Chapter 1

Server Types and Roles
For the most part, each chapter in this book closely maps to the Server+ Exam Blueprint from CompTIA. We have listed the exam objectives covered by each chapter at the start of the chapter, and fully covered each objective to the best of our ability in that one chapter. In a couple of cases objectives cross chapter lines, but this is infrequent, and is done to arrange the information in the most readable and useful format possible.

Chapter 1, though, sets the stage for our later studies, and as such does not specifically map to any objective. For many of you, this first chapter may be review. That’s excellent, and the authors won’t mind at all if you move quickly through this information—or any topic where your current level of experience makes you extremely confident. This chapter will help get you get started on achieving your goal of the Server+ Certification.

What Is a Server?

When preparing for the Server+ exam, one question needs to be gotten out of the way immediately: What exactly are servers, and what makes them special enough to deserve an entire exam dedicated to them? The answer to this requires that the term server itself be defined. Put simply, there are two key definitions of server in the Information Technology world:

**server (sûrvr), n.** 1. Computer software designed to assist other computers on a network by performing tasks for them or providing information to them. 2. Computer hardware optimized for the task of running server software.

Each of these definitions needs to be considered separately, along with its implications for what a “server” is. We’ll take some time in the following sections to dissect these definitions, taking care to examine servers as software as well as servers that operate solely as hardware. We will cover scalability versus expandability, the relationship between security and
dependability, and finally stability and redundancy. Before we move into discussing the various roles of a server, we will overview server add-ons.

**Server as Software**

Let’s start by examining the first definition. For any computer to function, it needs an *operating system (OS)*. This is the code that tells the computer how to function. You know that, of course. You have also probably encountered the term *NOS (network operating system)*, which is used to describe a server OS. Things become a bit tricky, though, when we start trying to distinguish an OS from a NOS.

The reason for this is that by the definitions we’ve just shown you, any OS that can perform services or share files on the network is a server. Many of you have used the file sharing capabilities of Windows 98, for instance. All of Microsoft’s modern OSs have the ability to share out files, and even to maintain NetBIOS browsing lists that allow computers to find each other on the network. Even so, we don’t generally think of Windows 98 as a “server OS,” and neither does the Server+ exam. Rather, the NOS term is reserved for products such as Novell NetWare, Microsoft Windows 2000 Server, or Sun Solaris.

In order to decide which software you will need as your NOS software, you will need to examine and consider the following characteristics:

- Scalability
- Security
- Stability
- Client prioritization

Reviewing each of these characteristics in full is a good starting place when considering server hardware for your NOS. As such, we will start by examining the concept of scalability and how it relates to server performance.

**Scalability**

Most computers serve only a single master, in that the user working locally on the machine is the only one giving orders. The user may run one application, or a number of them, but the amount of computer power a single user needs is relatively limited, especially as we enter the world of multi-gigahertz processing on the desktop. Because only one user is expected to be using the OS at a time, a normal OS is intended for use on machines with limited resources. Windows 98, for instance, cannot recognize or use more than one processor.
Windows 2000 Advanced Server, on the other hand, supports 8 processors, and Sun’s Solaris supports up to 128 processors on a single system.

Besides just allowing for more hardware, network operating systems are designed to allow for features like clustering and load balancing:

**Clustering**  Used to allow a number of servers to share resources, *clustering* essentially creates a single “virtual server” out of a number of machines. The computers share an IP address and generally use the same data array, as shown in Figure 1.1.

**FIGURE 1.1**  Server clustering

![Server clustering diagram](image)

**Load Balancing**  Similar to clustering, in that two or more servers team up to do a single job. The thing that distinguishes load balancing, though, is that each server retains its own identity and often keeps its own copy of needed resources, as shown in Figure 1.2.

**FIGURE 1.2**  Server load balancing

![Server load balancing diagram](image)
Scalability, then, is the ability of a computer system to support large numbers of users, and run extremely demanding software applications. That is just one job for a NOS, though. Next comes the ability to keep things safe.

**Security**

Network operating systems also are generally far more secure (or at least securable) than client operating systems. This enhanced security can take the form of a username/password database, access restrictions on files or services, or any number of system security policies.

One of the odd things about the Server+ Exam is that, because most of the questions follow a generic format, and because very little security information falls into this “generic” category, you will find few system security–related elements on the exam. This is strange, of course, because network security is among the primary job functions of a server administrator!

The physical security of the server, however, is a major concern of the exam. Locking down the server room will be dealt with in Chapter 13, “Managing and Securing the Server Environment.” Some general security topics will also be considered on a NOS-by-NOS basis in Chapter 7, “Network Operating Systems.”

**Stability**

While most desktop PCs are shut down each night, and are used only a few hours each day, servers are generally on 24/7, and as such they need an OS that is extremely stable. Moreover, as tens or hundreds of people are interacting with the server each day, it is critical that the OS be resilient and able to deal with this constant onslaught of requests without locking up or giving up.

To help guard the health of these machines, NOS software is often pickier about what software it allows to run, and which applications and drivers it will allow you to install. While this helps to insulate the server from problems caused by bad software, it also means that NOS applications often are specifically written for the OS, and can be extremely expensive.

**Client Prioritization**

One last characteristic of a server OS is that it gives priority to client connections when allocating resources. The primary purpose of a NOS is to take care of clients, and as such a user at the server console is treated as just another user, or sometimes even given a lower priority than network users.
Some examples of server-class operating systems are listed below, along with website information so you can go to learn more about each of them.

