

# Chapter | 2

## Networks and the Internet

### *Chapter Objectives*

**After reading this chapter and completing the exercises, you will be able to do the following:**

- Identify each of the major protocols used in network communication (for example, FTP and Telnet), and what use you can make of each
- Understand the various connection methods and speeds used on networks
- Compare and contrast a hub and switch
- Identify what a router is and what it's used for
- Understand how data is transmitted over a network
- Explain how the Internet works and the use of IP addresses and URLs
- Recount a brief history of the Internet
- Use network utilities such as these: `ping`, `IPConfig`, and `tracert`
- Describe the OSI model of network communication and the use of MAC addresses

### **Introduction**

To be able to manage network security, you will need knowledge about how computer networks operate. Those readers who already have a strong working knowledge of network operations may choose to skim this chapter or perhaps give it a quick read as a review. For other readers new to computer networking, studying this chapter will give you a basic introduction to how networks and the Internet work, including a history of the Internet. This understanding of networks and the Internet will be crucial to your comprehension of later topics presented in this book.

We will begin by examining the basic technologies, protocols, and methods used for networks and the Internet to communicate. Then we will take a look at the history of the Internet. This information forms the background knowledge you will need to understand various cyber attacks and how they are defended against. In the exercises at the end of the chapter, you will be able to practice using some protective methods, such as `IPConfig`, `tracert`, and `ping`.

## Network Basics

Getting two or more computers to communicate and transmit data is a process that is simple in concept but complex in application. Consider all the factors involved. First you will need to physically connect the computers. This connection usually requires either a cable that plugs into your computer or wireless connection. The cable then is plugged either directly to another computer or into a device that will, in turn, connect to several other computers.

Of course, wireless communication is being used with more frequency, and wireless connecting, obviously, doesn't require a cable. However, even wireless communication relies on a physical device to transmit the data. There is a card in most modern computers called a *network interface card*, or NIC. If the connection is through a cable, the part of the NIC that is external to the computer has a connection slot that looks like a telephone jack, only slightly bigger. Wireless networks also use a NIC; but rather than having a slot for a cable to connect to, the wireless network simply uses radio signals to transmit to a nearby wireless router or hub. Wireless routers, hubs, and NICs must have an antenna to transmit and receive signals. These devices are connective devices that will be explained in detail later in this chapter.

### The Physical Connection: Local Networks

As mentioned, cables are one of the ways that computers are connected to each other. The cable connection used with traditional NICs (meaning not wireless) is an RJ-45 connection. (*RJ* is short for Registered Jack, which is an international industry standard.) In contrast to the computer's RJ-45 jacks, standard telephone lines use RJ-11 jacks. The biggest difference between jacks involves the number of wires in the connector, also called the *terminator*. Phone lines have four wires (though some have six wires), whereas RJ-45 connectors have eight wires.

If you look on the back of most computers or the connection area of a laptop, you will probably find two ports that, at first glance, look like phone jacks. One of the two ports is probably for a traditional modem and accepts a standard RJ-11 jack. The other port is larger and accepts an RJ-45 jack. It would be extremely rare to find a modern computer that did not have a NIC.

This standard connector jack must be on the end of the cable. The cable used in most networks today is a Category 5 or 6 cable abbreviated as Cat 5 cable or Cat 6 cable. Table 2.1 summarizes the various categories of cable and their uses.

**TABLE 2.1** Cable Types and Uses

Category	Specifications	Uses
1	Low-speed analog (less than 1MHz)	Telephone, doorbell
2	Analog line (less than 10MHz)	Telephone
3	Up to 16MHz or 100Mbps (megabits per second)	Voice transmissions
4	Up to 20MHz/100Mbps	Data lines, Ethernet networks
5	100MHz/100Mbps	Most common a few years ago, still widely used
6	1000Mbps (some get 10Gbps)	Most common type of network cable
6a	10Gbps	High-speed networks
7	10Gbps	Very high-speed networks
8	40Gbps	Not yet commonly found

The type of cable used in connecting computers is also often referred to as unshielded twisted-pair (UTP) cable. In UTP, the wires in the cable are in pairs, twisted together without additional shielding. As you can see in Table 2.1, each subsequent category of cable is somewhat faster and more robust than the last. It should be noted that although Cat 4 can be used for networks, it almost never is used for that purpose, as it is simply slower, less reliable, and an older technology. You will usually see Cat 5 cable and, increasingly, Cat 6. You should note that we are focusing on UTP because that is what is found most often. There are other types of cable such as shielded twisted-pair (STP), but they are not nearly as common as UTP.

#### FYI: Cable Speed

Category 6 cable is for the new Gigabit Ethernet. Cat 5 cable works at speeds of up to 100Mbps, whereas Cat 6 works at 1000Mbps. Cat 6 is widely available and has been for several years. However, for Cat 6 to truly function properly, you need hubs/switches and NICs that also transmit at gigabit speeds; thus, the spread of gigabit Ethernet has been much slower than many analysts expected. We will discuss hubs, switches, NICs, and other hardware in more detail later in this chapter.

As shown in Table 2.1, a key specification is speed, measured in Mbps, or megabits per second (though more and more gigabits per second or Gbps is common). You are probably already aware that ultimately everything in the computer is stored in a binary format—namely, in the form of a 1 or a 0. These units are called bits. It takes 8 bits, which equals 1 byte, to represent a single character such as a letter, number, or carriage return. Remember that the data specification for each cable is the maximum that the cable can handle. A Cat 5 cable can transmit up to 100 mega (million) bits per second. This is known as the *bandwidth* of the cable. If multiple users are on a network, all sending data, that traffic uses up bandwidth rather quickly. Any pictures transmitted also use a lot of bandwidth. Simple scanned-in photos can easily reach 2 megabytes (2 million bytes, or 16 million bits) or much more. And streaming media, such as video, is perhaps the most demanding on bandwidth.

If you simply want to connect two computers, you can have the cable go directly from one computer to the other. You would have to use a crossover cable, but you could connect two computers directly. But what do you do if you wish to connect more than two computers? What if you have 100 computers that you need to connect on a network? There are three devices that can help you to accomplish this task: the hub, the switch, and the router. These each use Cat 5 or Cat 6 cable with RJ-45 connectors and are explained in the following sections.

### The Hub

The simplest connection device is the *hub*. A hub is a small box-shaped electronic device into which you can plug network cables. It will have four or more (commonly up to 24) RJ-45 jacks, each called a *port*. A hub can connect as many computers as it has ports. (For example, an 8-port hub can connect eight computers.) You can also connect one hub to another; this strategy is referred to as “stacking” hubs. Hubs are quite inexpensive and simple to set up; just plug in the cable. However, hubs have a downside. If you send a packet (a unit of data transmission) from one computer to another, a copy of that packet is actually sent out from every port on the hub. All these copies leads to a lot of unnecessary network traffic. This occurs because the hub, being a very simple device, has no way of knowing where a packet is supposed to go. Therefore, it simply sends copies of the packet out all of its ports. While you may go to your favorite electronic store and buy something called a “hub,” true hubs no longer exist. What you are really getting is a switch, which we will discuss later in this section.

