Chapter 9

Computer Security Technology

Chapter Objectives

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

- Evaluate the effectiveness of a scanner based on how it works
- Choose the best type of firewall for a given organization
- Understand antispyware methods
- Employ intrusion detection systems to detect problems on your system
- Understand honey pots

Introduction

Throughout this book, various aspects of computer security have been discussed. At this point in your studies, you should have a good idea of what the real dangers are and what adequate security measures include, as well as a basic understanding of the various forms of computer attacks. However, if you are striving to secure your network, you will need more technical details on the various security devices and software you might choose to employ. This chapter reviews these items with enough detail to allow you to make intelligent decisions on which types of products you will see.

Most of these devices have been mentioned and briefly described in the preceding chapters. The intent of this chapter is to delve more deeply into details of how these devices work. This information is of particular value to those readers who intend to eventually enter the computer security profession. Simply having a theoretical knowledge of computer security is inadequate. You must have some practical skills. This chapter will be a good starting point for gaining those skills, and the exercises at the end of the chapter will give you a chance to practice setting up and evaluating various types of firewalls, intrusion detection systems (IDSs), and antivirus applications.

Virus Scanners

A *virus scanner* is essentially software that tries to prevent a virus from infecting your system. This fact is probably abundantly obvious to most readers. Knowing how a virus scanner works, however, is another matter. This topic was discussed briefly in our previous discussions on viruses but will be elaborated on in this chapter.

In general, virus scanners work in two ways. The first method is that they contain a list of all known virus definitions. The virus definitions are simply files that list known viruses, their file size, properties, and behavior. Generally, one of the services that vendors of virus scanners provide is a periodic update of this file. This list is typically in a small file, often called a *.dat* file (short for data). When you update your virus definitions, what actually occurs is that your current file is replaced by the more recent one on the vendor's website.

The antivirus program can then scan your PC, network, and incoming email for known virus files. Any file on your PC or attached to an email is compared to the virus definition file to see whether there are matches. With emails, this can be done by looking for specific subject lines and content. The virus definitions often also include details on the file, file size, and more. This provides a complete signature of the virus.

The second way a virus scanner can work is to look for virus-like behavior. Essentially, the scanner is looking to see if the file in question is doing things that viruses typically do—things like manipulating the Registry or looking through your address book. Obviously, this second technique is essentially a best guess.

How Does a Virus Scanner Work?

Let's take a more detailed look at how antivirus software works. An article in the July 2004 issue of *Scientific American* titled "How Does a Virus Scanner Work," stated that a virus scanner is essentially software that searches for the signature or pattern of known virus. Keep in mind that the scanner only works if you keep it updated. And, of course, it only works with known viruses. While that article may seem a bit dated now, it is still accurate.

Recall that the second way a virus scanner works is to watch for certain types of behaviors that are typical of a virus. This might include any program that attempts to write to your hard drive's boot sector, change system files, automate your email software, or self-multiply. Programs that attempt to modify the system Registry (for Windows systems) or alter any system settings may also be indicative of a virus.

Another feature that virus scanners search for is a file that will stay in memory after it executes. This is called a *Terminate and Stay Resident (TSR)* program. Some legitimate programs do this, but it is often a sign of a virus. Additionally, some virus scanners use more sophisticated methods, such as scanning your system files and monitoring any program that attempts to modify those files.

Whatever the behavior, antivirus software uses specific algorithms to evaluate the likelihood that a given file is actually a virus. It should be noted that modern virus scanners scan for all forms of malware, including Trojan horses, spyware, and viruses.

There is a third method, called heuristic scanning. This is basically examining the file itself and is similar to signature scanning. However, in this case the file need not exactly match the signature. Heuristics refers to functions that rank various alternatives using a branching step in the algorithm. So the Heuristic scan checks to see the likelihood of a given file being a virus. This is based on file characteristics rather than behavior.

It is important to differentiate between on-demand virus scanning and ongoing scanners. An *ongoing virus scanner* runs in the background and is constantly checking your PC for any sign of a virus. *On-demand virus scanners* run only when you launch them. Many modern antivirus scanners offer both options.