- NetWare: www.novell.com/products/netware/
- OS/2: www-4.ibm.com/software/os/warp/
- Solaris: www.sun.com/software/solaris/

Solaris is a Unix-based system, and Unix operating systems are all based on a server-class platform. Linux is also Unix based. Still, both Linux and Solaris are often used as a desktop OS, and are as flexible as Windows in that they can be used for pretty much any role in the enterprise. Start at www.linux.org/dist/ to learn more about the emerging Linux challenge to the established NOS/OS leaders.

Servers as Hardware

The second definition of a server is one that involves specialized hardware designed to handle the extreme demands of NOS software and network users. Companies such as IBM and Compaq produce computers specifically for these needs, and sell them in separate product lines. Compaq, for instance, has its extremely popular Proliant series, and Dell sells the PowerEdge line. At a very general level, servers are essentially just enhanced PCs. Many managers look at the price of a new Compaq Proliant 1GHz server and say, “Why are we paying $10,000 for this computer, when we could get a Compaq PC that is just as fast for $1,000 at Circuit City?”

This is a valid question, because Windows 2000 Advanced Server or Sun’s Solaris can be installed on a desktop-class PC without any trouble. If you are in the position of proposing a server purchase to a manager or client, you should be prepared to explain the reasons behind the higher cost of specialized server hardware.

It is worth emphasizing here that server-class software can be installed on a desktop PC and that Windows 98 can be installed on an IBM server. To get best performance, though, both the hardware and software need to be server-class.
To help you with that explanation, let’s take a closer look at the benefits a server provides for that extra money:

**Expandability**

One of the most important characteristics of server-class hardware is that it is generally built with generous expansion capability. Most servers allow for far more RAM (often over 4GB), more drive space (most servers have 5–10 drive bays) and more processors—it is hard to get a desktop PC that fits 8 processors because the cases for normal PCs simply do not have room for that much hardware. Along with all of this additional hardware comes the need for additional fans and a larger power supply as well, which also take up room.

**Dependability**

Server hardware needs to be reliable. Unlike desktop PCs, which are generally shut down each evening, servers often are expected to run constantly for weeks or months. The length of time a server has been running, or sometimes the percentage of time it has been running, is referred to as its **uptime**. Some servers prominently display the amount of time they have been up on their console, while others (Windows, anyone?) tend to hide that information!

Any time a server is not running, the dreaded word **downtime** is used to describe the amount of time that it is off. Because servers are critical to modern networks, and networks are critical to modern organizations, a server down situation rarely goes unnoticed. E-mail doesn’t work, or users can’t get to files, or “the Internet doesn’t work,” and calls start flooding into the help desk.

Along with backup and security, the prevention of downtime is probably one of the most important jobs of an administrator. Server-class hardware helps to maximize uptime through higher quality hardware and the ability to duplicate critical hardware for redundancy.

**Quality**

One of the reasons servers cost more than desktop PCs is that the pieces used to build the server are better. No one argues about why a Porsche costs more than a Yugo, but somehow a lot of people who drive very nice cars find it difficult to understand why they should pay for quality in their server room as well.

Server components are manufactured to higher standards, both in terms of the materials used and the precision of the craftsmanship. Moreover, these components are tested to ensure that they work well together. This is done
to ensure that a server will remain operating and reliable regardless of the amount of work required of it. Much of this information is very different from the way those same resources are discussed in the A+ book, which deals with the maintenance of “normal” desktop PCs.

**Redundancy**

Quality components are great, but even the best machines sometimes fail, and computers are no exception. In order to try to prevent hardware problems from resulting in immediate downtime, though, most server-class computers support redundant hardware for key components. This practice is known as *redundancy*.

Redundant components can include power supplies, for instance. If a server has two or more power supplies, both of them can work together to power the system. However, if one of them fails (or is unplugged), the other is able to take on an increased load and power the entire system. Other examples of commonly duplicated hardware include hard drives, drive controllers, and network cards.

---

**WARNING**

Two items that are not redundant are processors and RAM modules. Even if you have four processors in a machine, if one processor fails, the server will go down. The same with RAM. Remember that expandability and redundancy are different things!

**Server-Only Features**

Besides just supporting more and better hardware, and offering helpful services not available on regular operating systems, modern servers also can be equipped with a dizzying array of add-on equipment. Although some of these components are making their way into the desktop computer environment, they are normally associated with server environments. These include RAID controllers (standard on most servers), SCSI controllers, an uninterruptible power supply (UPS), external drive arrays, fax or modem bank hardware, and tape backup drives. Any of these can be installed into desktop-class machines as well, but generally their expense and resource requirements dictate that they be used in server-class machines with server-class OSs.

You may already be in charge of a network, and have your own server(s) to refer to as you read this book, but if not it might be useful to get an idea of what these machines are like. Before reading too much further, you may want to take a look at some of the beasts that the Server+ certification
prepares you to deal with. Below is a sampling of some large computer hardware manufacturers, and their server lines:

- Compaq: www.compaq.com/products/servers/
- Gateway: www.gateway.com/work/products/sb_srv_catalog.shtml
- Hewlett-Packard: www.hp.com/products1/servers/
- IBM: www-1.ibm.com/servers/

While this book is being written, HP and Compaq are in the process of merging, so things may be changing a bit there. For now, though, they have separate product lines.