### Repeater

A *repeater* is a device used to boost signal. Basically if your cable needs to go further than the maximum length (which is 100 meters for UTP), then you need a repeater. There are two types of repeaters: amplifier and signal. Amplifier repeaters simply boost the entire signal they receive, including any noise. Signal repeaters regenerate the signal, and thus don’t rebroadcast noise.

### The Switch

The next connection device option is the *switch*. A switch is basically an intelligent hub; it works and looks exactly like a hub, with one significant difference. When a switch receives a packet, it will send that packet only out the port for the computer to which it needs to go. A switch is essentially a hub that is able to determine where a packet is being sent. How this determination is made is explained in the “Data Transmission” section.

### The Router

Finally, if you wish to connect two or more networks, you use a *router*. A router is similar in concept to a hub or switch, as it does relay packets; but it is far more sophisticated. You can program most routers and control how they relay packets. Most routers have interfaces allowing you to configure them. The more robust routers also offer more programming possibilities. The specifics of how you program the router are different from vendor to vendor, and there are entire books written specifically

on just programming routers. It is not possible to cover specific router programming techniques in this book; however, you should be aware that most routers are programmable, allowing you to change how they route traffic. Also, unlike using a hub or switch, the two networks connected by a router are still separate networks.

## Faster Connection Speeds

The previous explanation covers the connections between computers on a local network, but surely there are faster connection methods. Well, there are; in fact, your Internet service provider or the company for which you work probably has a much faster connection to the Internet. Table 2.2 summarizes the most common high-speed connection types and their speeds.

**TABLE 2.2** Internet Connection Types

Connection Type	Speed	Details
DS0	64Kbps	Standard phone line.
ISDN	128Kbps	Two DS0 lines working together to provide a high-speed data connection.
T1	1.54Mbps	Twenty-four DS0 lines working as one. Twenty-three carry data, and one carries information about the other lines. This type of connection has become common for schools and businesses.
T3	43.2Mbps	672 DS0 lines working together. This method is the equivalent of 28 T1 lines.
OC3	155Mbps	All OC lines are optical and do not use traditional phone lines. OC3 lines are quite fast and very expensive. They are often found at telecommunications companies.
OC12	622Mbps	The equivalent of 336 T1 lines, or 8,064 phone lines.
OC48	2.5Gbps	The equivalent of four OC12 lines.

It is common to find T1 connection lines in many locations. A cable modem can sometimes achieve speeds comparable to a T1 line. Note that cable modems were not listed on the chart simply because their actual speeds vary greatly depending on a variety of circumstances including how many people in your immediate vicinity are using the same cable modem provider. You are not likely to encounter the OC lines unless you work in telecommunications.

## Data Transmission

We've seen, briefly, the physical connection methods; but how is data actually transmitted? To transmit data, a packet is sent. The basic purpose of a cable is to transmit packets from one machine to another. It does not matter whether that packet is part of a document, a video, an image, or just some internal signal from the computer. So what, exactly, is a packet? As we discussed earlier, everything in a computer is ultimately stored as 1s and 0s, called *bits*, which are grouped into sets of eight, called a *byte*. A packet is a certain number of bytes divided into a header and a body. The header is 20 bytes at

the beginning of the packet that tells you where the packet is coming from, where it is going, and more. The body contains the actual data, in binary format, that you wish to send. The aforementioned routers and switches work by reading the header portion of any packets that come to them. This process is how they determine where the packet should be sent.

## Protocols

There are different types of network communications for different purposes. The different types of network communications are called protocols. A *protocol* is, essentially, an agreed-upon method of communication. In fact, this definition is exactly how the word protocol is used in standard, noncomputer usage, too. Each protocol has a specific purpose and normally operates on a certain port. (Ports are discussed in more detail later.) Some of the most important, and most commonly used, protocols are listed in Table 2.3.

**TABLE 2.3** TCP/IP Protocols

Protocol	Purpose	Port
FTP (File Transfer Protocol)	For transferring files between computers.	20 and 21
TFTP (Trivial File Transfer Protocol)	A quicker but less reliable form of FTP.	69
Telnet	Used to remotely log on to a system. You can then use a command prompt or shell to execute commands on that system. Popular with network administrators.	23
SMTP (Simple Mail Transfer Protocol)	Sends email.	25
WhoIS	A command that queries a target IP address for information.	43
DNS (Domain Name Service)	Translates URLs into web addresses.	53
HTTP (Hypertext Transfer Protocol)	Displays web pages.	80
POP3 (Post Office Protocol version 3)	Retrieves email.	110
NNTP (Network News Transfer Protocol)	Used for network newsgroups (Usenet newsgroups). You can access these groups over the Web via <a href="http://www.google.com">www.google.com</a> by selecting the Groups tab.	119
NetBIOS	An older Microsoft protocol that is for naming systems on a local network.	137, 138, 139
IRC (Internet Relay Chat)	Used for chat rooms.	194
ICMP (Internet Control Message Protocol)	Packets that contain error messages, informational messages, and control messages.	No specific port
HTTPS	Encrypted HTTP; used for secure websites	443

Each of these protocols will be explained in more detail, as needed, in later chapters of this book. You should also note that this list is not complete, as there are dozens of other protocols; but these are the basic protocols we will be discussing in this book. All of these protocols are part of a suite of protocols referred to as *TCP/IP* (Transmission Control Protocol/Internet Protocol). But no matter the particular protocol being used, all communication on networks takes place via packets, and those packets are transmitted according to certain protocols, depending on the type of communication that is occurring.

### **Ports**

You may be wondering what a port is, especially since we've already talked about the ports that are the connection locations on the back of your computer, such as a serial port, a parallel port, or RJ-45 and RJ-11 ports. A *port*, in networking terms, is a handle, or a connection point. It is a numeric designation for a particular pathway of communications. You can think of a port like a channel number on your television. You may have one cable coming into your TV, but you can tune to a variety of channels. The combination of your computer's IP address and port number is referred to as a *socket*. All network communication, regardless of the port used, comes into your computer via the connection on your NIC.

So, the picture we've drawn of networks, to this point, is one of machines connected to each other via cables, and perhaps to hubs, switches, or routers. These networks transmit binary information in packets using certain protocols and ports.

## **How the Internet Works**

Now that you have a basic idea of how computers communicate with each other over a network, it is time to discuss how the Internet works. The Internet is essentially a large number of networks that are connected to each other. Therefore, the Internet works exactly the same way as your local network. It sends the same sort of data packets, using the same protocols. These various networks are simply connected into main transmission lines called *backbones*. The points where the backbones connect to each other are called *network access points (NAPs)*. When you log on to the Internet, you probably use an *Internet service provider (ISP)*. That ISP has a connection either to the Internet backbone or to yet another provider that has a backbone. So, logging on to the Internet is a process of connecting your computer to your ISP's network, which is, in turn, connected to one of the backbones on the Internet.