Keep in mind that any antivirus program will have some false positives and some false negatives. A false positive occurs when the virus scanner detects a given file as a virus, when in fact it is not. For example, a legitimate program may edit a Registry key or interact with your email address book. A false negative occurs when a virus is falsely believed to be a legitimate program.

Due to false positives, it is recommended that you do not set your antivirus to automatically delete suspected viruses. Rather, they should be quarantined and the computer user notified.

Virus-Scanning Techniques

In general, there are five ways a virus scanner might scan for virus infections. Some of these were mentioned in the previous section, but they are outlined and defined here:

- **Email and attachment scanning:** Since the primary propagation method for a virus is email, email and attachment scanning is the most important function of any virus scanner. Some virus scanners actually examine your email on the email server before downloading it to your machine. Other virus scanners work by scanning your emails and attachments on your computer before passing it to your email program. In either case, the email and its attachments should be scanned prior to your having any chance to open them and release the virus on your system.
- **Download scanning:** Anytime you download anything from the Internet, either via a web link or through some FTP program, there is a chance you might download an infected file. Download scanning works much like email and attachment scanning but does so on files you select for downloading.
- **File scanning:** This is the type of scanning in which files on your system are checked to see whether they match any known virus. This sort of scanning is generally done on an on-demand basis instead of an ongoing basis. It is a good idea to schedule your virus scanner to do a complete scan of the system periodically. I recommend a weekly scan, preferably at a time when no one is likely to be using the computer.
- **Heuristic scanning:** This was briefly mentioned in the previous section. Perhaps the most advanced form of virus scanning, this uses rules to determine whether a file or program is behaving like a virus and is one of the best ways to find a virus that is not a known virus. A new virus will not be on a virus definition list, so you must examine its behavior to determine

whether it is a virus. However, this process is not foolproof. Some actual virus infections will be missed, and some nonvirus files might be suspected of being a virus.

- **Sandbox:** Another approach is the sandbox approach. This basically means that you have a separate area, isolated from the operating system, in which a download or attachment is run. Then if it is infected, it won't infect the operating system.

One way to accomplish sandboxing is for the operating system to set aside a protected area of memory to open the suspected file and to monitor its behavior. This is not 100% effective, but it is far safer than simply opening files on your system and hoping there is no infection.

A related concept is called a "sheep dip" machine. This is useful in corporate networks. You set up a system that is identical in configuration to your standard workstations. However, this sheep dip machine is not networked. Suspect files are opened first on this system. Then the system is monitored for a period of time for signs of infection. Once the file has cleared this check, it can then be opened on normal workstations.

A simple way to do this in a home or small office is to set up a virtual machine on your computer and to open suspected attachments or downloads in the virtual machine first. This virtual machine can have virus scanners running on it. Also, you can change the time in the virtual machine in order to detect logic bombs. Allow the suspect file to reside on the VM for a period of time before bringing it to the host computer.

FYI: How Most Commercial Scanners Work

Most commercial virus scanners use multiple methods, including most, if not all, of the methods listed here. Any virus scanner that uses only one scanning modality would be virtually worthless from a practical virus defense perspective. These modalities are how a scanner works regardless of whether it is using a heuristic scan, download scan, email scan, and so on.

- **Active code scanning:** Modern websites frequently embed active codes, such as Java applets and ActiveX. These technologies can provide some stunning visual effects to any website. However, they can also be a vehicle for malicious code. Scanning such objects before they are downloaded to your computer is an essential feature in any quality virus scanner.
- **False positives and false negatives:** Regardless of the type of virus scanner, any antivirus software will occasionally have an error. There are two types of errors that you should be concerned with. It is possible that your antivirus software will mistake a legitimate program for a virus. For example, you might have a program that is supposed to make some adjustment to the Windows Registry or to scan your email address book. Mistaking a legitimate program for a virus is referred to as a *false positive*. It is also possible that your antivirus will fail to recognize a virus. This is referred to as a *false negative*. The best way to minimize false negatives is to keep your antivirus software updated. For false positives, it is recommended that you simply quarantine suspected viruses and not automatically delete them.

