OBJECTIVES

This chapter covers the following TruSecure-specified objectives for the TICSA exam:

Describe, recognize, or select basic weaknesses in TCP/IP networking.

- TCP/IP is the foundation for any traffic passed on the Internet and most internal networks today. Understanding how TCP/IP functions and what vulnerabilities and security features to watch for is imperative to understanding network security as a whole. This includes how packets are designed, the TCP/IP handshaking process, and the OSI/DARPA models.

Identify the basic security issues associated with system/network design and configuration.

- Understanding the vulnerabilities and possible issues that can arise with major TCP/IP protocols and services can help you understand other security techniques. It is also important to determine what kind of security vulnerabilities and attacks are possible at each layer of the OSI model to consider what software and services you want to install or implement on your critical servers and workstations.

CHAPTER 2

Fundamentals of TCP/IP

Outline

Study Strategies

- The TICSA exam may contain questions relating to both the TCP/IP and OSI models and how various security elements work with those. For instance, at what level do Web servers work? Take some time to come up with a reliable way to remember which layer goes where, how each corresponds with one another, and what you will typically find working on each layer. It may be worth your while to spend a little time reading some TCP/IP primers on `Microsoft.com` or even just doing a search on your favorite Internet search engine for information on the deeper, inner workings of TCP/IP.
- You will probably not see many questions relating to TCP/IP headers and how packets are laid out, but it would be worthwhile to know how they are built. Another important scheme to memorize is how the handshake process goes. Making up a rhyme or acronym for this will help you remember it when you invariably get a question on it during the exam.

INTRODUCTION

Given the wide array of IP-related exploits that a hacker can make use of, the TCP/IP protocol can often be your greatest network security concern. In addition to the fundamentals of securing TCP/IP, this chapter explores various network models, including the OSI and TCP/IP network models, and how security is implemented with them. You learn about IP packets, how they are structured, and how to secure the data being sent using TCP/IP. Finally, you learn about packet routing and other network-level topics, such as Network Address Translation and Secure Sockets Layer.

BASIC TCP/IP PRINCIPLES

- Describe, recognize, or select basic weaknesses in TCP/IP networking.

TCP/IP has seen few changes over the years from its earliest inception, when security wasn't much of a concern. As the use of TCP/IP grew, in 1985 Bob Morris started to realize inherent problems with how TCP/IP connection handshakes between servers and client systems were generated that could allow a hacker to force a connection, despite whether or not they were allowed to. Findings such as these caused changes to the basic TCP/IP design over the years, but as mentioned, the fundamental TCP/IP design, and thus security model, has not changed very much for quite a while.

In addition to the basic TCP/IP security model, you will encounter a variety of security models that may or may not actually refer to a protocol or other network component. For example, in the security model for your office building, you may possibly have key cards, no access after a certain hour, or other similar guidelines and rules that make up the security model for your office. This variance in descriptions is mainly because of a discrepancy in terminology, although technically neither is incorrect. A security model is simply a different way to discuss how security is implemented or designed around a protocol (or other network component). It can also be defined as a security concept that is applied to a network security plan or piece of software.

The level of security you receive on your TCP/IP-enabled network is entirely dependent on the capability and design of your operating systems or software used on the network. In fact, the way that the implementation of TCP/IP (and other network resources) handles security is entirely dependent on the way TCP/IP is designed. TCP/IP security includes how that particular implementation of TCP/IP performs connection handshaking, monitoring of established connections, and handling of errors between systems on a network, among other aspects, such as additional security features or models that are applied alongside or on top of the core TCP/IP protocol. Variances in the security abilities (or entire lack of security) of third-party applications that use TCP/IP can cause you considerable grief. In many cases, the people working with this software package lack the expertise of a major vendor, such as Microsoft, Cisco, or similar companies, in programming TCP/IP security features; however, these companies fall victim themselves more often than not. Pay close attention to these types of specialized or obscure software packages for TCP/IP-related security issues.

IN THE FIELD

TCP/IP COMMON SECURITY ISSUES

In your travels, you will more often than not find more security issues relating to a piece of poorly written software installed on your server than any blatant security problem with your implementation. In many cases, a server that is otherwise bulletproof one day will be completely vulnerable the next after a new vulnerability is uncovered in some third-party application you have installed on your server.

It is very important to get involved in the security community and at the very least subscribe to any security mailing lists or notification agents for the software you have on your network and servers.

TCP Handshakes and Headers

The TCP handshake process and TCP headers are some of the most important issues to keep in mind because they can be some of the

most common targets of hackers. There are three main phases to the connection process that are handled by the TCP header:

- **SYN**. Synchronizes the sequence numbers (the numbers used to identify the packets).
- **FIN**. Signifies that there is no additional data being sent from the sender.
- **ACK**. Identifies the acknowledgment information contained within the packet.

Establishing a TCP connection is a basic three-way handshake that can be considered very similar to the way many people interact. The overall process happens very quickly even though the following steps may seem to contain a lot of information being exchanged.

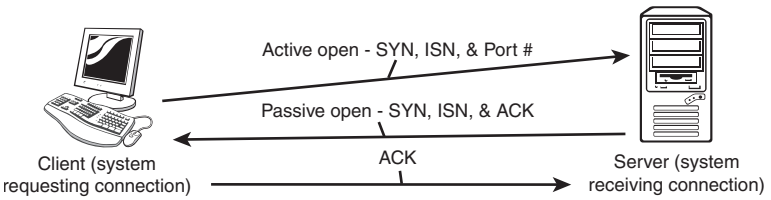
1. The system requesting the connection (client) performs an active open that activates the SYN flag in the TCP header, which also includes the port number being used for this connection (more on this in a bit) and the sequence number including the *Initial Sequence Number (ISN)*. The ISN is randomly generated at the start of a connection and is used to synchronize the client and server before data transfer proceeds.
2. The system receiving the connection (server) performs a passive open by sending a SYN back to the client system that includes the ISN and ACK of the client's SYN information that was sent in Step 1.
3. After the client system receives this information and the ACK from the server, the client system sends back an ACK command acknowledging the information sent to it in Step 2, and the connection is established. It is important to note that while this whole process is occurring, the server reserves some of its limited resources waiting for the remote connection to return an ACK response. Opening many connections but not closing them properly consumes resources on the server—until it stops or crashes entirely.

The process is rather simple, but sometimes can be hard to imagine based on the previous steps, which are outlined in Figure 2.1.

EXAM TIP

Handshake Process You should understand the handshake process as well as possible. The exam typically focuses on this process as well as potential security issues related to the handshake process.

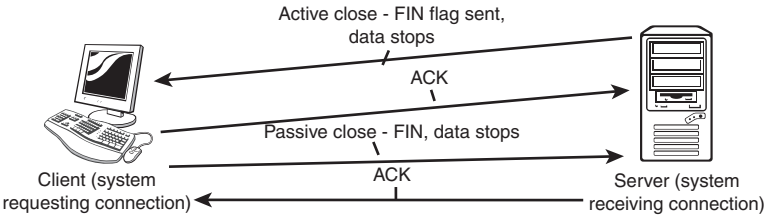
FIGURE 2.1
The TCP handshake process.



After the two systems complete their communication, they need to indicate to each other that the connection is ending. They end the communication cleanly in four steps, as follows (see Figure 2.2):

1. The server performs an active close by sending the FIN flag in one of the packets (even though the client system may break off the connection by, for example, closing the program being used, the server actually handles the ending of the connection). This step ends the flow of data from the server to the client system.
2. The client performs a passive close by sending an ACK back to the server indicating it received the FIN flag.
3. The client then sends its own FIN command to the server to indicate that its connection is ending and the flow of data going from the client to the server stops. Sometimes, Steps 2–3 are listed together (called a FIN-ACK command) because the data is sent typically at the same time. It is broken up here for illustration purposes.
4. Upon receiving the FIN from the client, the server sends back an ACK to the client and the connection is finally terminated.