Server Roles

Servers must perform a dizzying variety of tasks on the network. On smaller networks a single machine might perform many or all of these tasks, and that is perfectly workable because servers are designed to be good at doing multiple tasks simultaneously. On larger networks, though, specialization allows each machine to be tailored specifically for the tasks it is assigned.

This section will sample a few of these tasks for you and give you an overview of what servers do on a network and how each of these tasks takes its toll on server resources. Three general types of server roles will be detailed in this section: security, network, and user. These are loosely grouped, and some server roles cross over between the categories, so concentrate more on what they do than where they are grouped. As you read through this section, keep two questions in mind:

1. What roles does this server play on the network, and how does its performance impact network users?
2. What type of operating system and hardware should be used to improve the efficiency of the machine running this task?
Security Roles

Networks have evolved into the storage location for almost all documents and data in most large and midsize companies. The protection of this data—both from destruction and from unauthorized access—falls to the server administrator.

Network security is generally provided by the operating system, and as we look through each of the following sections, you will be given pointers to websites that detail how these services are implemented into various server systems. Because the Server+ exam itself is vendor-neutral, you don’t have to spend a lot of time studying this stuff, but if you are interested in learning more about a topic, some of these URLs could come in handy as a starting point.

Account Management

Most of the following services depend on the ability of the server to determine one fact—who it is that is trying to use the service. This is generally accomplished by using one or more servers on the network to store and authenticate user credentials.

Account management servers generally keep track of (among other things) two basic pieces of information—the user and the password.

User (or Username)

The user is the basic building block of network security. The user defines an individual (for example, “John Doe” becomes “JDoe”) or a network role (“administrator” or “root”), and comes in one of three basic types:

- **Administrative** At least one top-level account must be created during the server installation process. That account is given authority to manage the server. Sometimes called the “god account,” this first account is known by many names by the various computer tribes. Microsoft calls it the Administrator, Novell calls it Admin, and Unix/Linux calls it Root.

- **User Created** The administrator can then create individual system identities in order to differentiate between the various individuals on your network, all of whom have their own network resource needs and security clearances. Creating a user account registers the user as a security object on the server.

- **Guest/Anonymous** Sometimes access to the server needs to be granted to users who are not registered in the account database. This can be done by creating a default user. Anonymous users are most commonly seen on
Internet web servers, where thousands of users might visit a server to view web content. The users do not have individual accounts, and they all log on through a single generic account. FTP servers also often allow file retrieval access to anonymous users.

**Password**

It is almost inconceivable that anyone reading this book needs to have passwords explained. If you do, all I can say is you have a bit of work ahead of you! If a username says “this is who I am,” a password says “and here’s proof I am who I say I am.” Password security is a critical part of server security, because if your account passwords—especially administrative passwords—are discovered, the server and network security are compromised.

Many of the services listed in the authentication section that follows are specifically designed to protect system passwords. Sadly, even the best security software cannot defend passwords scribbled on sticky notes and posted on a user’s monitor. An effective password strategy needs to include a password scheme that balances security and usability.

Windows server administrators can use a nifty tool called L0phtCrack (recently renamed to LC3) to test their system’s defenses. LC3 and other and eye-opening toys for various OSs are available from @stake at www.atstake.com/research/.

### Real World Scenario

**Designing a Password Strategy**

You are in charge of a committee studying the current password structure on your company’s network. The purpose of the committee is to examine the corporate security needs and see whether the existing password structure is sufficient.

To your dismay, the committee reports back that not only are password standards not uniform throughout the company, but that many offices do not have any password policy at all. In attempting to craft a password strategy, you find the following:

- All users have Internet access.
- Many users are logging on and working from home.
This example is representative of the types of questions you might get. All of the problems and solutions are generic because getting specific would require mentioning particular authentication methods or require you to perform tasks on a particular directory structure. This would require dealing with a certain vendor, and in most cases the exam is more interested in giving you logic puzzles than it is testing your knowledge of a particular product.

**Authentication**

The process of submitting a username/password set and having it tested against credentials stored in a server database is called *authentication*. There are a number of methods of authentication available to a server. Here is a sampling, arranged roughly in order of least secure to most secure:

- **Plain Text** This is the simplest form of authentication. In plain text authentication, username and password information is simply sent out over the network in *clear text*—standard ASCII code that can be intercepted and read easily. Plain text authentication is highly frowned upon in secure environments. Scratch that...plain text is highly frowned upon for any environment.
Encryption  The process of protecting authentication information generally involves encrypting the username and password information as it is transferred between the client and server. Encryption takes many forms, but Solaris, for instance, supports Kerberos, Diffie-Hellman, and others. Go to www.sun.com/software/white-papers/wp-security/ for more on this topic.

Smart Cards  Security can also have a physical component, and smart card logon options are becoming more common. These depend on the user knowing a piece of information (password) and having physical proof of identity (generally a swipe card).

Biometric Identification  The most complex and futuristic authentication method—biometrics—uses retinal scans, voice scans, fingerprint analysis, and other tools to provide physical proof of identity. The key is using a human feature that is unique to each individual. Biometric ID integrates physical identification with authentication.

Real World Scenario

The Benefits of Security

Many people take security for granted. After all, what's the big deal? "There is nothing on my computer worth stealing." These are the famous last words of many who have been caught up in an incident where their computer has been broken into.

I recently experienced a situation that drives this point home. A private school that I worked at long ago called me for some advice with regard to their new network setup. The school had purchased a dedicated Internet connection that was being shared to an entire lab of computers. Unfortunately the server was not using a network operating system. The faculty elected instead to use a desktop operating system. Desktop operating systems do not provide the necessary level of security or authentication needed for a server role.