### **IP Addresses**

With tens of thousands of networks and millions of individual computers communicating and sending data, a predictable problem arises. That problem is ensuring that the data packets go to the correct computer. This task is accomplished in much the same way as traditional "snail" letter mail is delivered to the right person: via an address. With network communications, this address is a special one, referred to as an "IP" address. An IP address can be IP version 4 or version 6.

## IPv4

An *IP address* is a series of four values, separated by periods. (An example would be 107.22.98.198.) Each of the three-digit numbers must be between 0 and 255; thus, an address of 107.22.98.466 would not be a valid one. These addresses are actually four binary numbers; you just see them in decimal format. Since each of these numbers is really just a decimal representation of 8 bits, they are often referred to as octets. So there are four octets in an IP v4 address. Recall that a byte is 8 bits (1s and 0s), and an 8-bit binary number converted to decimal format will be between 0 and 255. So you don't have to do the math yourself, I will tell you that this rule means there are a total of over 4.2 billion possible IP addresses. You should not be concerned, however, that we will run out of new IP addresses soon. There are methods already in place (which are discussed later) to extend the use of addresses.

### FYI: Converting Binary Numbers

For those readers not familiar with converting decimal to binary, there are several methods, one of which is shown here. You should be aware that the computer will do this for you in the case of IP addresses, but here's one way—and perhaps the simplest—this is done:

Divide repeatedly by 2, using “remainders” rather than decimal places, until you get down to 1. For example, convert decimal 31 to binary:

$$31/2 = 15 \text{ Remainder } 1$$

$$15/2 = 7 \text{ Remainder } 1$$

$$7/2 = 3 \text{ Remainder } 1$$

$$3/2 = 1 \text{ Remainder } 1$$

$$1/2 = 0 \text{ Remainder } 1$$

Now read the remainders from bottom to top: The binary equivalent is 11111.

The IP addresses come in two groups: public and private. The public IP addresses are for computers connected to the Internet. No two public IP addresses can be the same. However, a private IP address, such as one on a private company network, only has to be unique in that network. It does not matter if other computers in the world have the same IP address because this computer is never connected to those other worldwide computers. Often network administrators use private IP addresses that begin with a 10, such as 10.102.230.17.

It should also be pointed out that often an ISP will buy a pool of public IP addresses and assign them to you when you log on. An ISP might own 1,000 public IP address and have 10,000 customers. Because all 10,000 customers will not be online at the same time, the ISP simply assigns an IP address to a customer when he logs on, and the ISP unassigns the IP address when the customer logs off.

The address of a computer tells you a lot about that computer. The first byte (or the first decimal number) in an address tells you to what class of network that machine belongs. Table 2.4 summarizes the five network classes.



**TABLE 2.4** Network Classes

Class	IP Range for the First Byte	Use
A	0–126	Extremely large networks. No Class A network IP addresses are left. All have been used.
B	128–191	Large corporate and government networks. All Class B IP addresses have been used.
C	192–223	The most common group of IP addresses. Your ISP probably has a Class C address.
D	224–247	These are reserved for multicasting (transmitting different data on the same channel).
E	248–255	Reserved for experimental use.

These five classes of networks will become more important later in this book (or should you decide to study networking on a deeper level). Observe Table 2.4 carefully, and you probably will discover that the IP range of 127 was not listed. This omission is because that range is reserved for testing. The IP address of 127.0.0.1 designates the machine you are on, regardless of that machine's assigned IP address. This address is often referred to as the *loopback address*. That address will be used often in testing your machine and your NIC. We will examine its use a bit later in this chapter in the section on network utilities.

These particular classes are important as they tell you what part of the address represents the network and what part represents the node. For example, in a Class A address, the first octet represents the network, and the remaining three represent the node. In a Class B address, the first two octets represent the network, and the second two represent the node. And finally, in a Class C address, the first three octets represent the network, and the last represents the node.

There are also some very specific IP addresses and IP address ranges you should be aware of. The first, as previously mentioned, is 127.0.0.1, or the loopback address. It is another way of referring to the network interface card of the machine you are on.

Private IP addresses are another issue to be aware of. Certain ranges of IP addresses have been designated for use within networks. These cannot be used as public IP addresses but can be used for internal workstations and servers. Those IP addresses are

- 10.0.0.10 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

Sometimes people new to networking have some trouble understanding public and private IP addresses. A good analogy is an office building. Within a single office building, each office number must be unique. You can only have one 305. And within that building, if you discuss office 305 it is immediately clear what you are talking about. But there are other office buildings, many of which have their

own office 305. You can think of private IP addresses as office numbers. They must be unique within their network, but there may be other networks with the same private IP.

Public IP addresses are more like traditional mailing addresses. Those must be unique worldwide. When communicating from office to office you can use the office number, but to get a letter to another building you have to use the complete mailing address. It is much the same with networking. You can communicate within your network using private IP addresses, but to communicate with any computer outside your network, you have to use public IP addresses.

One of the roles of a gateway router is to perform what is called network address translation (NAT). That takes the private IP address on outgoing packets and replaces it with the public IP address of the gateway router so that the packet can be routed through the Internet.

### **Subnetting and CIDR**

We have already discussed IP version 4 network addresses; now let's turn our attention to subnetting. If you are already familiar with this topic, feel free to skip this section. For some reason this topic tends to give networking students a great deal of trouble. So we will begin with a conceptual understanding. *Subnetting* is simply chopping up a network into smaller portions. For example, if you have a network using the IP address 192.168.1.X (x being whatever the address is for the specific computer), then you have allocated 255 possible IP addresses. What if you want to divide that into two separate subnetworks? Subnetting is how you do that.

More technically, the subnet mask is a 32-bit number that is assigned to each host to divide the 32-bit binary IP address into network and node portions. You also cannot just put in any number you want. The first value of a subnet mask must be 255; the remaining three values can be 255, 254, 252, 248, 240, or 224. Your computer will take your network IP address and the subnet mask and use a binary AND operation to combine them.

It may surprise you to know that you already have a subnet mask even if you have not been subnetting. If you have a Class C IP address, then your network subnet mask is 255.255.255.0. If you have a Class B IP address, then your subnet mask is 255.255.0.0. And finally, if it is Class A, your subnet mask is 255.0.0.0.

Now think about these numbers in relationship to binary numbers. The decimal value 255 converts to 11111111 in binary. So you are literally "masking" the portion of the network address that is used to define the network, and the remaining portion is used to define individual nodes. Now if you want fewer than 255 nodes in your subnet, then you need something like 255.255.255.240 for your subnet. If you convert 240 to binary, it is 11110000. That means the first three octets and the first 4 bits of the last octet define the network. The last 4 bits of the last octet define the node. That means you could have as many as 1111 (in binary) or 15 (in decimal) nodes on this subnetwork. This is the basic essence of subnetting.

