NetWare 3 server and a UNIX system. Sally cannot access the server, so you go to My Network Places and discover that Sally has only the NetBEUI protocol installed. Which of the following protocols should you install to enable Sally to communicate with the NetWare server and the other systems using UNIX? (Select all that apply.)

   A. Banyan VINES

   B. TCP/IP

   C. IPX/SPX

   D. NetBEUI

2. Of the following NOS server programs, which one can only be a server and never a client?

   A. Novell NetWare

   B. Microsoft Windows 2000

   C. Microsoft Windows 98

   D. UNIX

3. What type of system accesses a resource?

   A. Mac

   B. Server

   C. Client

   D. Terminal

4. Novell NetWare, Windows NT/2000, and UNIX/Linux all have a built-in, all-powerful user account that has total control of anything on the network. Each NOS uses a different name for this all-powerful user. UNIX/Linux calls it a(n) _ _____; Windows NT/2000 calls it a(n) _____; and NetWare calls it either _____ or _____.

   A. Root, Administrator, Admin, Supervisor

   B. Supervisor, Root, Administrator, Admin

   C. Admin, Supervisor, Root, Administrator

   D. Administrator, Admin, Supervisor, Root

5. Your network is made up of ten Windows 98 systems, and you installed the TCP/IP protocol on all the systems. Now Melissa wants to share her hard drive. She goes to My Computer and alternate-clicks the C: drive, but Sharing is not listed as one of her choices. You know that her cable and NIC are working, and she can see everyone on the network. What could be the problem?

   A. IPX/SPX needs to be installed on Melissa's system.

   B. File and Print Sharing has not been installed on Melissa's system.

   C. Client for Microsoft Networks has not been installed on Melissa's system.

   D. Melissa is not running the sharing protocol.

6. May wants to allow Mary Jane and Peter to view and modify a database stored on her server. She wants Betty to be able to view the database but not modify it, and she wants Jonah to have no access to the database whatsoever. Each user should have his or her own password. What kind of security should May implement?

   A. High-level

   B. Share-level

   C. User-level

   D. SMTP-level

7. NetWare 3.*x* servers store user account and password information in a database called the:

   A. Domain

   B. NDS tree

   C. Bindery

   D. Registry

8. You are running a Linux system on your network. In order for you to access root on this system, you need to know the:

   A. Location of the directory

   B. Computer's name

C. Password

D. Root code

9. You are running a Windows 2000 server on your network. You need to make sure that the TCP/IP protocol suite, IPX/SPX protocol suite, and NetBEUI protocol suites are installed. Which of these protocols are found natively on Windows 2000? (Select all that apply.)

   A. NetBIOS

   B. NetBEUI

   C. TCP/IP

   D. IPX/SPX

10. When using a common security database, Novell NetWare servers must be organized into a(n):

    A. NDS tree

    B. Domain

    C. Ring

    D. Web

Answers

1. **B, C.** You need to make sure that Sally's system has the IPX/SPX protocol for the NetWare server and the TCP/IP protocol for the UNIX systems.
2. **A.** With the exception of Novell NetWare, every operating system capable of networking (Windows, UNIX/Linux, and Macintosh) allows systems to act as both servers and clients at the same time. Novell NetWare cannot act as both a server and a client on the same system.
3. **C.** A client is a system that accesses the shared resource.
4. **A.** UNIX/Linux calls its all-powerful user account Root; Windows NT/2000 calls it Administrator; and NetWare calls it either Admin or Supervisor.
5. **B.** Melissa must turn on File and Print Sharing before her Windows 98 system can function as a server and share her hard drive.
6. **C.** May should implement user-level security, which lets her assign different rights and permissions to each user, and give each user a unique password. Share-level security assigns a password to each resource, but would not fulfill May's needs because Mary Jane and Peter would use the same password to access the database. Simple Mail Transfer Protocol (SMTP) is an e-mail protocol that has nothing to do with securing files on a server. High-level security is a bogus term.

7. **C.** Each NetWare 3.*x* server has its own security database called the Bindery. NetWare 4.*x* and 5.*x* servers share a common NDS database, and Windows NT servers share a domain database. The Registry is a central hierarchical database used in Windows 95, 98, NT, and 2000 to store information necessary to configure the system for one or more users, applications, and hardware devices.
8. **C.** You need to know the password to the root account to log onto a UNIX/Linux system with complete access to that system.
9. **C, D.** TCP/IP and IPX/SPX are native on a Windows 2000 system. NetBEUI comes with 2000, but you must install it.
10. **A.** Novell NetWare servers use Novell Directory Services when sharing a common security database. Servers sharing that sdatabase exist within an NDS tree.