In advising them on their network questions, I also cautioned them on the security holes that their Internet connection and operating systems were experiencing. Unfortunately it was only a few weeks later that I received another call from the school: This time, they were frantic. They had received a nasty letter from their Internet Service Provider, who was accusing them of illegal activity. The faculty was completely oblivious to what was going on.
Directory Services Server

Directory services servers allow the processes of account management and authentication to be handled centrally by a single machine or group of machines. This allows a single username and password set to be used throughout an entire network. Microsoft made the idea of “one user, one account” into a mantra when they were promoting their Windows NT domain structure, and the key premise is sound. Wherever a user goes in the enterprise, that user should be able to authenticate to the network using the same username and password. Figure 1.3 shows how a user can log on in two separate areas of a Windows 2000 domain using the Windows 2000 Active Directory.

**FIGURE 1.3** User logon to the Windows 2000 Active Directory

Both the Active Directory Server and the SQL Server run copies of the Active Directory and share the user database as well as the responsibility for authentication. Common directory services servers include the following:

- Microsoft’s Active Directory
- Novell’s NetWare Directory Service (NDS)
- Sun’s Solstice
The three services listed here are all based on the standards set by the International Telecommunication Union (www.itu.int). The ITU's x.500 directory specification defines how accounts are created and managed. Because of this, all three of these systems use similar structures and logic, making it far simpler to manage multiple systems than ever before.

Security servers do not need to be tremendously powerful in terms of hardware, but they must be powerful enough to respond quickly to client requests. If the directory server is unable to keep up with authentication requests, the network simply slows down to its speed. Nobody gets anywhere without permissions, and these servers are the gatekeepers.

If we were to pick just one resource to emphasize, it would probably be network throughput, which is critical to a security server. Being able to receive and send requests quickly can be facilitated by multiple network cards, or even by locating the directory server in a central part of the network. Also, encryption technologies can be heavily processor intensive, and so servers with enhanced authentication schemes may require additional processor power.

**Networking Roles**

Security work is the glamour job in the server world, and Active Directory and Kerberos (a network security system, developed at MIT, which verifies that a user is legitimate at login) seem to get all the attention in the trade rags. Still, in order to make a network function smoothly, a number of other services also need to be working in the background. These services assist the network in locating servers, identifying computers, connecting remote clients, or moving packets from one part of the network to another.

You won’t be asked to know about the specifics of any one of these technologies, since each of them is implemented a bit differently by different server platforms, but you should be familiar with what they do, and the basics of how they work.
Routing Services

One of the features that a server can offer to the network is to act as a router. Routing and bridging services allow a server with multiple network interfaces to link machines on either side. When acting as a router, the system must build a routing table that shows which machines are available on which interfaces.

As TCP/IP is by far the most common protocol you will need to deal with (some NetWare environments still use IPX/SPX), the IP routing protocols are the most important ones to keep in mind:

**RIP** The *Routing Information Protocol (RIP)* is a distance-vector protocol that enables computers to exchange routing information by means of periodic routing table updates. RIP updates are sent to neighboring networks and RIP information from other routers is returned. The path with the fewest hops (each router involved in a path is one hop) is used when sending data.

**OSPF** *Open Shortest Path First Protocol*, or OSPF, is an open protocol, meaning that it is a standard and that it is available for use in the public domain. OSPF is a *link-state* routing protocol. *Link-state advertisements (LSAs)* are sent to all other routers to allow them to update their routing tables. These LSAs include changes to the routing table, but the actual routing table itself is not sent, unlike RIP. OSPF is far more efficient than RIP on large networks.

Although server machines can be used as routers, in most cases it is better to purchase a dedicated router for this job. Cisco (www.cisco.com) and 3Com (www.3com.com) are two major router manufacturers.

Firewall Server

A firewall is essentially a router turned bouncer. Firewalls are placed at the edge of your network and are used to turn away communications from unwanted or distrusted clients. Nearly every large corporate or organizational network now has at least one high-speed connection to the Internet. While this makes it extremely easy for network clients to access the Web and e-mail, the process also works in reverse, and networks are vulnerable to attack from the Web. As such, nearly all firewalls are concerned with the...
need to protect a local area network (LAN) from the perils of the Internet. Figure 1.4 shows a common network configuration and introduces a concept we need to define.

**FIGURE 1.4** A common firewall configuration

![Firewall Configuration Diagram](image)

Note the DMZ, or *demilitarized zone*. The DMZ is the buffer zone between the Internet and your internal network. It is where any servers that need to be exposed to the Web should be housed. In this case, a web server is sitting in the DMZ, and a server running firewall software protects the intranet. Any requests sent to the web server—including malicious DoS (denial of service) attacks or Internet worms—will be able to reach that server unhindered. The same requests or attacks directed toward the internal network, though, will be intercepted by the firewall, which will be configured to allow only particular information through.

In many cases, it is best to put the web server behind the firewall as well. The firewall can let through HTTP requests while protecting the server.

**Proxy Server**

A special kind of Internet access server is a *proxy server*, which is part accountant and part traffic cop. Proxy servers are used to funnel all Internet traffic through a single location, and because of this central point, they can effectively manage Internet traffic. Notice in Figure 1.5 that clients on the
network direct all requests for Internet information to the proxy server \Trantor. The server then checks for a number of things:

1. Is the user allowed Internet access?
2. Is the user allowed to go to the intended website?
3. Is the web page already requested in cache?