FIGURE 2.2
The TCP disconnection process.



Sometimes, failing to start or end a TCP connection cleanly can result in hackers taking advantage of this in the form of a DoS attack. If the system isn't aware that the communication has ended, it keeps that particular resource open. This becomes a problem because there are not infinite network resources available. Available network resources are often dictated by network performance and the actual capabilities of the server itself in regard to memory. After the available resources are depleted, the server either stops accepting new connections or crashes. To take advantage of this problem, a hacker simply executes hundreds of thousands of connections and forces them to fail. This, in turn, takes up available resources and brings the server to a slow crawl or worse yet, crashes it completely. With many current TCP/IP stacks, you will see this type of problem less frequently; however, these types of attacks are still more than possible and should be kept in mind.

SYN Flood Attacks

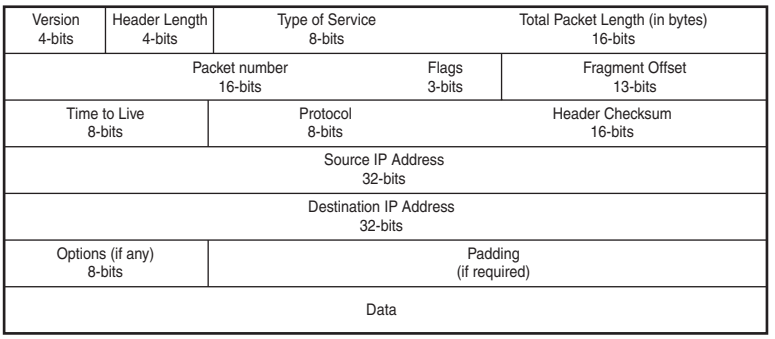
Now that you have taken a closer look at how TCP connections are established and completed, you are better prepared to understand the most common attack against TCP, a SYN flood attack. In a SYN flood attack, the connection is established by issuing a SYN request (as it would normally), except this SYN request is not complete (bad data or the request is left without completing). The hacker then continues to send these same modified SYN requests until the server cannot respond to actual requests by valid client systems, effectively taking it offline. (A system could be taken offline by this type of attack because without any resources available to receive new connections, the system is more or less unavailable for regular use.)

You may have heard of a man-in-the-middle attack, or, as it's sometimes known, hijacking. So, how is this type of attack actually done? As mentioned earlier in this chapter, a vulnerability found in the mid-1980s showed a flaw in how TCP generated the ISN. It was found that determining this number was terribly easy, which led to hackers being able to predict what the sequence number of each following TCP packet would be (remember, each packet has its own number in sequence); by doing so, they could hijack a connection by jumping into the connection with the proper sequence of packets. This is much harder to do today, but it is still more than possible.

Structure of an IP Datagram

Every IP address used on the Internet is 32 bits long (this is not the case when using Ipv6, which is 64 bits) and is used to uniquely identify a system on a TCP/IP network, such as the Internet or a local area network in your office. This address information is typically contained within the *IP header*, which holds different pieces of TCP/IP-related information, and is 20 bytes in size. Among the information contained within an IP header is the IP version number, length of the packet, type of IP service, and other configuration information. In addition to this other configuration information, the destination address (which is also 32-bit, assuming Ipv6 hasn't been implemented) is included with the source address in the datagram. This information is tacked on to a piece of data and called an IP *datagram* (or more commonly called a *packet*). These are the data items that are actually transferred among systems. Larger files have many thousands of packets because they need to be broken up into smaller chunks for transmission. Each one of these smaller chunks has the header information in the datagram so that they will be delivered properly. The basic structure of an IP datagram is shown in Figure 2.3.

FIGURE 2.3
Structure of an IP datagram.



When the data is received by the recipient system (sometimes out of order), it is rebuilt, using all the received datagrams, into a proper set of data. It is this segmentation and potential jumble of datagrams that allows hackers to potentially take advantage of the TCP/IP protocol by slipping in data or modifying the header information. Unfortunately, this situation is an inherent problem with the TCP/IP design that cannot be changed without a major revision on the specification (which is in the works). Using an add-on software

package or additional configuration, such as Secure Internet Protocol (IPSec) or authentication and data encryption, can curb many of the issues associated with this design flaw.

Network Reference Models and TCP/IP

There are many different types of network models that you will encounter, including both TCP/IP and general network security models, the latter leaning toward an actual security implementation on your network. In the following sections, take a look at the two network models that comprise much of the inner workings of networking as we know it today: the OSI model and the TCP/IP (DARPA) model. It is important to understand the two major models that form the basis for most internal networks and the fundamental networking design of the Internet (and as such, the main delivery of attacks by hackers). Later, you take a look at the issues you may run into with different protocols and services, including possible considerations when trying to secure them.

EXAM TIP

OSI and TCP/IP Models Ensure that you are very familiar with the various layers of the OSI and TCP/IP models and how they associate with one another. Of great importance is to know what occurs at each level.

The OSI Network Reference Model

Years ago, the International Standards Organization (ISO) created a networking model that consists of seven individual layers, called the Open Systems Interconnection (OSI) reference model. Each layer of the model (top to bottom) represents a different element of a network communication protocol, as follows:

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

In the next sections, some of the major elements are covered in more detail starting from the bottom (Physical layer), including possible security issues or attacks that can be levied against each one.

EXAM TIP

7-Layer Dip! It can be useful for your own purposes either down the road or when studying for the exam to think of the OSI network model as a "7-Layer Dip" and even come up with a sentence using the first letter from each layer to keep it straight in your head. For example, you can remember the sentence, "Please Do Not Take Sales Person's Advice" to remember the layers of the OSI model (Physical, Data Link, Network, Transport, Session, Presentation, and Application).

Physical Layer

- Identify the basic security issues associated with system/network design and configuration.

This layer comprises the actual physical connection medium, such as the network cable or network card. Little can be done to actually secure a physical medium, outside of locking away sensitive network equipment, such as hubs, switches, and servers, and allowing access only to authorized people. Effectively, if a hacker has direct access to your physical network, there isn't much they can't do unless you make use of encryption to protect any data captured using a network sniffer or data copied down from systems and servers.

One glaringly simple security concept that I have seen in my travels is the consideration of disabling unused network access ports in unused offices or in odd places, such as waiting rooms or in the middle of hallways (these aren't as rare as you might think with the amount of office shuffling that occurs in this day and age). For most organizations, typical practice is to simply wire up to the network any jacks in case a user needs one (for instance, a new employee starts and occupies an office that was otherwise vacant). By disconnecting these types of unused connections from your physical network, you minimize a potential avenue for a hacker to squirrel away in an unoccupied office and hack away on your network with an otherwise unused, but still connected, network jack.

Network Layer

The Network layer is often referred to as the IP layer (see Figure 2.4) and is the layer used mainly for addressing host systems and routing traffic. This layer does not provide any kind of error correction or flow control; rather, it relies on other layers to provide delivery of data from one system to another. Other protocols, such as TCP, User Datagram Protocol (UDP), ICMP, and Internet Gateway Message Protocol (IGMP), are used to provide the delivery of IP traffic, although TCP and UDP are the most common methods of delivery you will encounter. The other two protocols—ICMP and IGMP—are relegated to the IP layer only. Protocols that operate on the Network layer of the OSI are used only to assist in communication between two hosts, rather than the actual delivery of data. Few potential attacks against the Network layer of the OSI model exist because most current operating systems are patched to protect against them.