Commercial Antivirus Software

There are four brands of antivirus software that virtually dominate the antivirus market today and a number of companies that offer a commercial scanner also offer a free version that does not provide as many features as the commercial product. For example, AVG AntiVirus, available from www.grisoft.com, is a commercial product, but the company also offers the AVG AntiVirus Free Edition. McAfee, Norton, and Kapersky are three very well-known antivirus vendors. All three products are good choices and come with additional options, such as spam filters and personal firewalls. Any of these three products can be purchased for a home machine for about $30 to $60 (depending on options you add). This purchase price includes a one-year subscription to update the virus files so that your antivirus software will recognize all known virus attacks. Organizational licenses are also available to cover an entire network.

Malwarebytes is another popular product that has both free and commercial versions. Of course, there are other antivirus solutions available. Several free virus scanners can easily be found on the Internet. McAfee, Norton, AVG, Malwarebytes, and Kapersky are mentioned here because they are so commonly used, and it is likely that you will encounter them frequently. But that is not to indicate that I am discouraging you from using other systems. I do strongly recommend that you stick with widely used, well-supported antivirus products.

Firewalls

A *firewall* is, in essence, a barrier between two computers or computer systems. The most common place to encounter a firewall is between your network and the outside world. However, firewalls on individual computers and between network segments are also quite common. At a minimum, a firewall will filter incoming packets based on certain parameters such as packet size, source IP address, protocol, and destination port. Linux and Windows (beginning with Windows XP and in all subsequent Windows versions) ship with a simple firewall. For Windows, the firewall in Windows 7 was expanded to handle filtering both inbound and outbound traffic. Windows 8 and Windows 10 have not significantly changed the firewall functionality in Windows. You should turn on and configure your individual computer firewalls in addition to perimeter firewalls.

In an organizational setting, you will want, at a minimum, a dedicated firewall between your network and the outside world. This might be a router that also has built-in firewall capabilities. (Cisco Systems is one company that is well known for high-quality routers and firewalls.) Or, it might be a server that is dedicated solely to running firewall software. Selecting a firewall, however, is an important decision. If you lack the expertise to make that decision, then you should arrange for a consultant to assist you in this respect.

Benefits and Limitation of Firewalls

A firewall, no matter what type you get (types are described in the next section), is basically a tool to block certain traffic. A set of rules determine what traffic to allow in and what traffic to block. Obviously, a firewall is a critical piece of your security strategy. I cannot even conceive of a reason to run

a system without one. However, it is not a panacea for security because it cannot block every attack. For example, a firewall won't stop you from downloading a Trojan horse. It also cannot stop internal attacks. But a firewall can be an excellent way to stop a denial of service (DoS) attack or to prevent a hacker from scanning the internal details of your network.

Firewall Types and Components

There are numerous types of firewalls and variations on those types. But most firewalls can be grouped into one of the following three families of firewalls.

- Packet inspection
- Stateful packet inspection
- Application

The following sections will discuss each of these and assess the advantages and disadvantages of each.

Packet Filtering

Basic packet filtering is the simplest form of firewall. It looks at packets and checks to see if each packet meets the firewall rules. For example, it is common for a packet filtering firewall to ask three questions:

1. Is this packet using a protocol that the firewall allows?
2. Is this packet destined for a port that the firewall allows?
3. Is the packet coming from an IP address that the firewall has not blocked?

Those are three very basic rules. Some packet filter firewalls have additional rules to check. But what is not checked is the preceding packets from that same source. Essentially, each packet is treated as a singular event without reference to the preceding conversation. That makes packet filtering firewalls quite susceptible to some DoS attacks, such as SYN floods.

Stateful Packet Inspection

The Malwarebytes firewall will examine each packet, denying or permitting access based not only on the examination of the current packet, but also on data derived from previous packets in the conversation. This means that the firewall is aware of the context in which a specific packet was sent. This makes these firewalls far less susceptible to ping floods and SYN floods, as well as less susceptible to spoofing. For example, if the firewall detects that the current packet is an ICMP packet and a stream of several thousand packets have been continuously coming from the same source IP, it is clearly a DoS attack and the packets will be blocked.

The SPI firewall can also look at the actual contents of the packet, which allows for some very advanced filtering capabilities. Most high-end firewalls use the stateful packet inspection method; when possible, this is the recommended type of firewall.