**Figure 1.5** A proxy server at work

![Proxy Server Diagram]

Although a standard firewall can perform the first two of these tasks, the proxy’s ability to cache pages for users makes it invaluable in saving on limited bandwidth. If three users request a page, only the first user’s request actually hits the Internet—the other users are then given the page from inside the cache.

One service that both firewalls and proxy servers provide is logging of user requests. Administrators can parse the log files for any of the key unacceptable words or phrases, or just check to see who is downloading .mp3s at work.

---

**Note**

Most of us didn’t sign up as network engineers with the intention of becoming morality police or productivity enforcers. Still, it is an unfortunate fact that this is a part of the job on modern networks.

Most routing services require little in the way of resources, with extremely fast network connections being the key to their success. Making sure that these machines have quality network cards and up-to-date network card drivers is critical. Also, because the proxy server caches large amounts of data, it may need a large drive array.
For a great look at how routers, firewalls, and proxy servers work, check out the excellent film, *Warriors of the Net*. It is available to view free at www.warriorsofthe.net, but you will need a fast connection, as the high-res version is 150MB. It's worth the time and the wait, though.

**Remote Access Services**

Ten years ago, a user leaving for a business trip must have felt an amazing sense of freedom. No cell phones! No laptop! Check in by phone every day or so, and otherwise you were safe from whatever disasters might need your attention at the office.

Not so anymore.

As computer-based tasks become a more critical part of the normal workday, networks have evolved to provide easier access to resources for users who can't get to their desks. Among the most important tools available to us who use laptops are remote access servers. These are servers that allow remote users to function on the network as though they were in the office—even if they are thousands of miles away!

Two of these are the traditional dial-in server and the virtual private networking (VPN) server. Each of these has advantages and drawbacks.

**Dial-In Server**

A dial-in server is essentially a router that has a modem as one of its network interfaces. The server answers calls coming in from remote clients, authenticates them with the network, and then acts as a conduit, allowing them to access resources on the network.

In order for this process to work, a dial-in protocol must also be available. The most common of these are Point-to-Point Protocol (PPP) and the less-used SLIP (Serial Line Internet Protocol). PPP is newer, and is generally more efficient because it has error-checking mechanisms built into it. For more information on how PPP and SLIP differ, check out www.ccsi.com/survival-kit/slip-vs-ppp.html for one ISP's explanation of the two. SLIP is not used much anymore, but it is good to at least recognize the acronym, just in case.

The hardware requirements of a dial-in server can be quite specialized because the task of providing many—often dozens—of modem connections is beyond the comm port capabilities of a standard machine. Specialized expansion boards by companies like Digi (www.digi.com) allow a server to support this higher hardware level.
In Figure 1.6, a Windows 2000 Server is using the Routing and Remote Access Service (RAS) to support dial-in clients. The clients authenticate to a Windows domain controller on the network, and are then able to connect for e-mail and file access.

**Figure 1.6** A dial-in configuration for Windows

**VPN Server**

A virtual private network (VPN) is similar to a dial-in connection in that it allows users to access their network remotely. Unlike standard dial-in, though, a VPN connection involves a two-step process:

1. Users attach to the public Internet using a dial-up or by configuring their machine to use a high-speed connection.

   Once an Internet link is established, users can start a VPN client to make a connection across the Internet to a VPN server on their own network. This server also needs to have a separate Internet connection. This process involves creating a secure tunnel connection through the existing connection established in step 1. This secure connection is called a Point-to-Point Tunnel Protocol (PPTP) connection.

   The VPN connection is encrypted, and because all communication is encapsulated within the VPN protocol, users can access network resources through the VPN that they would otherwise be unable to see using standard TCP/IP connectivity. Figure 1.7 shows a common VPN configuration. Note how similar this process is to the one shown in Figure 1.6.
TCP/IP Services

TCP/IP has been mentioned briefly already this chapter and as you go through this book you will continue to hear about it. As earlier mentioned, this is essentially the only major protocol standard that you can depend on any server to support. Because it is everywhere, certain server functions needed by TCP/IP networking must be included on nearly all networks. We will just mention these here, as Chapter 8, “TCP/IP,” deals in-depth with understanding and configuring TCP/IP networking.

DHCP Server

The Dynamic Host Configuration Protocol is used to simplify TCP/IP configuration on network clients. DHCP servers store the information needed to bring a TCP/IP client onto the network, and when a client first starts up they contact the server to obtain an address, gateway, subnet mask, and DNS server, among other things.

If these terms are not already familiar to you, Chapter 7 alone may not be enough! In that case, Andrew Blank’s *TCP/IP JumpStart* (Sybex, 2000) is a good reference. Vendor-specific TCP/IP books are also available, but the *JumpStart* book is nice because it maintains the vendor neutrality that CompTIA espouses.

The DHCP server itself can run on any platform, and clients from multiple operating systems can use a single DHCP server. It is important to
remember, though, that DHCP requests are done through network broadcasts, so you generally need either a DHCP server or a DHCP relay agent on each network segment. More on relay agents in Chapter 8!

**Name Server**

Computers talk to each other quite happily using numbers, as digital information all boils down to ones and zeros eventually. Human beings, on the other hand, generally have an easier time with information presented to them in the form of words and text characters. Because of this, name servers allow both people and machines to have their own way. There are two primary types of name servers to keep in mind:

**DNS Server**  The Domain Naming System (DNS) has been in use for nearly two decades now, and is the worldwide standard for identifying computers on TCP/IP networks. DNS servers resolve TCP/IP host names to IP addresses. In Figure 1.8, the host Client1 requests access to server1.sybex.com. The name is resolved by the DNS server, and Client1 can start the connection.