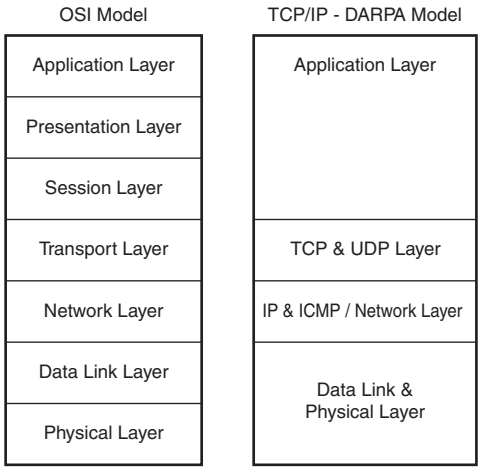


FIGURE 2.4
Comparison of the OSI and TCP/IP models.

The IP (Network) layer uses *ICMP* to transfer information about communication errors and other information relating to the layer. A good example of ICMP in action is when you perform a simple ping on another system—the information you get back is an ICMP-generated message. Although there is little that can be levied against ICMP, it can be used to attack a remote network or host.

Among some of the more high-profile attacks that use ICMP are the *Tribal Flood Network (TFN)* programs and Winnuke. The TFN series of programs varied in name, but all carried the same basic attack plan. This attack plan was to use ICMP traffic to consume all the available bandwidth a site had and, more or less, take the Web site, or any type of server, offline because it would be unable to respond to any requests made by client systems. This attack was, in effect, one of the earliest types of Denial-of-Service attacks in which available resources are consumed to stop or limit the capabilities of a system or server.

The other high-profile attack, *Winnuke*, plagued many people for some time, especially users of the older Windows 95 operating systems (newer operating systems are protected against this particular attack). Winnuke was designed by a hacker to send very specialized ICMP data in the form of a plain and harmless-looking ping request. Any system with an older version of the Microsoft TCP/IP stack, mainly Windows 95, could not properly handle this fake request and caused the network connection to fail and, often, the entire system to crash (hence the *nuke* part of the name). Another

NOTE

Patches To patch a piece of software or operating system is to simply install an update, typically in the form of rewritten or new code. These patches are used to fix problems not found in testing or, in some cases, to add features. Patches are typically released by the vendor on an as-needed basis; although, occasionally, a new security vulnerability forces a smaller patch to be sent out as quickly as possible.

Microsoft has two common types of patches available for their products: *service packs* and *hot fixes*. A service pack is a large collection of smaller patches and fixes that is issued on an as-needed basis (often, once a year). These are geared toward fixing common problems or issues that have arisen since the product release or last service pack and in cases in which urgency is not a factor.

A hot fix is a patch for security, bug fix, or similar urgent issue that is released for immediate download so that the end users can access the fix without delay. This is how most security fixes are released because they are typically of a sensitive nature. Keep in mind that any hot fixes will also be bundled with the rest of the available patches in the service pack when one is issued.

similar attack is called the Ping of Death, which is when a modified packet is sent that is larger than TCP/IP can handle and it causes the system to consume resources, quite possibly crashing the system. Because of attacks such as Winnuke, TFN, or Ping of Death, many networks and Web sites, including `http://www.microsoft.com` and other popular Web portals, do not respond to ICMP traffic any longer.

Transport Layer

The Transport layer is used to control the flow of data between systems. There are two protocols that reside on this layer, TCP and UDP. Each protocol is different in how it provides services and as such requires a slightly different approach when dealing with security.

NOTE

TCP/IP Terminology You will see the term *TCP/IP* used quite a bit, as well as just TCP and IP. As you have learned, TCP and IP each reside on a separate layer of the OSI network model and each has its own capabilities. The two are used together to create a reliable and quick protocol for use on networks, such as the Internet.

The main protocol that you will undoubtedly be dealing with is TCP; this protocol is termed as being connection oriented. This means that for two systems to communicate with each other, they must go through a handshaking procedure and information exchange for the communication to progress. By performing this handshaking process, as you read earlier in the TCP handshake process section of this chapter, you gain the ability to have validated delivery of your information, as much as it is possible to validate with the potential for attack. This is a bit different when compared to other protocols, such as UDP, that do not require a negotiated connection prior to sending data, which is somewhat like firing a gun in the dark—you don't really know where the bullet may go.

The real benefit to the negotiated connection is that, as you learned earlier, when data is sent in segments, it can get scrambled and sometimes lost en route to its final destination. If this happens, a TCP connection retries sending the data to complete the data delivery. TCP is used by nearly every major service, such as Web surfing, file transfers, and email.

Application Layer

The final and often most difficult network layer to protect is the Application layer. The problem with securing the Application layer is that TCP/IP-enabled applications are able to perform in any fashion they want and typically different applications are programmed differently, greatly increasing the potential problems in securing this layer. The Application layer can best be described as where the actual

software interacts with the network. Applications that use this layer include Web browsers, File Transfer Protocol (FTP) clients, games, and similar software.

As mentioned, securing the Application layer is difficult because a wide variety of configurations, as well as ports and protocols, are being used. For instance, a single application could make use of any number and combination of TCP or UDP plus the 65,536 ports for each protocol, leaving you any number of the possible 131,000 ports to which you could restrict access. As you can imagine, this process would be very counterproductive if it were attempted on an application-by-application process. Instead, it is far more logical and practical to simply allow only a few select applications to send data through your network (using a firewall or similar technology) rather than to stop each application one by one.

The TCP/IP DARPA Network Model

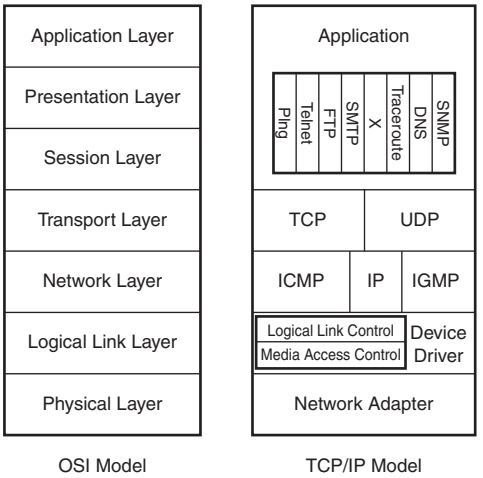
The TCP/IP DARPA network model is fundamentally the same as the OSI network reference model, except that it has fewer layers (only four) and some of the OSI model layers are combined into one TCP/IP DARPA model layer. The four main layers of the TCP/IP network model, starting at the topmost layer, are as follows:

- Applications
- TCP & UDP (often referred to as the IP stack)
- IP & ICMP
- Data Link & Physical

As you can see, there isn't much in the way of core differences between the way OSI and TCP/IP models work. They contain the same information, but the TCP/IP model is much simpler and groups layers together in a more logical method.

Some people, including experts, sometimes interpret the associations in different ways. Many opinions differ on exactly where the TCP and UDP layer matches with the OSI model. Sometimes, it's considered to correlate to the Session layer; sometimes it's considered to match up with the Transport layer. Neither answer is wrong; it's an entirely personal consideration. Figure 2.5 shows the relationship between OSI and TCP/IP models.

FIGURE 2.5
A more detailed comparison of the TCP/IP and OSI network models.



One of the benefits of seeing how the TCP/IP model (often called the TCP/IP *stack* for its layers) compares with the OSI model is that if someone references the TCP layer of the TCP/IP stack, you can relate that to one of the layers of the OSI model to gauge where it resides in the scheme of things. Because no real functional difference exists between the number and names of the layers and how they are grouped, there is no need to go into more depth on the TCP/IP model. Suffice it to say that it's important to understand both models.