Application Gateway

An *application gateway* (also known as *application proxy* or *application-level proxy*) is a program that runs on a firewall. When a client program, such as a web browser, establishes a connection to a destination service, such as a web server, it connects to an application gateway, or proxy. The client then negotiates with the proxy server in order to gain access to the destination service. In effect, the proxy establishes the connection with the destination behind the firewall and acts on behalf of the client, hiding and protecting individual computers on the network behind the firewall. This process actually creates two connections. There is one connection between the client and the proxy server and another connection between the proxy server and the destination.

Once a connection is established, the application gateway makes all decisions about which packets to forward. Since all communication is conducted through the proxy server, computers behind the firewall are protected.

Essentially, an application firewall is one that is used for specific types of applications such as database or web server. It is able to examine the protocol being used (such as HTTP) for any anomalous behavior and block traffic that might get past other types of firewalls. It is common to have an application firewall that also includes stateful packet inspection.

Firewall Configurations

In addition to the various types of firewalls, there are various configuration options. The type of firewall tells you how it will evaluate traffic and hence decide what to allow and not to allow. The configuration gives you an idea of how that firewall is set up in relation to the network it is protecting. Some of the major configurations/implementations for firewalls include the following:

- Network host–based
- Dual-homed host
- Router-based firewall
- Screened host

Each of these is discussed in the following sections.

Network Host–Based

A *network host–based firewall* is a software solution installed on an existing machine with an existing operating system. The most significant concern in using this type of firewall is that no matter how good the firewall solution is, it is contingent upon the underlying operating system. In such a situation, it is absolutely critical that the machine hosting the firewall have a hardened operating system.

Dual-Homed Host

A *dual-homed host* is a firewall running on a server with at least two network interfaces. The server acts as a router between the network and the interfaces to which it is attached. To make this work,

the automatic routing function is disabled, meaning that an IP packet from the Internet is not routed directly to the network. You can choose what packets to route and how to route them. Systems inside and outside the firewall can communicate with the dual-homed host but cannot communicate directly with each other.

Router-Based Firewall

As was previously mentioned, you can implement firewall protection on a router. In larger networks with multiple layers of protection, this is commonly the first layer of protection. Although you can implement various types of firewalls on a router, the most common type used is packet filtering. If you use a broadband connection in your home or small office, you can get a packet-filtering firewall router to replace the basic router provided to you by the broadband company. In recent years, router-based firewalls have become increasingly common and are in fact the most common firewall used today.

Screened Host

A *screened host* is really a combination of firewalls. In this configuration, you use a combination of a bastion host and a screening router. The screening router adds security by allowing you to deny or permit certain traffic from the bastion host. It is the first stop for traffic, which can continue only if the screening router lets it through.

Commercial and Free Firewall Products

There is a variety of commercial firewall products from which you can choose. If all you want is a basic packet-filtering solution, many software vendors offer this. Major antivirus software vendors (including those mentioned previously in this chapter) often offer the firewall software as a bundled option with their antivirus software. Other companies, such as Zone Labs, sell firewall and intrusion detection software (IDS). Zone Labs, for example, offers the ZoneAlarm Security Suite, which provides all the tools for complete Internet security. Major manufacturers of routers and hubs, such as Cisco Systems, also offer firewall products. How much security you need is a difficult question to answer. A bare minimum recommendation is to have a packet-filtering firewall/proxy server between your network and the Internet—but that is a bare minimum. There are also many free firewall applications available. Zone Labs, mentioned earlier for their commercial product, also offers a free download of the ZoneAlarm firewall protection.

Outpost Firewall, available from www.agnitum.com/products/outpost/, is a product designed for the home or small office user. Like the Zone Labs product, it has both a free version and an enhanced commercial version. Information on this product is shown in Figure 9.1. Note that the free version is an older version of the software and does not include many of the enhancements of the commercial version. But it may be sufficient for your needs.