**WINS Server**  Figure 1.8 also demonstrates the functionality of a WINS server. WINS stands for Windows Internet Naming Service, and the “Windows” part of that is a pretty good clue that this is not a vendor-neutral service. Microsoft has been using its own naming structure—NetBIOS naming—since the days of DOS, and a WINS server is used to support this in a TCP/IP environment. Client1 can also access Server2, but does so through a WINS server rather than a DNS server.

**FIGURE 1.8** Name servers in action
A configuration like the one in Figure 1.8, where both DNS and WINS servers are in use, is common. These services can coexist, and even can help each other out on occasion (see Chapter 7 for more on that).

**Management Server**
TCP/IP also provides a protocol specifically designed for network management functions. The aptly named Simple Network Management Protocol (SNMP) is used to allow a server on the network to collect information about other devices and issue commands in return. We won’t spend a lot of time on SNMP, as it is doubtful you will find detailed SNMP questions on the exam. Even so, [www.snmp.org/protocol/](http://www.snmp.org/protocol/) is a good place to go for an overview of what SNMP is about.

In most cases, naming and management services are almost unnoticeable in terms of their effect on the server. Even a very small server can support thousands of clients with no problems. Again, the key is having sufficient bandwidth to the server.

**User Services**
Finally we arrive at the services that users can see and interact with. When most people talk about a server, they are concerned with the tasks discussed below. This does not, of course, mean that they are the most important services. Like a quarterback on a football team, user services get all the press and most of the resources. The underlying network services listed above, though, are as important and underappreciated as offensive linemen!

Notice that when services are started, the result is significant resource usage. You should be able to identify how to plan for each of the following types of servers by planning to boost critical resource needs.

**File Server**
The classic task of a network server is to store information that needs to be shared among multiple users; in this role it is known as a *file server*. To successfully store and share information, the server normally has to have a few different elements in place. First, some sort of security needs to be present to protect the files on the server. Different network operating systems handle security in very different ways, but in all cases the server needs to be able to ensure that users do not have access to files they should not see. Servers can
also make more subtle distinctions, such as allowing users to read a file but not modify it.

The requirements for a file server are heavily weighted toward its hard drives. Server hardware often comes with multiple drives, because file servers are expected to store enormous amounts of data. File servers also need fast drives—Compaq uses 10,000RPM drives in its servers. The drive controller is also important, as is the network bandwidth available to the server. In Chapter 4, “Storage Devices,” we will examine server drive configuration, and you will notice that SCSI hardware is the overwhelming choice for servers. This is because SCSI is faster and more expandable than IDE/EIDE.

It is interesting to consider that a traditional web server is actually nothing more than an Internet-based file server. Same with FTP servers. Web servers receive requests for HTML pages and serve those pages out, just as a file server sends out .doc and .txt files.

Print Server

If you were interning to be a server on a network, it is likely that you would start as a print server. Print servers require very little in the way of resources, outside of requiring sufficient drive space to store files submitted for printing. Even this is a relatively small requirement, because print jobs are generally stored only until they are printed, at which time they are deleted. The process of network printing is enumerated below. Two terms you should be familiar with when discussing printing are queue and spool:

Queue A queue is a list of documents waiting to be printed. The term also describes the location where these documents are held.

Spool Spooling is the process of writing a document into the queue. The queue is often called the spool file in fact. Spooling allows a print job to be sent to the server even if the printer is busy, thereby freeing up the client to continue on other tasks.

1. The client chooses to print a document. Part of this process involves choosing a printer (or accepting the default printer).

2. The client’s printer driver is used to format the document for printing on the particular printer chosen.

3. The document is submitted as a job to a local print queue on the client. This is optional, but as it immediately frees up the client PC to work on other tasks, local spooling is pretty standard.
4. The job is then sent from the local print spool to the network print server. This server has another print queue, and the document is placed here.

5. The print job is placed in line with the jobs of other users, and when the printer is prepared to print it, the job is spooled out to the printer.

6. The printer produces the document, and reports back to the server, which deletes the job from its queue.

It is possible to tell the print server to keep all print jobs rather than delete them. While this is good for seeing what has printed, it can eat up drive space and is not normally recommended.

A print server’s primary task is to interface with machines that are painfully slow by computer standards—even the fastest printers move at a glacial pace compared to PC speeds. Because of this, print servers require minimal hardware, and print services can often be combined with other tasks rather than having a dedicated print server.

Application Server

File and print servers are in many ways the backbone of a network—application servers are its brain. App servers are machines running server processes that perform tasks on behalf of users, or interact with client machines in the completion of tasks.

There are a number of different application servers, but three of the most common are these:

- Database server
- E-mail server
- Active web server

The key to a server being classified as an application server rather than a file server has to do with how much work the server does on the data before sending it to the client. A great example of this can be found in Microsoft’s database family.

Microsoft Access is a database program that can share a database among multiple users. Because of this, the Access data file itself can be placed on a server and made available to network users. At that point, the server is sharing out a database, but it is not an application server. The reason for this is
that if a client requests information from the database, the process shown in Figure 1.9 is initiated.