### **CIDR**

Subnetting only allows you to use certain, limited subnets. Another approach is CIDR, or classless interdomain routing. Rather than define a subnet mask, you have the IP address followed by a slash and a number. That number can be any number between 0 and 32, which results in IP addresses like these:

192.168.1.10/24 (basically a Class C IP address)

192.168.1.10/31 (much like a Class C IP address with a subnet mask)

When you use this, rather than having classes with subnets, you have variable-length subnet masking (VLSM) that provides classless IP address. This is the most common way to define network IP addresses today.

## **IPv6**

You have probably heard talk of IP version 6, or IPv6, as an extension of IPv4. Essentially, IP version 4 is limited to 4.2 billion IP addresses. Even with the use of private IP addresses, we will run out of available IP addresses. Think of all the computers, printers, routers, servers, smart phones, tablets, and so on connected to the Internet. IP version 6 was designed to alleviate this problem. And if you looked around in the network settings described in the last section, you probably saw the option to enable IPv6. IPv6 utilizes a 128-bit address (instead of 32), so there is no chance of running out of IP addresses in the foreseeable future. IPv6 also utilizes a hex numbering method in order to avoid long addresses such as 132.64.34.26.64.156.143.57.1.3.7.44.122.111.201.5. The hex address format will appear in the form of 3FFE:B00:800:2::C, for example.

There is no subnetting in IPv6. Instead, it only uses CIDR. The network portion is indicated by a slash followed by the number of bits in the address that are assigned to the network portion, such as

/48

/64

There is a loopback address for IPv6, and it can be written as ::1/128. Other differences between IPv4 and IPv6 are described here:

- **Link/machine-local.**

IPv6 version of IPv4's APIPA or Automatic Private IP Addressing. So if the machine is configured for dynamically assigned addresses and cannot communicate with a DHCP server, it assigns itself a generic IP address. DHCP, or Dynamic Host Configuration Protocol, is used to dynamically assign IP addresses within a network.

IPv6 link/machine-local IP addresses all start with fe80::. So if your computer has this address, that means it could not get to a DHCP server and therefore made up its own generic IP address.

- **Site/network-local.**

IPv6 version of IPv4 private address. In other words, these are real IP addresses, but they only work on this local network. They are not routable on the Internet.

All site/network-local IP addresses begin with FE and have C to F for the third hexadecimal digit: FEC, FED, FEE, or FEF.

- **DHCPv6 uses the Managed Address Configuration Flag (M flag).**

When set to 1, the device should use DHCPV6 to obtain a stateful IPv6 address.

- Other stateful configuration flag (O flag).

When set to 1, the device should use DHCPv6 to obtain other TCP/IP configuration settings. In other words, it should use the DHCP server to set things like the IP address of the gateway and DNS servers.

- M flag

This indicates that the machine should use DHCPv6 to retrieve an IP address.

This is the essence of IPv6. You still have all the same utilities you used with IPv4. However, there is a number 6 after the `ping` or `tracert`, so if your computer has IPv6 enabled, you can use the following:

```
ping6 www.yahoo.com
```

We will be discussing `ping`, `tracert`, and other commands later in this chapter.

## Uniform Resource Locators

After you connect to your ISP, you will, of course, want to visit some websites. You probably type names, rather than IP addresses, into your browser's address bar. For example, you might type in `www.chuckeasttom.com` to go to my website. Your computer, or your ISP, must translate the name you typed in (called a *uniform resource locator [URL]*) into an IP address. The DNS protocol, mentioned in Table 2.3, handles this translation process. So you are typing in a name that makes sense to humans, but your computer is using a corresponding IP address to connect. If that address is found, your browser sends a packet (using the HTTP protocol) to port 80. If that target computer has software that listens and responds to such requests (like web server software such as Apache or Microsoft Internet Information Server), then the target computer will respond to your browser's request and communication will be established. This method is how web pages are viewed.

If you have ever received an Error 404: File Not Found, what you are seeing is that your browser received back a packet (from the web server) with error code 404, denoting that the page you requested could not be found. There are a series of error messages that the web server can send back to your web browser, indicating different situations. Many of these problems the browser handles itself, and you never see the error message. All error messages in the 400 series are *client errors*. This term means something is wrong on your side, not the web server. Messages in the 500 series are *server errors*, which means there is a problem on the web server. The 100 series messages are simply informational; 200 series messages indicate success (you usually do not see these, the browser simply processes them); and 300 series messages are redirectional, meaning the page you are seeking has moved and your browser is then directed to the new location.

Email works the same way as visiting websites. Your email client will seek out the address of your email server. Then your email client will use either the POP3 protocol to retrieve your incoming email or the SMTP protocol to send your outgoing email. Your email server (probably at your ISP or your company) will then try to resolve the address you are sending to. If you send something to

chuck@chuckeasttom.com, your email server will translate that email address into an IP address for the email server at yahoo.com, and your server will send your email there. Note that there are newer email protocols available, but POP3 is still the most commonly used.

Many readers are probably familiar with chat rooms. A chat room, like the other methods of communication we have discussed, works with packets. You first find the address of the chat room; then you connect. The difference here is that your computer's chat software is constantly sending packets back and forth, unlike email, which only sends and receives when you tell it to (or on a predetermined time interval).

Remember that a packet has a header section and that header section contains your IP address and the destination IP address that you are going to (as well as other information). This packet structure will become important as we proceed through this book.

## **What Is a Packet?**

We have mentioned network packets and how they are routed through a network and through the Internet. What we have not discussed is exactly what a packet is. You probably know that network traffic is really a lot of 1s and 0s that are in turn transmitted as voltages (over UTP), light wave (over optic cable), or radio frequencies (over Wi-Fi). The data is divided into small chunks called *packets*.

Packets are divided into three sections. Those are header (actually there are at least three headers, but we will get to that in just a moment), data, and footer. The header will contain information about how to address the packet, what kind of packet it is, and related data. The data portion is obviously the information you want to send. The footer serves both to show where the packet ends and to provide error detection.

As we mentioned, there are usually at least three headers. In normal communications there is usually an Ethernet header, a TCP header, and an IP header. Each contains different information. Combined they have several pieces of information that will be interesting for forensic investigations.

Let's begin with the TCP header. It contains information related to the transport layer of the OSI model. (We will be discussing the OSI model later in this chapter.) It will contain the source and destination port for communications. It will also have the packet number, such as packet 10 of 21.

There is also an IP header. The most obvious useful information are source and destination addresses. The IP header has the source IP address, the destination IP address and the protocol. The IP header also has version number, showing if this is a version 4.0 or 6.0 IP packet. The size variable describes how large the data segment is. There is also information regarding the protocol this packet represents.

The Ethernet header contains information regarding the source MAC address and destination MAC address. When a packet gets to the last network segment in its journey, it is the MAC address that is used to find the NIC that the packet is being sent to.

## **Basic Communications**

The packet headers described in the last section also contain some signal bits. These are single bit flags that are turned on to indicate some type of communication. A normal network conversation starts with

one side sending a packet with the SYN (SYNchronize) bit turned on. The target responds with both SYN and ACK (ACKnowledge) bits turned on. Then the sender responds with just the ACK bit turned on, and communication commences. After a time, the original sender terminates the communication by sending a packet with the FIN (FINish) bit turned on.