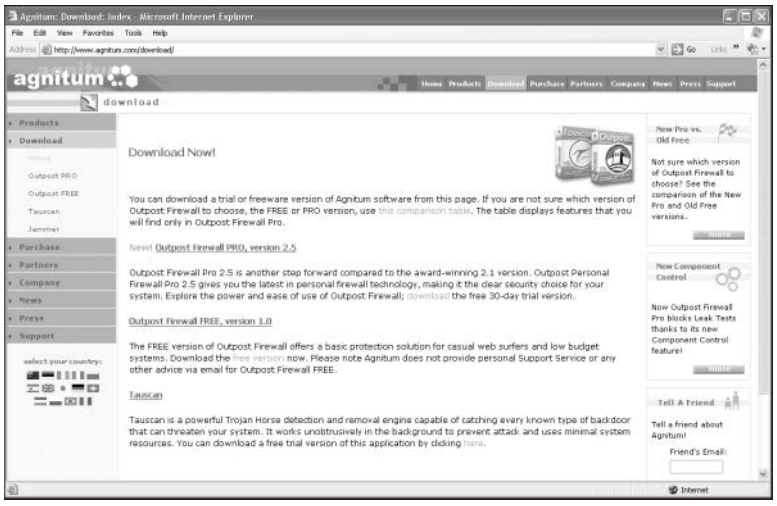

FIGURE 9.1 Firewall protection from Agnitum.

Listed and shown next are other sources for information on free firewall protection. Each of these websites offers links to a number of sources for free firewall protection as well as to other useful security tools. You may want to explore these sites as well as add them to your list of resource sites:

- http://www.zonealarm.com/software/free-firewall/
- https://www.comodo.com/home/internet-security/firewall.php

Firewall Logs

Firewalls are also excellent tools when attempting to ascertain what has happened after an incident occurs. Almost all firewalls, regardless of type or implementation, will log activity. These logs can provide valuable information that can assist in determining the source of an attack, methods used to attack, and other data that might help either locate the perpetrator of an attack or at least prevent a future attack using the same techniques. Any security-conscious network administrator should make it a routine habit to check the firewall logs.

Antispyware

Antispyware, as discussed earlier in this book, scans your computer to see whether there is spyware running on your machine. This is an important element of computer security software that was at one time largely ignored. Even today, not enough people take spyware seriously or guard against it. Most antispyware works by checking your system for known spyware files. Each application must simply be checked against a list of known spyware. This means that you must maintain some sort of subscription

service so that you can obtain routine updates to your spyware definition list. Most antivirus solutions now also check for spyware.

In today's Internet climate, running antispyware is as essential as running antivirus software. Failing to do so can lead to serious consequences. Personal data, and perhaps sensitive business data, could easily be leaking out of your organization without your knowledge. And, as was pointed out earlier in this book, it is entirely possible for spyware to be the vehicle for purposeful industrial espionage.

Barring the use of antispyware, or even in conjunction with such software, you can protect yourself via your browser's security settings as was discussed in a previous chapter. Additionally, several times throughout this book, you have been warned to be cautious about attachments and Internet downloads. You would also be well advised to avoid downloading various Internet "enhancements," such as "skins" and "toolbars." If you are in an organization, prohibiting such downloads should be a matter of company policy. Unfortunately, many websites today require some sort of add-in such as Flash in order to function properly. The best advice for this situation is to only allow add-ins on trusted, well-known sites.

IDS

IDS has become much more widely used in the last few years. Essentially, an IDS will inspect all inbound and outbound port activity on your machine/firewall/system and look for patterns that might indicate an attempted break-in. For example, if the IDS finds that a series of ICMP packets were sent to each port in sequence, this probably indicates that your system is being scanned by network-scanning software, such as Cerberus. Since this is often a prelude to an attempt to breach your system security, it can be very important to know that someone is performing preparatory steps to infiltrate your system.

Entire volumes have been written on how IDS systems work. This chapter cannot hope to cover that much information. However, it is important that you have a basic idea of how these systems work.

The sections that follow will first examine the broad categories in which IDS systems tend to be viewed and then will also look at some specific approaches to IDS. While this information is not all inclusive, it does address the more common terminology used.

IDS Categorization

There are a number of ways in which IDS systems can be categorized. The most common IDS categorizations are as follows:

- Passive IDS
- Active IDS (also called Intrusion Prevention System, or IPS)

Passive IDS

A passive IDS just monitors suspicious activity and then logs it. In some cases it may notify the administrator of the activity in question. This is the most basic type of IDS. Any modern system should have, at a minimum, a passive IDS along with the firewall, antivirus, and other security measures taken.