**Figure 1.9** Requesting data from an Access database

Notice that the client needs only a specific set of data, yet the server sends the entire database across the network to the client, which is then responsible for sorting out what it wants and discarding the rest of the information. This is inefficient in two critical ways:

1. Time and bandwidth are wasted transferring unneeded rows of data.
2. Processing of the query is done on the less powerful client, meaning it will take longer to complete the task.

**Client-Server Architecture**

The solution to this problem is the use of a client-server architecture, such as the one available in Microsoft’s SQL Server. Client-server applications are computer programs that are specifically designed to use the processing power of both the server and the client machines in the completion of their tasks. Generally this means that the client makes an initial request to the server, and the server then does some initial processing on the request. The result of that processing is then returned to the client, or to another machine for additional work to be done with it.

If more than just a single client and server are involved, this is called an “n-tier” architecture; n stands for the number of machines used in the processing, meaning you could have a client and two servers involved in the transaction, and it would be a “3-tier” design.

Microsoft SQL Server is a server-side application that runs as a service and works with clients to ensure that they are given only the data they request.
Figure 1.10 shows the same request being issued by the client as in Figure 1.9, but with a significantly different response.

**FIGURE 1.10 Requesting data from SQL Server**

Do you see how this time the server has actually looked at what the client needs and has preselected the data? By doing this, both the network and the client are less heavily taxed, and the server is able to justify its expensive hardware by actually doing something.

Because a large database server may be doing tasks for dozens—or even hundreds—of users all at once, the hardware requirements on an application server can be extreme. Moreover, because app servers do a lot of “thinking,” faster processors or multiple processors can be crucial.


**Internet Server**

The last server type we will consider is one intended to deal specifically with web-related or other Internet-related client requests. A number of Internet services can be provided by network servers, but probably the most common of these are the web server, the mail server, and the FTP server. Increasingly, though, streaming media servers, online database servers, and Internet-specific application servers are coming into use.

**Web Server**

The World Wide Web started out as a collection of HTML (HyperText Markup Language) pages stored on Internet servers. Over the past few years,
though, web servers have gotten progressively more complex, and HTML has evolved from a static file server technology to an active client-server model.

The interaction between web servers and web clients (browsers) has now become quite complex. Java, ActiveX, server-side scripting, and database connectivity through the Web have all increased the power and potential of web servers. Many enterprises now find that their web servers are an integral part of both daily business environment through intranets and web-based applications.

**FTP Server**

FTP servers, on the other hand, remain very much the same today as they were 10 years ago. An FTP server is just a file server for the Internet, operating over the FTP protocol. Clients connect to the server, authenticate, and add (PUT) or retrieve (GET) files just as you would on any file server.

The hardware requirements for web servers are fluid, as these servers can support a few concurrent (simultaneous) connections or a few thousand. As your expectation of the number of people using the site rises, so should your hardware levels.

**Mail Server**

There are a number of different e-mail server options available for use with your network. Most NOS vendors have e-mail packages available for their server operating systems, and a number of freeware or shareware e-mail servers are in use as well.

Besides providing for the critical ability to send and receive messages, e-mail servers can filter out inappropriate messages, provide protection from e-mail borne viruses attempting to enter the system, and act as a repository of information and communication data for the organization.

Because they do so much more than just shuffle mail around, these applications are often referred to as groupware rather than just as e-mail servers.

**Summary**

In this chapter, we have discussed what servers are and how to identify server-class hardware and software. Knowing how to tell what hardware components are appropriate, and which operating systems are designed for
server work, is critical when you are choosing a new server or deciding whether an existing box is up to a new task.

We also looked at a sampling of the jobs that servers do, and examined what types of hardware are needed for certain tasks. If you haven’t already, spend a bit of time browsing the Internet links associated with the topics in this chapter. There is a lot of good information there, and Web data hunting is among the most important skills you will need to develop as a server admin!

Throughout the rest of the book, you will take a Chapter Review Test. Each test will consist of 20 questions designed to quiz you on the objectives and content that you reviewed within the chapter. As stated, this chapter does not cover any particular exam objectives—but, to keep your test-taking skills sharp, we included 20 questions to reinforce some of the material you’ve just reviewed. Much like the Assessment Test you took in the Introduction, this test will help you target areas you may need to refresh before forging ahead with the exam preparation. Good luck!

Exam Essentials

Know what a server is. Servers can be hardware or software that provides a service for other devices connected to the network.

Know the characteristics of server operating systems. This includes scalability, security, stability, and client prioritization.

Know the benefits of using server hardware. Expandability, dependability, quality, and redundancy are the benefits of using server hardware over server software.

Be familiar with common server roles. Servers can perform the following roles within a network: security (account management, authentication) and directory services.

Know the major routing protocols. RIP and OSPF are the main routing protocols used today.

Know what a proxy server is. Proxy servers perform Internet tasks on behalf of the computers on the network.

Know the main types of remote access servers. This includes dial-in servers and VPN servers.
Be familiar with the different types of user services that a server can perform. Servers can fulfill the following user services: file server, print server, application server, Internet server, web server, FTP server, and e-mail server.