There are some attacks that depend on sending malformed packets. For example, the common denial of service (DoS) attack, the SYN flood is based on flooding the target with SYN packets but never responding to the SYN/ACK that is sent back. Some session hijacking attacks use the RST command to help hijack communications.

## History of the Internet

At this point, you should have a basic understanding of how networks and the Internet work, as well as some familiarity with IP addresses, protocols, and packets. It is also helpful to know the history of the Internet, as many find that this overview helps put all of the material learned thus far into historical perspective.

The Internet traces its roots to the Cold War. One positive thing that can be said about the Cold War is that it was a time of significant investment in science and technology. In 1957, after the Soviet Union launched the Sputnik satellite, the U.S. government formed the Advanced Research Projects Agency (ARPA) within the Defense Department. ARPA's sole purpose was to fund and facilitate research into technology. Obviously, this aim would include weapons technology, but the total focus would also include communications technology.

In 1962, a study by the Rand Corporation proposed devising a communication method wherein data was sent in packets between locations. If a packet was lost, the originator of the message would automatically resend the message. This idea was a precursor to the Internet communication methodologies that would eventually arise.

In 1968, ARPA commissioned the construction of ARPANET, a simple Internet web of four points (called *nodes*): UCLA, Stanford, UC Berkley, and the University of Utah. Although no one knew it at the time, this small web was the birth of what would become the Internet. At this point, ARPANET had only these four nodes connected.

The year 1972 was a milestone for the development of the Internet, in more than one sense. That year ARPA was renamed DARPA, the Defense Advanced Research Projects Agency. Also that year, Ray Tomlinson invented the first email program. At this point, four years after the birth of ARPANET, there were 23 hosts on the network. (A *host* is a machine with data on it, to which you can connect; for example, a web server is a host.)

The following year, 1973, would mark the birth of the TCP/IP protocol, which allowed the various computers to communicate in a uniform fashion, regardless of their hardware or operating system.

In 1974, Vince Cerf published a paper on the TCP protocol, and for the first time in computer history used the term *Internet*. In 1976, Ethernet cable was developed (the same cabling we use today),

and DARPA began to require the use of TCP/IP protocol on its network. This year also marked the beginning of widespread distribution of the UNIX operating system. The development of UNIX and the Internet would go hand in hand for many years to come. By this time, 8 years after the birth of ARPANET, there were 111 hosts on the network.

In 1979, a major development occurred: the birth of Usenet newsgroups. These groups are essentially bulletin boards open to the entire world. (Today you can access these groups via newsgroup reader software or via the Web by navigating to [www.google.com](http://www.google.com) and selecting Groups. There are thousands of newsgroups devoted to every topic imaginable.) Just 2 years later, the National Science Foundation (NSF) created CSNET for universities and research centers that were not part of ARPANET. That same year, Cerf proposed connecting CSNET and ARPANET. By 1981, the University of Wisconsin had created DNS (Domain Name Service) so that people could find nodes on the network via a name rather than the actual IP address. At this point (1981), there were 562 hosts on the network.

The early 1980s saw enormous growth in the early Internet. DARPA divided its ARPANET into military and nonmilitary segments, thus allowing more people to use the nonmilitary segment. And the NSF introduced the T1 line (a very fast connection). In 1986, the Internet Engineering Task Force (IETF) was formed to oversee the creation of standards for the Internet and Internet protocols. By this time, the Internet consisted of 2,308 hosts.

A pivotal year for Internet development turned out to be 1990. That year, Tim Berners-Lee, working at CERN laboratories in Europe, developed the *Hypertext Transfer Protocol (HTTP)* and gave the world its very first web pages. Via the HTTP protocol and the *Hypertext Markup Language (HTML)*, people could publish ideas on the Internet for anyone (with a connection) to view. By 1990, there were over 300,000 hosts on the Internet. (Fast-forward to 2004; Tim Berners-Lee receives the first Millennium Prize for contributions to technology. He is widely regarded as the father of the *World Wide Web [WWW]*.)

Internet growth and activity exploded in the 1990s. In 1992, CERN released the invention of web pages to the world at large. In 1993, the first graphical web browser, named Mosaic, was invented. By 1994, Pizza Hut began taking orders via web pages. The Internet has continued to grow; today, there are millions of websites around the world. Every organization has a site, from university departments, government agencies, corporations, schools, and religions to nearly any group you can imagine. Many individuals have personal websites as well. Lots of you will use the Web for banking, shopping, information, and entertainment. Additionally, you likely use email on a daily basis. (By the way, I primarily use email for communication, so that is the best way to contact me if you wish: [chuck@chuckeasttom.com](mailto:chuck@chuckeasttom.com).) The Internet has become a virtual “living level” of interaction in our society. What company does not have a website? What movie release does not have a website? What political candidate does not have a website? In just over three decades, the Internet has become an integral part of our society.

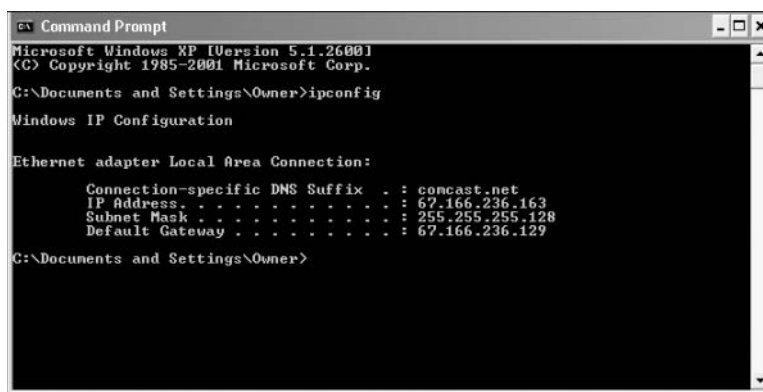
## **Basic Network Utilities**

Later in this book, you will use information and techniques that are based, in part, on certain techniques anyone can perform on her own machine. There are network utilities that you can execute from a command prompt (Windows) or from a shell (UNIX/Linux). Many readers are already familiar with

Windows, so the text's discussion will execute the commands and discuss them from the Windows command prompt perspective. However, it must be stressed that these utilities are available in all operating systems. In this section, you will read about `IPConfig`, `ping`, and `tracert` utilities.