Active IDS

An active IDS or IPS takes the added step of shutting down the suspect communication. Just like antivirus, it is possible for an IDS to have a false positive. It might suspect something is an attack when in fact it is legitimate traffic. Whether one uses an IDS or IPS is a decision that must be made after a thorough risk analysis.

Imagine an IDS that is looking at threshold monitoring to determine if an attack is occurring. A particular user normally works between the hours of 8 a.m. and 5 p.m. and uses a relatively small amount of bandwidth. The IDS detects the user at 10 p.m. using 10 times his normal bandwidth. This seems like it might be an attack, and the active IDS (IPS) shuts down the offending traffic. However, it is later found that this was a legitimate user working late on a critical project that was due to a client the next day, and your IPS prevented that from happening.

This is an excellent place to consider risk analysis. You have to weigh the hazards of false positives against the risk of allowing an attack to proceed undetected before deciding if a passive IDS or an IPS is appropriate for your organization. It is often the case that different network segments will have different risk profiles. You may find that a passive IDS is appropriate for most of your network but that an IPS is needed for the most sensitive network segments.

Identifying an Intrusion

There are really two ways of identifying an intrusion. The first method is signature based. This is similar to the signatures used by antivirus. However, IDS signatures cover issues beyond malware. For example, certain DoS attacks have specific signatures that can be recognized.

The second method is statistical anomaly. Essentially, any activity that seems outside normal parameters and far enough said parameters to be a likely attack is identified as a probable attack. Any number of activities can trigger this type of alert. An example can be a sudden increase in bandwidth utilization or user accounts accessing resources they have never accessed before.

Most IDSs will use both forms of attack recognition. The two real issues for selecting an IDS are its ease of use and its signature database. There are certainly other considerations such as price, but those two are the most central to deciding on an IDS.

IDS Elements

Whether it is an active IDS or a passive IDS, and regardless of whether it is commercial or open source, certain elements/terms are common to all IDSs.

- A *sensor* is the IDS component that collects data and passes it to the analyzer for analysis.
- The *analyzer* is the component or process that analyzes the data collected by the sensor.
- The *manager* is the IDS interface used for management. It is a software component to the IDS.
- The *operator* is the person primarily responsible for the IDS.

- *Notification* is the process or method by which the IDS manager makes the operator aware of an alert.
- An *activity* is an element of a data source that is of interest to the operator. It may or may not be a possible attack.
- An *event* is any activity that is deemed to be suspicious and a possible attack.
- An *alert* is a message from the analyzer indicating that an event has occurred.
- The *data source* is the raw information that the IDS is analyzing to determine if there has been an event.

All these elements are part of an IDS and function together to capture traffic, analyze that traffic, and report anomalous activity to the operator of the IDS. An IPS will have additional elements capable of shutting down offending traffic.

Snort

There are a number of vendors who supply IDS systems, each with its own strengths and weaknesses. Which system is best for your environment is contingent on many factors including the network environment, security level required, budget constraints, and skill level of the person who will be working directly with the IDS. One popular open-source IDS is Snort, which can be downloaded for free from www.snort.org/.

We will examine Snort briefly in this section. While it is not the only IDS available, it is free, and that makes it an attractive option for many people. We will walk through the basic configuration of Snort for Windows.

First you must visit www.snort.org and register. It is free. Then download the Snort installation program and the latest rules. Make certain you download the installer that has an .exe extension. The .rpm extensions are for Linux. Also, I have found that certain versions of Microsoft Internet Explorer do not work well with the Snort website, so it is recommended that you use an alternative browser such as Mozilla Firefox.

Once you have downloaded both the rules and the installation, start the installation. Most of it is quite simple. There is a screen that asks you if you wish to support database connectivity. For most live situations, you would want to dump your Snort records to some database. However, for demonstration purposes, choose “I do not plan to log to a database,” as shown in Figure 9.2.

FIGURE 9.2 Snort database selection.

Other than this, simply use all default settings. At the end of the install, it will also attempt to install WinPCAP. If for some reason this fails, you will need to download and install it separately. WinPCAP is an open source tool for capturing packets. All IDSs depend on packet capturing.