Key Terms

Before you take the exam, be certain you are familiar with the following terms:

- application servers
- authentication
- biometrics
- client-server
- clustering
- directory services
- servers
- downtime
- file server
- firewall
- load balancing
- network operating system (NOS)
- Open Shortest Path First Protocol (OSPF)
- operating system (OS)
- Point-to-Point Protocol (PPP)
- proxy server
- queue
- redundancy
- Routing Information Protocol (RIP)
- server
- smart card
- spooling
- spool file
- uptime
- virtual private network (VPN)
Review Questions

1. Which technology allows a number of servers to share resources and create a single virtual server out of a number of machines?
   A. Failover  
   B. Clustering  
   C. Scalability  
   D. Mirroring

2. Which server is used to funnel local Internet requests through a single location?
   A. Proxy server  
   B. Firewall server  
   C. VPN server  
   D. Directory Services server

3. Which of the following is not a reason for purchasing server hardware:
   A. It is less expensive.  
   B. Multiprocessor support.  
   C. Expanded software support.  
   D. Expanded hardware support.

4. What is a key problem with using a desktop PC as a server?
   A. Can’t install NOS software.  
   B. Unable to add security features.  
   C. Case does not have space or power capacity needed.  
   D. Nothing. Desktops are the recommended server platform.
5. Which term is used to describe a server failure that causes users to be unable to use the system?
   A. Restore
   B. Backup
   C. Uptime
   D. Downtime

6. Which type of password authentication is the least secure?
   A. Kerberos
   B. Plain text
   C. Smart cards
   D. All are equally secure

7. x.500 is a standard of __________.
   A. Microsoft
   B. CompTIA
   C. ITU
   D. OSI

8. Which of the following are routing protocols?
   A. RIP
   B. DMZ
   C. OSPF
   D. SLIP

9. PPTP is used with what type of network service?
   A. DNS resolution
   B. Authentication
   C. Virtual private networking
   D. Firewall access
10. A DNS server is what type of server?
   A. Application
   B. File
   C. Naming
   D. Remote Access

11. What is a server (select all that apply)?
   A. Computer software designed to assist other computers on a network
   B. Computer hardware optimized for the task of running server software
   C. A computer within a network that is used to perform advanced network calculations
   D. A user’s computer

12. Which of the following is not a priority consideration in deciding on a network operating system?
   A. Scalability
   B. Security
   C. Ease of administration
   D. Stability

13. What is clustering?
   A. Grouping of client computers together on a network
   B. Using more than one server on a network
   C. Grouping servers together to share resources for users
   D. Linking servers together to share work loads
14. What is load balancing?
   A. Using more than one server to perform a single job
   B. Using more than one server on a network
   C. Sharing a single server across multiple resources
   D. Adding more than one component within a server to prevent single component failure

15. What is considered the basic building block of network security?
   A. Users
   B. Servers
   C. Client operating systems
   D. Network operating systems

16. Which of the following is not a secure password recommendation?
   A. Requiring a minimum password length.
   B. Requiring passwords change monthly.
   C. Not letting users use old passwords.
   D. Maintaining the same passwords locally as through remote access.

17. Which of the following is not a form of authentication?
   A. Biometrics
   B. Encryption
   C. Write-protect tabs
   D. Plain text

18. What type of protocol is RIP?
   A. Distance vector
   B. Link state
   C. NetBIOS
   D. IPX/SPX
19. What is a queue?
   A. A list of printing protocols
   B. A type of printer driver
   C. A list of documents waiting to print
   D. A pathway between a computer and a printer

20. Which of the following is not a type of application server?
   A. Database server
   B. E-mail server
   C. Web server
   D. TCP/IP server
Answers to Review Questions

1. B. Clustering allows two or more PCs to act as a single server, providing higher availability and performance than a single PC could handle. They can share an IP address and use the same data array.

2. A. A proxy server is a specialized Internet access server. Just as the word proxy implies, it acts as an intermediary between a client and a server. Because proxy servers are used to funnel all Internet traffic through a single location, they can be extremely effective in managing Internet traffic. This central point allows the firewall software to filter requests more easily.

3. A. Server hardware is expandable and flexible, but it isn’t cheap. The expense of server hardware is the key hangup with purchasing server-class hardware in many organizations.

4. C. Although it is possible to use a standard desktop PC as a server, these machines are generally not designed to handle the multiple drives, multiple processors, or large amounts of RAM that servers need.

5. D. **Uptime** is how long the server has been running since its last shutdown, while **downtime** is the amount of time the server is unavailable.

6. B. Plain text passwords are not encrypted, and can easily be intercepted by others on the network. Kerberos and smart card technologies are both encrypted, and are far more secure.

7. C. The ITU created the x.500 directory structure that is the basis for the NetWare Directory Service (NDS), the Microsoft Active Directory, and other network directory services.

8. A, C. RIP and OSPF are both used to help routers build routing tables. A DMZ is a border area between a public and private network, and SLIP is a dial-up protocol.

9. C. PPTP and L2TP are both used with VPN access. The other options are not directly associated with PPTP.

10. C. DNS stands for Domain Naming System, and DNS servers are using for maintaining and resolving TCP/IP host names.

11. A, B. Servers can be either hardware or software that assist other computers on the network.
12. C. Although ease of administration may be a consideration, it is not a priority in deciding on a network operating system.

13. C. Clustering is grouping servers together to share resources for users with redundancy.

14. A. Clustering is using more than one server to perform a single task. This prevents one server from becoming overrun with requests.

15. A. Due to the fact that they control their own passwords, users are considered the basic building block of network security. If users do not keep passwords secure, the entire network security is jeopardized.

16. D. Allowing users to maintain a common password for both internal and remote connections compromises network security.

17. C. Write-protect tabs are not a form of authentication but rather a form of data protection.

18. A. RIP is considered to be a distance vector routing protocol.

19. C. A queue is a location where documents are kept in order until they can be printed.

20. D. TCP/IP is not a type of application server. It is a network protocol.