## IPConfig

The first step in studying networks is to get information about your own system. To accomplish this fact-finding mission, you will need to get to a command prompt. In Windows XP, go to the Start menu, select All Programs (in Windows Vista or 7), and then choose Accessories. You will then see an option called Command Prompt. (For Windows 2000 users, the process is identical, except the first option is simply called Programs rather than All Programs.) Next, type in `ipconfig`. (You could input the same command in UNIX or Linux by typing in `ifconfig` once inside the shell.) After typing `ipconfig` and pressing the Enter key, you should see something much like what is shown in Figure 2.1.



```
C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Owner>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : comcast.net
    IP Address. . . . . : 67.166.236.163
    Subnet Mask . . . . . : 255.255.255.128
    Default Gateway . . . . . : 67.166.236.129

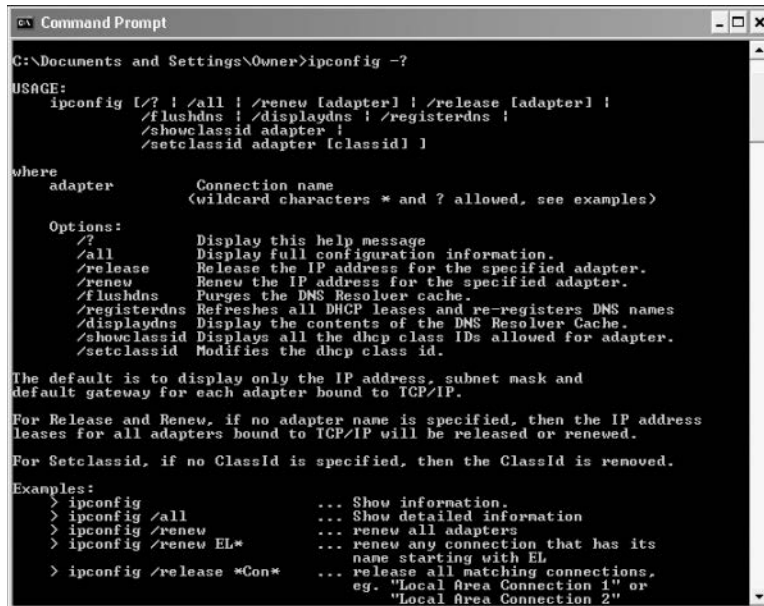
C:\Documents and Settings\Owner>
```

FIGURE 2.1 IPConfig.

This command gives you some information about your connection to a network (or to the Internet). Most importantly, you find out your own IP address. The command also has the IP address for your default gateway, which is your connection to the outside world. Running the `IPConfig` command is a first step in determining your system's network configuration. Most commands that this book will mention, including `IPConfig`, have a number of parameters, or flags, that can be passed to the commands to make the computer behave in a certain way. You can find out what these commands are by typing in the command, followed by a space, and then typing in hyphen question mark, `-?`. Figure 2.2 shows the results of this method for the `IPConfig` command.

As you can see in Figure 2.2, there a number of options you might use to find out different details about your computer's configuration. The most commonly used method would probably be the `IPConfig/all`, shown in Figure 2.3. You can see that this option gives you much more information. For example, `IPConfig/all` gives the name of your computer, when your computer obtained its IP address, and more.





```

C:\Documents and Settings\Owner>ipconfig -?

USAGE:
    ipconfig [/? | /all | /renew [adapter] | /release [adapter] |
    /flushdns | /displaydns | /registerdns |
    /showclassid adapter |
    /setclassid adapter [classid] ]

where
    adapter          Connection name
                     (wildcard characters * and ? allowed, see examples)

Options:
    /?              Display this help message
    /all            Display full configuration information.
    /release        Release the IP address for the specified adapter.
    /renew          Renew the IP address for the specified adapter.
    /flushdns       Purges the DNS Resolver cache.
    /registerdns    Refreshes all DHCP leases and re-registers DNS names
    /displaydns     Display the contents of the DNS Resolver Cache.
    /showclassid   Displays all the dhcp class IDs allowed for adapter.
    /setclassid    Modifies the dhcp class id.

The default is to display only the IP address, subnet mask and
default gateway for each adapter bound to TCP/IP.

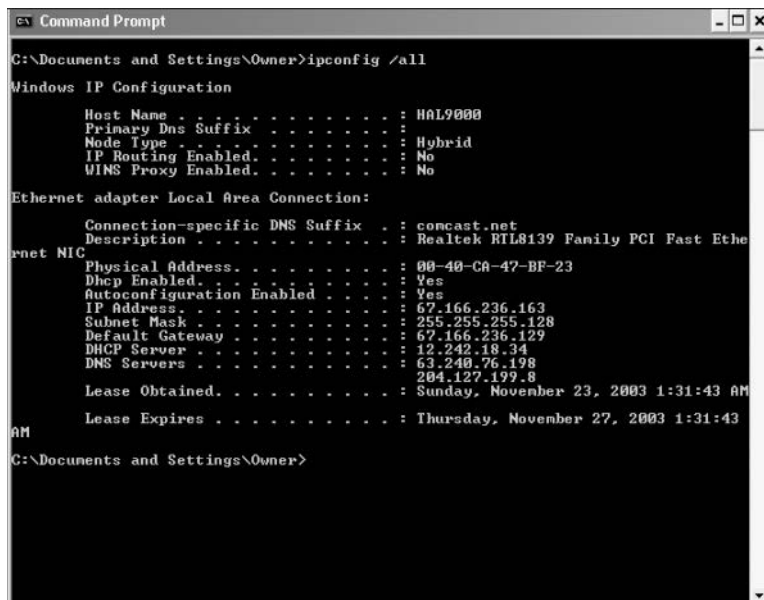
For Release and Renew, if no adapter name is specified, then the IP address
leases for all adapters bound to TCP/IP will be released or renewed.

For Setclassid, if no Classid is specified, then the Classid is removed.

Examples:
> ipconfig          ... Show information.
> ipconfig /all     ... Show detailed information
> ipconfig /renew   ... renew all adapters
> ipconfig /renew EL* ... renew any connection that has its
                        name starting with EL
> ipconfig /release *Con* ... release all matching connections,
                        eg. "Local Area Connection 1" or
                        "Local Area Connection 2"

```

FIGURE 2.2 IPConfig help.



```

C:\Documents and Settings\Owner>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : HAL9000
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : comcast.net
    Description . . . . . : Realtek RTL8139 Family PCI Fast Ethernet NIC
    Physical Address. . . . . : 00-40-C8-47-BF-23
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 67.166.236.163
    Subnet Mask . . . . . : 255.255.255.128
    Default Gateway . . . . . : 67.166.236.129
    DHCP Server . . . . . : 12.242.18.34
    DNS Servers . . . . . : 63.240.76.198
                           204.127.199.8
    Lease Obtained. . . . . : Sunday, November 23, 2003 1:31:43 AM
    Lease Expires . . . . . : Thursday, November 27, 2003 1:31:43 AM

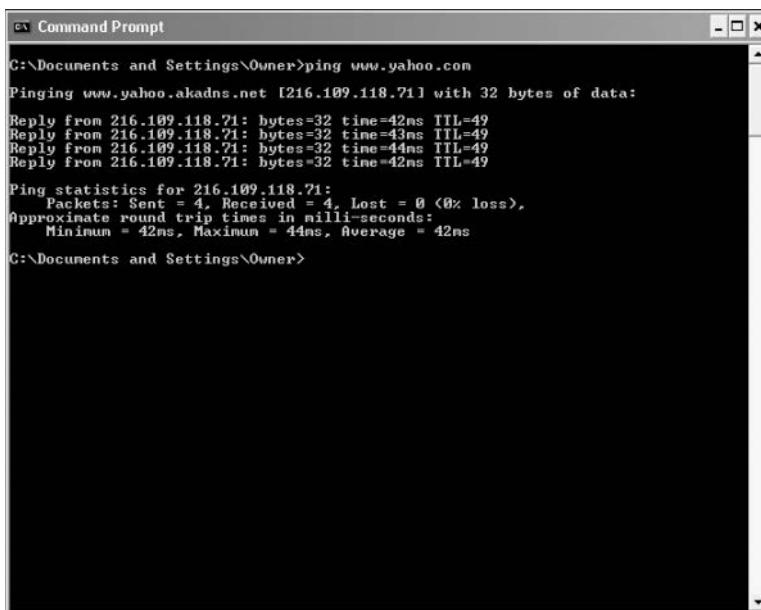
C:\Documents and Settings\Owner>