IP Protocols and Services

The Application layer can be the most difficult layer to secure against hackers because you are at the mercy of different operating systems and applications. Add to this mix specialized software written for you by a software developer and you have the propensity for long hours researching and securing the software on your systems and servers. The next few sections outline a few of the major concerns at the Application level—mainly Internet services that your users might use and of which hackers might try to take advantage. Although the following information represents by no means the definitive works on these topics and is not the complete listing of potential points of contention, it constitutes a list of the major targets of hackers that you will encounter.

NOTE

Internet Resources Many resources are available on the Internet to gain information on security vulnerabilities—not only for software, but also for services. To find more information on the topics covered in this section of the chapter, check out the Web sites in the "Suggested Readings and Resources" section of this chapter.

Simple Mail Transfer Protocol

Although Simple Mail Transfer Protocol (SMTP) itself is not vulnerable, the services that use it, namely email servers, are targets for attack by hackers. SMTP servers can be particularly useful to hackers if they are spamming other users, servers, or mailing lists with messages and they want to hide their actual location, often called *spamming* or *Email Relay* attacks. By routing messages through a compromised server, a hacker can effectively make the messages look like they were coming from the compromised server, not the hacker. This activity is quite common when a hacker is trying to release an email virus and doesn't want to be found out. Another common attack against email servers using SMTP is to deluge the server with email in a form of a Denial-of-Service (DoS) attack. By sending more email than the server can handle, the hacker can effectively flood the server, causing it to either shut down or back up so badly that it cannot process legitimate email being sent and received by users on the server. The only real protection against these types of attacks is to keep your email services patched with the most up-to-date security fixes. You can also ensure that message routing is not available to servers other than those on your network, and ensure that you have a robust and up-to-date antivirus software package on your server and client systems.

When it comes to protecting against viruses, above and beyond a good antivirus software package, your next best defense is user education. By spending a bit of time educating your users as to what are potentially dangerous email attachments, you can help minimize the possibility of infection.

File Transfer Protocol

File Transfer Protocol (FTP) is another TCP/IP service that rarely, if ever, sees attacks directly levied against it. FTP, like SMTP, finds itself the unwilling accessory to hackers attempting to infiltrate your servers. The main benefit of FTP to a hacker is that it provides a method of placing files on the server for later use. For instance, if a hacker wants to place a Trojan horse on your system and they can obtain access to your server through FTP, they may be able to place the file on your server and have the Trojan activated remotely.

This situation typically results from a hacker gaining access to the system by guessing or cracking the password of a user account.

Another factor has to take place for this to be effective: The hacker must have some way of remotely executing the Trojan. In most cases, a Trojan needs to be manually run (quite often, this needs to be done on the system itself or, if possible, remotely). To execute the Trojan remotely, the hacker usually needs to be able to execute the program by connecting through the Web server on the server itself (assuming it has one). This is typically feasible only if the Web server isn't patched or secured properly. Yet, there are many exploits available against Web services, such as Internet Information Services version 4 and 5 that are usually installed by default on Microsoft Windows NT 4 and 2000 servers. The real problem is that when the services are installed, default FTP and Web servers are created that are both active and entirely unsecured. If the administrator isn't aware of this fact or neglects to disable or remove the services, secure them, or at minimum remove the default Web and FTP sites, he is allowing potential hackers a huge amount of possibilities.

As you can imagine, the best way to defend against these types of situations is to ensure that you have either removed the default services or sites if you aren't using them, or if you are using them, to secure them according to the guidelines from the software vendor. Another very important aspect of security in regard to FTP is that you should always keep your default FTP folders and data on an entirely separate partition or hard drive from your operating system. This is mainly to help limit the possibilities of a hacker leaving the default folders and possibly running amok in sensitive areas. When in these areas, a hacker could possibly manipulate or delete the log files that show any activities the hacker attempts or take that opportunity to place more Trojans or back doors into the server.

Finally, it is important to make sure that you have properly secured all the folders and directories on your server according to your needs. For instance, not everyone needs access to your log files directory. Only administrators need to have access to these types of files; secure the folders containing the log files as necessary.

IN THE FIELD

FTP SECURITY ISSUES

FTP can be a huge hole in any security implementation if not configured properly. Be sure to disable FTP if it is not absolutely needed and, if it is needed, you may want to consider having a dedicated system that is well secured to host this service.

You can make an FTP server more secure by doing two things. First, arrange for the data to be stored on a separate drive and partition from your operating system and limit access to the operating system drive. Second, use only anonymous logons and limit their access to viewing and downloading only, unless there is a specific need for that user to have uploading rights.

Hypertext Transfer Protocol

The most highly used protocol on the Internet today is Hypertext Transfer Protocol (HTTP). Web browsers (and on occasion other Internet applications, such as file-sharing programs) use HTTP to communicate and transfer data to and from the client. Although there aren't any direct attacks against HTTP itself, there are security issues in relation to the client software using HTTP and external programs that are used by HTTP (Web) servers, such as printing over the Web or online purchasing databases. Among the highest concerns to Web browsers is the downloading of ActiveX or Java applets that could contain malicious programming or scripts designed to exploit a security weakness in the Web browser itself or some other program used by the browser, such as a video player or the operating system. This method could also be used to install other malicious code on a user's system, such as a Trojan horse, a back-door utility, or other virus that can be used to compromise a system. The only real defense against these types of attacks is similar to protecting SMTP-related services—use an up-to-date antivirus software solution and educate your users as to appropriate and inappropriate downloads.

The potential security issues with Web servers can be many—especially with Microsoft IIS versions 4 and 5, both of which have had a host of security vulnerabilities associated with them. Although a few problems have been reported with Linux-based Web servers, they are certainly nowhere near the problems faced by users of IIS. Many of the problems associated with Web servers and IIS in general involve unnecessary services left active, which hackers can exploit. Many times, these services either have a vulnerability associated with them or they simply are not secured or configured properly, allowing hackers an even easier task of accessing a Web server. Often, the fix for many problems (too numerous to mention here) is to disable IIS or Web services if you are not using them. You should also patch and secure your HTTP-related services with the latest fixes and patches from the appropriate vendor. In addition to patching, if you are using Web services on your servers, disable any features or options of which you are not taking advantage, and be sure you secure access to your log files for your Web server. Many Web services, such as IIS, allow you to place log files in a different, more secure, location than the default. Hackers know the default location for log files, which makes it that much easier to modify or delete the log files.

WARNING

IIS Attacks Attacks against Microsoft IIS servers are constantly on the rise. Before you expose any Microsoft servers to the Internet, make sure that you either uninstall or disable IIS or spend a fair amount of time securing it. If left unsecured, an IIS server can be the victim of an attack or intrusion within hours!

Because of the growing number of issues with Web services in general, your best plan of attack is to arm yourself with information. Peruse the vendors' Web sites and third-party security Web sites for vulnerability and security-configuration information for your product. It can mean the difference between having secure Web services or being hacked in very little time. For more information on securing Microsoft IIS version 5, visit `http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools.asp`. For information on securing other Microsoft products, go to the Microsoft security-related Web site at `http://www.microsoft.com/security`.

User Datagram Protocol

The User Datagram Protocol (UDP) is termed as a connectionless protocol because it is used only for broadcast-type traffic that doesn't require an authentication process to establish a connection. UDP is

typically a much faster protocol than TCP mainly because it lacks the consistent exchange of connection information, but the downfall is a general reliability issue because the connection is not maintained by the systems. In addition, if during the transfer of data a packet is missing or data is corrupted, UDP does not try to transfer the data again as TCP does. This is why UDP is quite commonly used for video and audio applications (such as an Internet radio station), in which a few lost packets of data isn't problematic.