Now copy rules you downloaded from wherever you saved them to C:\snort\rules. Then copy the configuration file from C:\snort\rules\etc\snort.conf to C:\snort\etc. Open that configuration file using WordPad, not Notepad. Notepad does not support word wrap, and it will be difficult to read the configuration file in Notepad.

The first step is to change the HOME_NET *any* to your machine's IP address, as shown in Figure 9.3. In a live situation, we would also set the other IP addresses (web server, SQL server, DNS server, and so on).

```
##################################################
# Step #1: Set the network variables.  For more information, see
README.variables
##################################################

# Setup the network addresses you are protecting
var HOME_NET any

# Set up the external network addresses.  A good start may be
"any"
var EXTERNAL_NET any

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET

# List of web servers on your network
var HTTP_SERVERS $HOME_NET

# List of sql servers on your network
var SQL_SERVERS $HOME_NET
```

FIGURE 9.3 HOME_NET address.

Now you need to find and change the rules paths. They will have Linux-style paths, as shown in Figure 9.4.

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an
absolute path,
# such as:  c:\snort\rules
var RULE_PATH ../rules
var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH ../preproc_rules
```

FIGURE 9.4 Linux-style paths.

You will need to change them to Windows-style paths, as shown in Figure 9.5.

```
var RULE_PATH c:\snort\rules
var SO_RULE_PATH c:\snort\rules\so_rules
var PREPROC_RULE_PATH c:\snort\rules\preproc_rules
```

FIGURE 9.5 Windows-style paths.

You will now need to find and change the library paths. This is a bit harder because the names of the paths and the files are a bit different in Windows. The Linux style library paths will appear as the ones shown in Figure 9.6.

```
# path to dynamic preprocessor libraries
dynamicpreprocessor directory
/usr/local/lib/snort_dynamicpreprocessor/

# path to base preprocessor engine
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so

# path to dynamic rules libraries
# dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

FIGURE 9.6 Linux-style library paths.

You can find your Windows pathnames and filenames by looking in the folder shown in Figure 9.7.

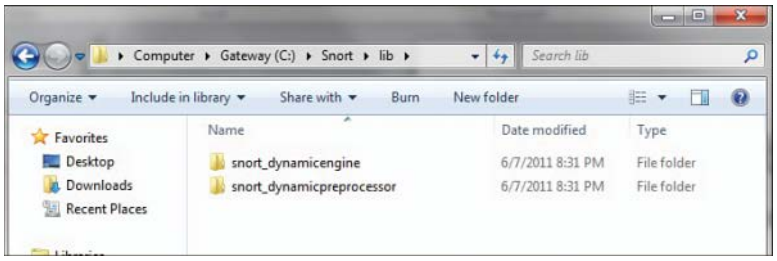

FIGURE 9.7 Windows-style library paths.

Note: If you find that you do not have a particular file or path in your system, just make sure it is commented out in the configuration file.

You must find the reference data and change it from Linux-style paths to Windows-style paths, as shown in Figure 9.8.

```
# metadata reference data.  do not modify these lines
include C:\Snort\etc\classification.config
include C:\Snort\etc\reference.config
```

FIGURE 9.8 Reference paths.

You are almost done. Now search for

```
#output log_tcp dump
```

and after that put this line

```
output alert_fast: alert.ids
```

Note: the pound sign (#) indicates a comment.

Now you will need to use the command line to start Snort. Simply navigate to C:\snort\bin. There are several different ways to start Snort. Many of the common ones are shown here in Table 9.1. I recommend you try the simplest first.

TABLE 9.1 Snort Commands

Command	Purpose
`snort -v`	Start Snort as just a packet sniffer.
`snort -vd`	Start Snort as a packet sniffer, but have it sniff packet data rather than just the headers.
`snort -dev -l ./log`	Start Snort in logging mode so it logs packets.
`snort -dev -l ./log -h` *192.168.1.1/24* `-c snort.conf` (Put your IP address where you see italics.)	Start Snort in IDS mode.