```

FIGURE 2.3 IPConfig/all.

## Ping

Another commonly used command is `ping`. `ping` is used to send a test packet, or echo packet, to a machine to find out if the machine is reachable and how long the packet takes to reach the machine. This useful diagnostic tool can be employed in elementary hacking techniques. In Figure 2.4 you see a `ping` command executed on `www.yahoo.com`.



```
C:\Documents and Settings\Owner>ping www.yahoo.com
Pinging www.yahoo.akadns.net [216.109.118.71] with 32 bytes of data:
Reply from 216.109.118.71: bytes=32 time=42ms TTL=49
Reply from 216.109.118.71: bytes=32 time=43ms TTL=49
Reply from 216.109.118.71: bytes=32 time=44ms TTL=49
Reply from 216.109.118.71: bytes=32 time=42ms TTL=49

Ping statistics for 216.109.118.71:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 42ms, Maximum = 44ms, Average = 42ms
C:\Documents and Settings\Owner>
```

FIGURE 2.4 Ping.

This figure tells you that a 32-byte echo packet was sent to the destination and returned. The TTL (Time To Live) item shows how many intermediary steps, or hops, the packet should take to the destination before giving up. Remember that the Internet is a vast conglomerate of interconnected networks. Your packet probably won't go straight to its destination; it will take several hops to get there. As with `IPConfig`, you can type in `ping -?` to find out various ways you can refine your ping.

## Tracert

The final command we will examine in this chapter is the `tracert` command. This command is a more or less "ping deluxe." `tracert` not only tells you if the packet got to its destination and how long it took, but also tells you all the intermediate hops it took to get there. This utility will prove very useful to you later in this book. Figure 2.5 illustrates a `tracert` to `www.yahoo.com`. (This same command can be executed in Linux or UNIX, but there it is called `traceroute` rather than `tracert`.)

```

C:\Documents and Settings\Owner>tracert www.yahoo.com
Tracing route to www.yahoo.akadns.net [216.109.118.73]
over a maximum of 30 hops:
  0  8 ms  9 ms  8 ms  10.180.228.1
  1  7 ms  29 ms  8 ms  12.244.113.33
  2  9 ms  9 ms  9 ms  12.244.73.10
  3  9 ms  10 ms  9 ms  gbr5-p80.dlstx.ip.att.net [12.123.17.26]
  4  9 ms  10 ms  8 ms  thr1-p012401.dlstx.ip.att.net [12.122.12.65]
  5  9 ms  8 ms  8 ms  ggr2-p300.dlstx.ip.att.net [12.123.17.81]
  6  10 ms  9 ms  10 ms  att-gw.dc.genuity.net [192.205.32.114]
  7  9 ms  8 ms  10 ms  so-1-2-0.bbr2.Dallas1.Level3.net [209.244.15.165]
  8  40 ms  41 ms  41 ms  so-1-2-0.bbr1.Washington1.Level3.net [64.159.0.1]
  9  41 ms  40 ms  39 ms  ge-7-0.ipcol01.Washington1.Level3.net [64.159.18.31]
 10  43 ms  44 ms  51 ms  unknown.Level3.net [63.210.59.254]
 11  43 ms  42 ms  45 ms  v130.bas1-m.dcn.yahoo.com [216.109.120.142]
 12  42 ms  42 ms  43 ms  p10.www.dcn.yahoo.com [216.109.118.73]
 13
Trace complete.
C:\Documents and Settings\Owner>

```

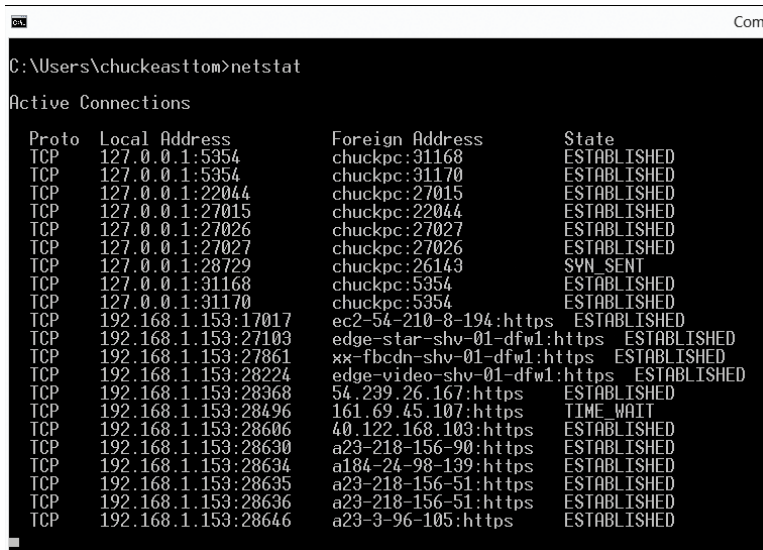
FIGURE 2.5 Tracert.

With `tracert`, you can see (in milliseconds) the IP addresses of each intermediate step listed and how long it took to get to that step. Knowing the steps required to reach a destination can be very important, as you will find later in this book.

Certainly there are other utilities that can be of use to you when working with network communications. However, the three we just examined are the core utilities. These three (`IPConfig`, `ping`, and `tracert`) are absolutely essential to any network administrator, and you should commit them to memory.