There is only a handful of possible attacks against UDP because the protocol isn't expecting a response to the data it is sending out; it is hard for hackers to plant malicious data in a reply packet to crash or intercept a connection with the server. Typically, there is little to worry about with UDP and security, but, as always, keep an eye out for new and up-to-date information in case something is found.

How Hackers Exploit TCP/IP

Because hackers are constantly scanning IP addresses on the Internet, you could have a system hacked within a matter of minutes of connecting to the Internet. Before you consider installing a server to the Internet, you must secure the services and apply the latest patches and service packs. If a hacker happens to see your newly built server on the Internet, he will try to leverage a laundry list of exploits, scripts, and prebuilt intrusion utilities against it in hopes of finding a hole. Unless you have spent time securing and patching the server, the hacker will likely penetrate your server in short order.

Other possibilities include substituting fake packets of information for others or executing attacks, such as a Denial-of-Service attack, by flooding the target system with an overwhelming amount of IP packets, causing the system to either crash or become unavailable for the duration of the attack. Some of the more complicated attacks, such as substitution of packet data, are far more complex and the typical hacker does not go to such great lengths or simply lacks the skill to orchestrate such an attack. More often, the typical hacker resorts to commonly available attacks to get the job done. However, if the hacker wants to attack or penetrate the target, he typically does whatever is necessary to get in.

IN THE FIELD

HACKING REALITIES

Many hackers are just people interested in playing around and learning the intricacies of how things work. Popular opinion has hackers always looking for new targets and being malicious in their activities; however, that is not always the case.

These types of casual hackers are not good enough to pull off the more difficult hacks. In many cases, if things are even a bit difficult, those who might be casually hacking will likely look elsewhere rather than throw themselves at a problem, especially if there appears to be the slightest risk of being caught for no real gain.

REVIEW BREAK

- Hackers use long-standing flaws in the fundamental design of a TCP/IP packet to help attack systems and servers.
- When disconnecting connections between a client and server system, the port used to connect the systems can sometimes hang open—consuming resources. Some attacks by hackers focus on consuming all available resources by creating connections and then not properly closing them.
- There are seven layers to the OSI network model (Physical, Data Link, Network, Transport, Session, Presentation, and Application). Each has its own security issues, although some are more vulnerable than others—in particular, the Application layer.
- The TCP/IP network model (also known as the DARPA model) has only four layers, but each of these layers corresponds with a layer or combination of layers in the OSI network model. The four layers of the TCP/IP network model are Data Link & Physical, IP & ICMP, TCP & UDP, and Application.
- It is important to spend time researching possible vulnerabilities in your network services, such as HTTP, FTP, and SMTP, because there are numerous issues (in particular, HTTP) that could cause security issues on your servers and network.

NETWORK-LEVEL TOPICS

- Identify the basic security issues associated with system/network design and configuration.

By understanding key TCP/IP networking concepts, you can better understand how to secure your networks from intrusion or snooping from hackers or even malicious users on your network. Some key networking technology you will learn about in the next two sections, "Network Address Translation (NAT)" and "Secure Sockets Layer (SSL)," can greatly help in securing your network traffic.

Packet Routing Basics

In previous sections, you have learned about the basic anatomy of an IP packet and how TCP/IP is modeled for use on networks, such as the Internet. In this section, you learn how packets are routed around a network. On its deepest level, routing packets on the Internet is a very complex topic, but to gain a fundamental view of how it works, take a look at Figure 2.6.

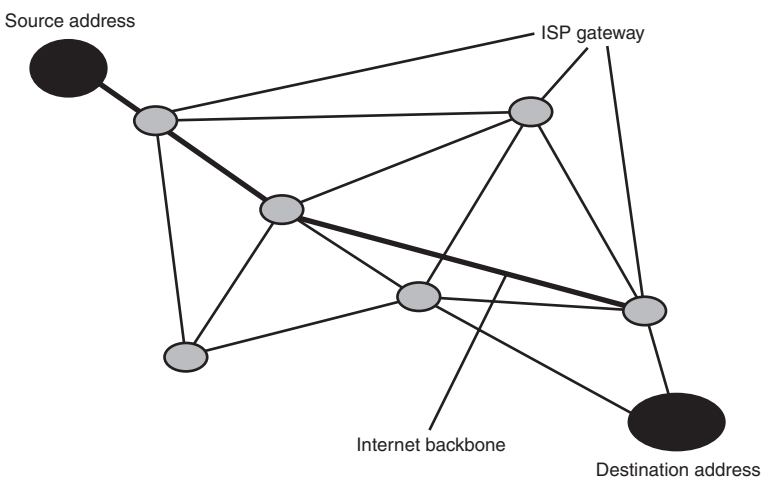
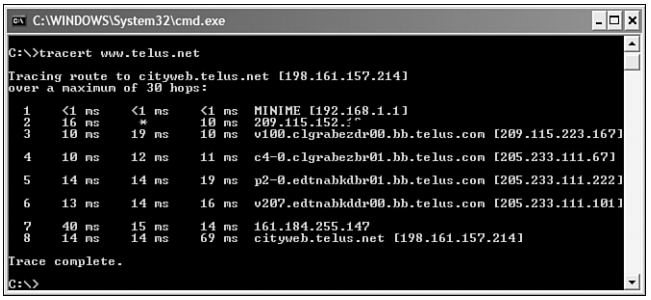


FIGURE 2.6
Basic Internet network topology.

As you can see, it is a large web of smaller interconnected networks (Internet). A good analogy is to relate it to the body's nervous system. You have the big, main pipe that stretches out to essential areas and then all the midsize networks hook into that larger pipe and then the smaller into the midsize networks and so on.

The great design theory behind a network such as the Internet is that you can theoretically get anywhere one way or another, even if a main gateway is blocked or down, but keep in mind that sometimes this isn't the case and you will have points of failure. Usually, you have multiple routes or pathways for your packet so it should make it to its destination; it just may take longer than normal to arrive. The packet knows how to get to its destination on the network using the destination IP address (much like a ZIP Code); it pinpoints an exact spot on the Internet and every router or gateway on the Internet has a routing table of some type or another. Internet service providers (ISPs), or larger carriers, typically manage these routing tables. In many cases, a packet can make several stops along the way called hops as it hits various gateways on the Internet. One way to see where your packets are going is to use a utility such as Tracert in Windows. Figure 2.7 shows a sample output of Tracert.

FIGURE 2.7
Output of Tracert following a packet over the Internet.



One Internet routing fact is that when a packet is traveling over the Internet, gateways on the Internet don't pay attention to the actual source address of the packet, just the destination. This one particular design feature allows many common attacks that hackers employ, such as Denial of Service and IP spoofing. These attacks often allow hackers to carry out their attacks without the victim knowing where exactly the hacker is coming from.

System Ports

You have undoubtedly heard of the network term *ports*, which only the TCP and UDP protocols make use of on a system. Ports are the result of a single system's need to communicate with multiple services and protocols on a network connection, such as the Internet. A classic example is a server that provides email services and an FTP site for users to upload and download files. For the most part, any major program type or service has its own default-specified port (for instance, FTP uses ports 20 and 21, whereas HTTP uses port 80).

Each packet of data coming in has a destination port listed inside the header information; this port number indicates where exactly on the server the packet is destined. On a system, there are 65,536 ports for use by TCP and UDP, but only the first 1,023 have been marked as well-known ports that will be used by common services. Understanding which ports your system has active can help provide you with better insight into the possible ports hackers might use to gain access to your system. Remember, if the port isn't in use (much like the physical network jack earlier), don't have it open; it simply provides another avenue into your system. Another tactic to help prevent attack through known ports and services in use is to change the port the service is using. For instance, change your FTP server from using port 21 to using port 2100. This works well, but you must also reconfigure the port information for any clients connecting to that server.

Network Address Translation

Network Address Translations (NAT) is a popular way to share a common Internet IP address to any systems inside your network. NAT basically takes the external valid IP address provided by your Internet service provider (ISP) and then passes the information to your internal invalid IP address on your system using NAT services. An invalid address is one that does not work on the Internet, but works fine on internal networks, such as those within the 192.168.x.x range. When a system on the internal network requests something on the Internet, such as a Web page, it sends the request through the NAT service. NAT assigns a particular port (for instance, 3003) that the system's communications pass through and records that data in a table within the NAT service. NAT then picks up the packet and stamps its own valid IP address on it that it uses to connect to the Internet. When the Web site from which you are requesting data receives the packet, it completes the request and then sends it back to the NAT server because that is the IP address that was on the packet. The NAT server then takes the data out of the packet, reads the return address port that was marked in the table, and routes it to the appropriate system on the internal network. For example, a client system sends a packet out to a Web server requesting a Web page. The packet has the client's internal network address (for example, 192.168.1.10) and port 3003. The destination is the Web

EXAM TIP

NAT Fundamentals The exam will likely cover a few questions on Network Address Translation. Understanding the fundamentals of NAT provides useful information for this section of the exam, as well as for questions on other topics, such as firewalls.

NOTE

Out-of-the-Box NAT Services A NAT service is available out of the box with Windows 2000/.NET Server and many Linux distributions. If your company lacks any other protection from the Internet, this can be a good and low-cost alternative to costly firewall or proxy software. By using the security by obscurity rule, you can help protect your workstations connected to the Internet by making the server act as a middleman. Just make sure if you go this route to secure your server very well and look into a low-cost firewall or similar software package.

server's address (198.167.0.9) and port 80, which is received by the NAT service. The NAT service then chooses a free port number, such as 5008, on its system and adds an entry into the NAT table that associates any incoming packets to its own port 5008 to the client address 192.168.1.10 and port 3003. NAT then replaces the source address and packet on the packet with its own and routes the packet out to the Internet. The Web server eventually gets the packet and responds with the appropriate information, sending the packets back to the NAT server on port 5008. The NAT server receives that information and replaces the original client system addresses and port information with the client's and then returns the packet.

Secure Sockets Layer

A popular way to secure traffic passed to and from Web servers and Web browsers is to use Secure Sockets Layer (SSL) security on Web pages and servers. SSL uses encryption to encrypt the data passing from the Web page or directory (for instance, a directory that holds an online store's customer database) to the client (Web browser), allowing for secure data transmission. Because it requires additional processor and network bandwidth because of the encryption and decryption of packets, SSL is generally reserved for those Web pages or sites that really need it. The places in which SSL is actually needed are locations such as an online store, in which you enter your sensitive information (such as your credit card number) to complete a transaction. You want these types of Web pages to be encrypted; however, you do not need to encrypt certain pages using SSL, such as the main page or the news page in which there isn't any sensitive information. Most Web servers support the use of SSL, including Windows 2000 through IIS version 5. Another great feature of SSL is that most, if not all, current Web browsers support SSL. The security options you have learned about in this chapter are certainly not the only ones available to you. They do, however, represent a good cross section of security options of which you can take advantage when trying to secure TCP/IP communications and authentication on your servers and network in general. Likely, you will use a mix of these options to design the best overall security solution for your network based on the potential results, your organization's security needs, and the time it takes to implement and support your new TCP/IP security options.

CASE STUDY: KEY AND TUMBLER SAFES

ESSENCE OF THE CASE

- K & T Safes has brought your consulting company in and they want a general assessment of their network before they even broach bringing clients' data and equipment onto their network. Given their configuration as you know it and from what you have learned to this point in the book, what four basic things would you recommend to secure their network?
- Because K & T Safes will be providing sensitive data storage for their clients onsite and allowing them to connect remotely to access their information, the president of the company wants to know what security implementation is the most beneficial. You need to figure this out based on their current hardware and operating system configuration, and provide the most secure network communications inside their networks and to their clients. In addition, keeping in mind that K & T Safes is providing systems for their clients to remotely connect to the network, what do you recommend for securing their network communications and what would you do to help secure the systems that are provided to their clients?
- The IT administrator at K & T Safes has a few clients requesting that they retain Web servers onsite that they can remotely manage using a remote access client. The administrator was considering using Telnet, which interfaces with their client's management software package. What would you recommend to provide additional security in this situation, and why?

SCENARIO

Key and Tumbler Safes is a small, but fast-growing safe design and security firm, based in an affluent area of the western United States. Many of K & T Safes' clients are well-to-do businesspersons, actors, and large corporations who all rely on K & T Safes' safe-design expertise to build safes for them to use in their homes and businesses, as well as to store valuable documents in their own vault onsite at K & T Safes. K & T Safes is looking into the potential market for expanding into providing secure storage of data on their servers that can be accessed over the Internet and they have called in your consulting company to help guide them down that path.

K & T Safes' network is composed entirely of Windows 2000 systems and servers, all with the default configuration and no additional security or other configuration applied. They currently do not have any additional network security hardware established on their network. K & T Safes will provide to prospective clients systems that are preconfigured specifically for connecting to their resources held onsite.

ANALYSIS

Four main things should be done. First, implement a security plan or checklist to appraise and secure their network plan, systems, and servers. Second, implement internal security to secure their servers and physical equipment. Third, implement a good antivirus security solution. Finally, implement a network security solution, such as a firewall or similar package. In this case, it will work, but for the utmost security,

continues

CASE STUDY: KEY AND TUMBLER SAFES

continued

they should use the SSH client instead of a regular Telnet client to ensure their data is secure when they are managing their servers remotely. This is because Telnet is very insecure and passes data and authentication methods in clear text, which can be intercepted. In the final instance, the easiest solution is to enable SSL to secure their authentication process at a minimum; these servers should also have some type of encryption, such as IPSec, enabled to secure the data being transferred. Placing these servers on a separate network from the rest of the servers inside the company is also a good idea to isolate them from other sensitive data.

CHAPTER SUMMARY

KEY TERMS

- datagram
- Initial Sequence Number (ISN)
- Internet Control Message Protocol (ICMP)
- IP header
- Tribal Flood Network (TFN)
- Winnuke

There are many issues related to TCP/IP security—among the most prominent is that any system with TCP/IP installed and configured and attached to the Internet becomes a target very quickly. If this system is not secured properly, it has a high risk of being found and compromised by a hacker performing scans of IP addresses for vulnerabilities. It is usually recommended to apply the latest patches and service packs and take any vendor-recommended steps to secure the product in question. Also, disabling any unused services, ports, and software can be a good way to augment a system's security.

All address information directly relating to a packet is contained in the header for a packet of information. This header includes such information as a source IP address, a destination IP address, and contents of the packet. Each header also includes a sequence number, which is applied to each packet that identifies the order in which the packets are to be sent and then reassembled at the receiving computer. Often, packets arrive out of order or some won't make it to the destination at all; by having these sequence numbers, the system can reorganize the information into usable data, or request the missing packet(s) of information from the source as needed.

Hackers can use several methods to take advantage of the header and sequence numbers of a packet, among these attacks the most prominent are Denial-of-Service attacks, packet sniffing, and IP spoofing. These attacks can cause a disruption of service, provide valuable information to a hacker, or hide the source of an attack.