## Netstat

`Netstat` is another interesting command. It is an abbreviation for Network Status. Essentially this command tells you what connections your computer currently has. Don't panic if you see several connections; that does not mean a hacker is in your computer. You will see many private IP addresses. This means your network has internal communication going on. You can see this in Figure 2.6.



```

C:\Users\chuckeasttom>netstat
Active Connections
Proto Local Address          Foreign Address         State
TCP    127.0.0.1:5354          chuckpc:31168           ESTABLISHED
TCP    127.0.0.1:5354          chuckpc:31170           ESTABLISHED
TCP    127.0.0.1:22044         chuckpc:27015           ESTABLISHED
TCP    127.0.0.1:27015         chuckpc:22044           ESTABLISHED
TCP    127.0.0.1:27026         chuckpc:27027           ESTABLISHED
TCP    127.0.0.1:27027         chuckpc:27026           ESTABLISHED
TCP    127.0.0.1:28729         chuckpc:26143           SYN_SENT
TCP    127.0.0.1:31168         chuckpc:5354            ESTABLISHED
TCP    127.0.0.1:31170         chuckpc:5354            ESTABLISHED
TCP    192.168.1.153:17017     ec2-54-210-8-194:https  ESTABLISHED
TCP    192.168.1.153:27103     edge-star-shv-01-dfw1:https ESTABLISHED
TCP    192.168.1.153:27861     xx-fbcdn-shv-01-dfw1:https ESTABLISHED
TCP    192.168.1.153:28224     edge-video-shv-01-dfw1:https ESTABLISHED
TCP    192.168.1.153:28368     54.239.26.167:https     ESTABLISHED
TCP    192.168.1.153:28496     161.69.45.107:https     TIME_WAIT
TCP    192.168.1.153:28606     40.122.168.103:https    ESTABLISHED
TCP    192.168.1.153:28630     a23-218-156-90:https    ESTABLISHED
TCP    192.168.1.153:28634     a184-24-98-139:https    ESTABLISHED
TCP    192.168.1.153:28635     a23-218-156-51:https    ESTABLISHED
TCP    192.168.1.153:28636     a23-218-156-51:https    ESTABLISHED
TCP    192.168.1.153:28646     a23-3-96-105:https      ESTABLISHED

```

FIGURE 2.6 Netstat.

## NSlookup

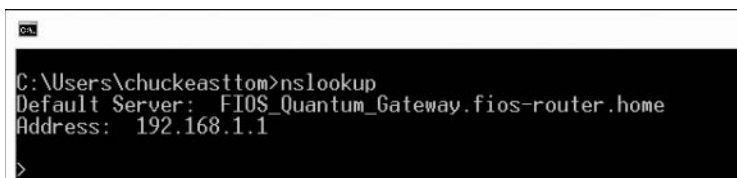
This command is an abbreviation for Name Server lookup. It is used to connect with your network's DNS server. Often it can be used just to verify the DNS server is running. It can also be used to execute commands. Recall from Chapter 1, "Introduction to Computer Security," that we discussed DNS poisoning. One of the first steps in DNS poisoning is to see if the target DNS server will perform a zone transfer. (It should not do so with any machine other than another DNS server that is authenticated in the domain.) That can be attempted with `nslookup`, as shown here:

```

run: nslookup
type: ls -d domain_name <enter>

```

You can see the basic `nslookup` command in Figure 2.7.



```

C:\Users\chuckeasttom>nslookup
Default Server:  FIOS_Quantum_Gateway.fios-router.home
Address: 192.168.1.1
>

```

FIGURE 2.7 nslookup.

## Other Network Devices

There are other devices involved in networking that work to protect your computer from the outside world, some of which were briefly mentioned in Chapter 1. Now we will review a couple of them in a bit more detail. The two most common are the firewall and the proxy server. A *firewall* is essentially a barrier between your network and the rest of the Internet. A personal computer (PC) can be used as a firewall; in many cases, a special router can function as a firewall. Firewalls use different techniques to protect your network, but the most common strategy is packet filtering. In a packet-filtering firewall, each incoming packet is examined. Only those packets that match the criteria you set are allowed through. (Commonly, only packets using certain types of protocols are allowed through.) Many operating systems, such as Windows (all versions since XP) and many Linux distributions, include basic packet-filtering software.

The second very common type of defensive device is a *proxy server*. A proxy server will almost always be another computer. You might see the same machine used as both a proxy server and a firewall. A proxy server's purpose is quite simple: It hides your entire network from the outside world. People trying to investigate your network from the outside will see only the proxy server. They will not see the actual machines on your network. When packets go out of your network, their headers are changed so that the packets have the return address of the proxy server. Conversely, the only way you can access the outside world is via the proxy server. A proxy server combined with a firewall is basic network security. It would frankly be negligent to ever run a network that did not have a firewall and proxy server. We examine firewalls in more detail in Chapter 9.

## Advanced Network Communications Topics

These subjects are not absolutely required for you to understand this book, but they will give you a broader understanding of networks in general. If you have any intention of delving into network security on a professional level, then you will need this information—and much more.

### The OSI Model

Let's begin with the *OSI model*, or Open Systems Interconnection model. This model is a description of how networks communicate. It describes the various protocols and activities, and it tells how the protocols and activities relate to each other. This model is divided into seven layers, as shown in Table 2.5, and was originally developed by the International Standards Organization (ISO) in the 1980s.

**TABLE 2.5** The OSI Model

Layer Number	Layer	Description	Protocols
7	Application	This layer interfaces directly to the application and performs common application services for the application processes.	POP, SMTP, DNS, FTP, and so on
6	Presentation	The presentation layer relieves the application layer of concern regarding syntactical differences in data representation within the end-user systems.	
5	Session	The session layer provides the mechanism for managing the dialogue between end-user application processes.	NetBIOS
4	Transport	This layer provides end-to-end communication control.	TCP, UDP
3	Network	This layer routes the information in the network.	IP, Address Resolution Protocol, Internet Control Message Protocol
2	Data Link	This layer describes the logical organization of data bits transmitted on a particular medium. Data link is divided into two sublayers: the Media Access Control layer (MAC) and the Logical Link Control layer (LLC).	Serial Line Internet Protocol, Point-to-Point Protocol
1	Physical	This layer describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals. In other words, the physical layer is the actual NIC, Ethernet cable, and so forth. This layer is where bits are translated into voltages, and vice versa.	None

Many networking students memorize this model. It's good to at least memorize the names of the seven layers and to understand basically what they each do. From a security perspective, the more you understand about network communications, the more sophisticated your defense can be. The most important thing for you to understand is that this model describes a hierarchy of communication. One layer will only communicate with the layer directly above it or below it.

## Media Access Control (MAC) Addresses

*MAC addresses* are unique addresses for a NIC. (MAC is also a sublayer of the data link layer of the OSI model.) Every NIC in the world has a unique address that is represented by a 6-byte hexadecimal number, and an Address Resolution Protocol (ARP) is used to convert IP addresses to MAC addresses.

When you type in a web address, the DNS protocol is used to translate that into an IP address; then the ARP protocol will translate that IP address into a specific MAC address of an individual NIC.

This brings us to how DNS is accomplished; or rather, how does a URL get translated into an IP address? How does the computer know what IP goes with what URL? There are servers known as DNS servers that are set up just to do this task. If you are on a corporate network, you probably have a DNS server on your network. If you are not, then your ISP has one. These servers maintain a table of IP-to-URL entries. From time to time there are transfers of DNS data, called *zone transfers*, that allow one DNS server to send its changes to another. Across the Internet, there are root DNS servers that are maintained with centralized data for all registered URL/IP addresses.