CHAPTER SUMMARY

The OSI network model has seven layers, each with its own security considerations. The more common layers to be concerned about are the Physical, Network, Transport, and Application layers. Among these layers, the Application layer is by far the hardest to secure because it is the layer on which software operates, requiring you to ensure that not only the software used by the server, but also the client, is secure to provide the best possible security.

The TCP/IP network model is based on the OSI model and follows the same basic orientation, except there are only four layers rather than seven as in the OSI model. Some layers that are related to each other are combined into one layer for the sake of convenience. Although both models represent the same thing, they are both very worthwhile to keep in mind as you work with TCP/IP security. The four layers of the TCP/IP network model are Applications, TCP & UDP, IP & ICMP, and Data Link & Physical.

You should keep in mind several issues when considering Application layer security, including the major Internet services that you make use of on your servers. Services such as SMTP (email), FTP, and HTTP (Web) all have their own security issues that, although not directly related to the protocols themselves, are more geared toward the server or client software used to provide them. Ensuring that the appropriate steps required to secure them are taken is the best plan of action, and as with any other system, these services should either be secured or disabled entirely before placing the server on the Internet.

Several methods are available for securing TCP/IP authentication and data being sent between connections. Among the most common and reliable choice is IPSec, which is included in some operating systems and is also available from third-party solution providers. SSL can be used to secure Web page transactions and viewing. Other security technologies, such as public-key certificate services, can be used to secure the authentication process, or a combination of all different methods can be used to enhance your entire security scheme.

APPLY YOUR KNOWLEDGE

Exercises

2.1 Attempting an ICMP Attack

In this project, you use a Winnuke program to send an ICMP attack against another system. Any system you attack with Winnuke will likely repel the attack (especially any operating system beyond Windows 95). Regardless, if the attack succeeds, this project shows you how quick and easy it is to use a Winnuke attack against a system.

Estimated Time: 15 minutes

1. Start your computer and log on, if necessary.
2. Install Winnuke GUI on your Windows 2000 system. Note that you can also obtain Winnuke as a script for Linux/Unix, or for DOS/Windows.
3. Run Winnuke. A GUI appears asking you to insert an IP address and enter a message to display on the screen when the nuke is successful. Type the IP address for a Windows system that can be attacked safely.
4. Click the NUKE button. After a few seconds, either Winnuke reports a successful nuke or it reports "nuke failed" or "couldn't connect," if the system being attacked is either patched or was an unrecognized operating system.
5. If the nuke was successful, the system attacked should show the attack message you entered in a pop-up box, and the system should display either a General Protection error or be entirely unstable and crash after a bit of usage.
6. Close Winnuke.

2.2 Viewing Active TCP and UDP Connections

In this project, you learn how to see currently active UDP and TCP connections on a system using Netstat, a simple utility included with Windows (any version). Netstat can provide you with information about the network connection(s) in your computer, including open ports, the amount of data transferred, and more. This information can help you see what ports are open on your system that may have to be closed.

Estimated Time: 10 minutes

1. To open a command-prompt window, click Start, click Run, type **`cmd`** in the Open text box, and then click OK.
2. On the command line, type **`netstat -a`**, and then press Enter. Running the `netstat` utility with the `-a` switch tells *netstat* to display all active connections and open ports on the system. A sample output of this command is shown in Figure 2.8. This particular output is from a Windows 2000 Server system with Exchange 2000 installed and running.

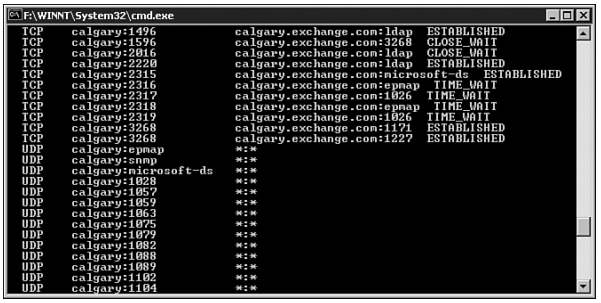


FIGURE 2.8
Output of the `netstat -a` command.

APPLY YOUR KNOWLEDGE

> **NOTE**
>
> The first column indicates the protocol (TCP or UDP), the next signifies the name of the system (the port number for this particular connection or port comes after the colon), and the final column generally gives connection information including whether it is established or closed.

3. After you finish viewing the output of this command, close the command-prompt window by typing **`exit`** and then pressing Enter.

2.3 Creating an IPSec Policy

In this project, you learn how to create a quick and simple IPSec security policy in Windows 2000. Windows 2000 allows for an easily manageable way to provide stronger IP security right out of the box, without the use of additional software. Even the basic IPSec included in Windows 2000 is flexible enough to allow you to create simple or even very specific IPSec rules and security settings for you to use on your network.

Estimated Time: 20 minutes

1. To open the Local Security Settings console, click Start, point to Programs, point to Administrative Tools, and then click Local Security Policy. The console is shown in Figure 2.9. In this console, you can configure various security-related settings for Windows 2000, but the option you focus on in this project is the IP Security Policies on Local Machine.

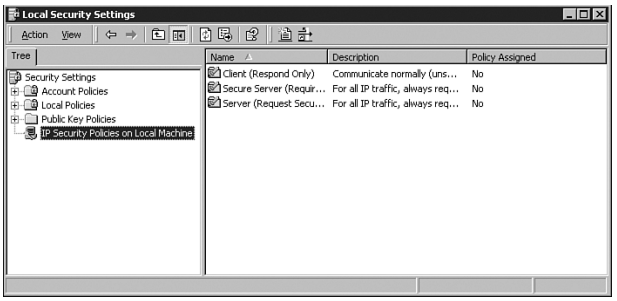


FIGURE 2.9
Local Security Settings console.

2. Right-click IP Security Policies on Local Machine in the Tree pane, and then click Create IP Security Policy to open the IP Security Policy Wizard.
3. In the Welcome to the IP Security Policy wizard window, click Next to start the wizard. The IP Security Policy Name dialog box opens.
4. In the Name text box, delete the default name and type **`Test Security`**. In the Description text box, type **`"Testing the security policy wizard"`**. Click Next. The Requests for Secure Communication dialog box opens.
5. This dialog box allows you to specify that the default response rule is enabled, which allows IPSec to respond to any requests for a secure communication when using this policy. Verify that there is a check mark next to the <Dynamic> option (as shown in Figure 2.10), and then click Next. The Default Response Rule Authentication Method dialog box opens.

APPLY YOUR KNOWLEDGE

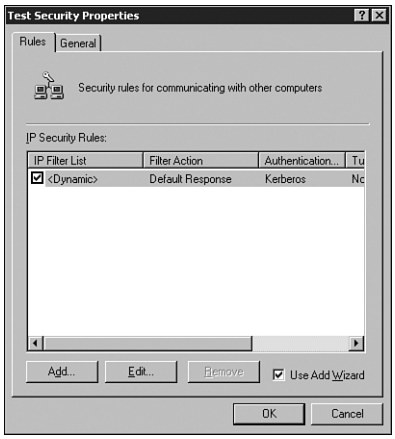


FIGURE 2.10
Test Security Properties dialog box.

6. You must specify the type of security rule you want to use for this policy. If the only systems you will connect to are Windows 2000 systems, use the Windows 2000 default (Kerberos V5 protocol) option. If you connect to other operating systems, you should use a certificate from a certificate authority, or a key that you type in yourself. For this project, verify that Windows 2000 default (Kerberos V5 protocol) is checked, and then click Next. The Complete the IP Security Policy Wizard dialog box opens.
7. It states you have successfully created a new policy and gives you the option to edit the properties of the policy after the wizard is complete. Verify that Edit Properties is checked, and then click Finish to complete the wizard.

Review Questions

1. After data has passed through the various layers of the TCP/IP model and all the header and frame information has been added to the packet, the result is called a _____.
2. What two popular attacks can be used to flood ICMP data in the hopes to crash the targeted server?
3. How could having your FTP server configured to allow uploads on the same partition as your operating system lead to potential problems with intrusion and other attacks?
4. To secure data passed between Web browsers and Web servers, you can use Secure Sockets Layer (SSL) to improve data security. What technology is used when implementing SSL to secure this data?
5. What is the main benefit of having a negotiated TCP/IP connection as you would with TCP rather than an unnegotiated connection using UDP?

Exam Questions

1. In 1985, Bob Morris noticed problems in the way TCP/IP handshakes took place when systems were connecting to one another. These problems could lead to hackers doing what?

   A. Sniffing a network connection

   B. Forcing a network connection

   C. Decrypting a packet

   D. Crashing a server

APPLY YOUR KNOWLEDGE

2. From the following selections, which two are possible causes for the lack of a single one-size-fits-all security model for every server and system?

   A. Companies have their own specific security needs—some more, some less.

   B. Standard security models are ineffective and unreliable.

   C. The operating system software on servers and clients differs from organization to organization.

   D. No one has developed a single, comprehensive model.

3. The address information in the IP header can be very useful, but it adds some additional size to a packet. What size is an IP header?

   A. 32 bits

   B. 32 bytes

   C. 20 bits

   D. 20 bytes

4. Because of the way TCP/IP sends data across a network, a hacker can sometimes take advantage of TCP/IP data. Which of the following two options could cause vulnerabilities?

   A. The IP header in a packet can be modified, and there is no way to know whether it has been modified.

   B. TCP/IP transmissions are broken into segments and can arrive out of order, allowing for rogue data to be added.

   C. TCP/IP has been available for a long time and as such isn't secure.

   D. TCP/IP packets are rather large in size, which allows hackers to add a small amount of extra data without it being noticed.

5. Which layer of the OSI model is considered to be the base, or lowest level, of the model?

   A. Session

   B. Presentation

   C. Data Link

   D. Physical

6. Which command is used to end a TCP/IP handshake?

   A. SYN

   B. ACK

   C. FYN

   D. FIN

7. Which layer of the OSI network model controls the flow of data between systems?

   A. Physical

   B. Transport

   C. Network

   D. Application

8. Which layer of the OSI network model is considered the most difficult layer to secure?

   A. Data Link

   B. Transport

   C. Application

   D. Physical

APPLY YOUR KNOWLEDGE

9. What is a major difference between the UDP and TCP protocols?

   A. UDP is less secure than TCP.

   B. TCP allows for missing data to be re-sent.

   C. TCP is faster than UDP.

   D. UDP allows for more open ports than TCP.

10. Approximately how many ports are available on any given system for TCP connections?

    A. 65,000

    B. 6,500

    C. 1,000

    D. Unlimited

11. Which of the TCP/IP network model layers is often called the IP stack?

    A. Data Link & Physical

    B. IP& ICMP

    C. TCP & UDP

    D. Application

12. Of the major TCP/IP network services that can be accessed, which would most likely be used to plant a Trojan horse on a server for a hacker to activate manually?

    A. SMTP

    B. FTP

    C. HTTP

    D. Telnet

13. Keeping Question 12 in mind, what TCP/IP network service would the hacker most likely use to activate that planted Trojan horse?

    A. SMTP

    B. FTP

    C. HTTP

    D. Telnet

14. Which two of the following options are components of a basic IP packet?

    A. IP datagram

    B. IP header

    C. IP footer

    D. IP payload

15. An IP packet has frames added to every packet sent. These frames are used to tell the NIC sending and receiving the packet when to start receiving that packet and when to stop. What are the names used to signify the beginning and end of a frame of an IP packet?

    A. Frame header

    B. Frame payload

    C. Frame footer

    D. Frame trailer

Answers to Review Questions

1. The resultant packet with the additional information is called a TCP Segment. See the section, "Basic TCP/IP Principles."

2. Winnuke and Tribal Flood Network (TFN) are two very popular ICMP attacks that could, if effective enough, crash systems and servers that are not patched to protect against these attacks. Most common operating systems are now

APPLY YOUR KNOWLEDGE

patched to stop these types of attacks from being effective. See the section "Network Layer."

3. A hacker could potentially upload a Trojan or other malicious program and these could be executed remotely via a Web browser or if someone runs the Trojan or malicious program locally by accident. See the section "File Transfer Protocol."
4. Encryption is the technology used behind SSL and most major TCP/IP security solutions. See the section "Secure Sockets Layer."
5. When data is sent in segments, packets could become lost or corrupted on their way to the destination. By having a negotiated connection, TCP/IP can determine whether any packets are missing and rerequest the information from the connection sending the data. See the section "Transport Layer."

Answers to Exam Questions

1. **B**. Bob Morris noticed problems that would allow hackers to force a network connection. See the section "Basic TCP/IP Principles."
2. **A, C**. Possible causes for a lack of a one-size-fits-all security model are companies that have their own specific security needs and that software and configuration differs from organization to organization. See the section "The OSI Network Reference Model."
3. **D**. The additional size added to a packet for the IP header is 20 bytes. See the section "Structure of an IP Datagram."
4. **A, B**. Two circumstances in which vulnerabilities could exist are when the header in a packet can be modified and when a packet arrives out of order, which allows for possible rogue packets to be introduced into the packet stream. See the section "Structure of an IP Datagram."
5. **D**. The Physical layer is considered to be the base level of the OSI model because it represents the layer that actually connects networks together, upon which the other layers are built. See the section "The OSI Network Reference Model."
6. **D**. The command used to end a TCP/IP handshake is `FIN`. See the section "TCP Handshakes and Headers."
7. **B**. The layer of the OSI model that controls the flow of data between systems is the Transport layer. See the section "The OSI Network Reference Model."
8. **C**. The layer most commonly considered the hardest to secure is the Application layer because of the wide variance in software and configuration. See the section "The OSI Network Reference Model."
9. **B**. The main difference between TCP and UDP. TCP allows for missing data to be re-sent. See the section "Transport Layer."
10. **A**. There are typically 65,000 or more ports available for TCP connections on any system. See the section "System Ports."
11. **C**. The layer that is commonly called the IP stack is the TCP & UDP layer in the TCP/IP network model. See the section "The TCP/IP DARPA Network Model."
12. **B**. The most likely network service that would be used to plan Trojan horse programs is FTP because of its simplicity and generally direct access to the server's hard drive and directories if not set up properly. See the section "File Transfer Protocol."

APPLY YOUR KNOWLEDGE

13. **C**. Once planted on the server, a hacker would likely use HTTP to activate the Trojan because it can be used to navigate directories and remotely execute programs. See the section "Hypertext Transfer Protocol."

14. **B, D**. Both the IP header and IP footer are components of a basic core IP packet. See the section "Structure of an IP Datagram."

15. **A, D**.The name of the beginning of an IP packet is a frame header, and the name of the end of an IP packet is a frame trailer. See the section "Structure of an IP Datagram."

Suggested Readings and Resources

1. `www.microsoft.com/security/`—General Microsoft security Web site
2. `http://www.microsoft.com/ntserver/techresources/commnet/TCPIP/TCPIntrowp.asp`—A general introduction to TCP/IP by Microsoft
3. `http://www.microsoft.com/mind/1098/tcpip/tcpip.asp/inthisissuecolumns1098.htm`—The ABCs of TCP/IP from the MSDN