Snort is free and open source, but many people have a great deal of difficulty working with it the first time. The slightest error in your configuration file or the command-line startup will cause it to not run correctly. The purpose of this section is just to introduce you to Snort. For more information on Snort, try the following:

- **Snort Manual:** http://manual-snort-org.s3-website-us-east-1.amazonaws.com/
- **Writing your own Snort rules:** http://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm

Honey Pots

A honey pot is an interesting technology. Essentially, it assumes that an attacker is able to breach your network security. And it would be best to distract that attacker away from your valuable data. Therefore, one creates a server that has fake data—perhaps an SQL server or Oracle server loaded with fake data, and just a little less secure than your real servers. Then, since none of your actual users ever access this server, monitoring software is installed to alert you when someone does access this server.

A honey pot achieves two goals. First, it will take the attacker's attention away from the data you wish to protect. Second, it will provide what appears to be interesting and valuable data, thus leading the attacker to stay connected to the fake server, giving you time to try to track them. There are commercial solutions, like Specter (www.specter.com). These solutions are usually quite easy to set up and include monitoring/tracking software. You may also find it useful to check out www.honeypots.org for more information on honey pots in general, and on specific implementations.

Database Activity Monitoring

Database activity monitoring (DAM) is monitoring and analyzing database activity that operates independently of the database management system (DBMS). It is separate from the DBMS auditing, logging, and monitoring. Database activity monitoring and prevention (DAMP) is an extension to DAM that goes beyond monitoring and alerting to also block unauthorized activities.

Other Preemptive Techniques

Besides IDS, antivirus, firewalls, and honey pots, there are a variety of preemptive techniques an administrator can use to attempt to reduce the chances of a successful attack being executed against her network.

Intrusion Deflection

This method is becoming increasingly popular among the more security-conscious administrators. The essence of it is quite simple. An attempt is made to attract the intruder to a subsystem set up for the purpose of observing him. This is done by tricking the intruder into believing that he has succeeded in accessing system resources when, in fact, he has been directed to a specially designed environment. Being able to observe the intruder while he practices his art will yield valuable clues and can lead to his arrest.

This is often done by using what is commonly referred to as a *honey pot.* Essentially, you set up a fake system, possibly a server that appears to be an entire subnet. You make that system look very attractive by perhaps making it appear to have sensitive data, such as personnel files, or valuable data, such as account numbers or research. The actual data stored in this system is fake. The real purpose of the system is to carefully monitor the activities of any person who accesses the system. Since no legitimate user ever accesses this system, it is a given that anyone accessing it is an intruder.

Intrusion Deterrence

This method involves simply trying to make the system seem like a less palatable target. In short, an attempt is made to make any potential reward from a successful intrusion attempt appear more difficult than it is worth. This approach includes tactics such as attempting to reduce the apparent value of the current system's worth through camouflage. This essentially means working to hide the most valuable aspects of the system. The other tactic in this methodology involves raising the perceived risk of a potential intruder being caught. This can be done in a variety of ways, including conspicuously displaying warnings and warning of active monitoring. The perception of the security of a system can be drastically improved, even when the actual system security has not been improved.

Authentication

When a user logs on to a system, the system needs to authenticate her (and sometimes the user needs to authenticate the system). There are many authentication protocols. A few of the more common are briefly described here:

- **PAP:** Password Authentication Protocol is the simplest form of authentication and the least secure. Usernames and passwords are sent unencrypted, in plain text. This is obviously a very old method that is not used anymore. However, in the early days of computing, there were no widely available packet sniffers, and security was far less of a concern.
- **SPAP:** Shiva Password Authentication Protocol is an extension to PAP that does encrypt the username and password that is sent over the Internet.
- **CHAP:** Challenge Handshake Authentication Protocol calculates a hash after the user has logged in. Then it shares that hash with the client system. Periodically the server will ask the client to provide that hash. (This is the challenge part.) If the client cannot, then it is clear that the communications have been compromised. MS-CHAP is a Microsoft-specific extension to CHAP. The steps are basically these:
    1. After the handshake phase is complete, the authenticator (often the server) sends a "challenge" message to the peer.
    2. The peer responds with a value calculated using a "one-way hash" function.
    3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise, the connection should be terminated.

4. At random intervals, the authenticator sends a new challenge to the peer and repeats steps 1 to 3.

The entire goal of CHAP is to not only authenticate, but periodically reauthenticate, thus preventing session hijacking attacks.

- **EAP:** A framework frequently used in wireless networks and point-to-point connections. It was originally defined in RFC 3748 but updated since then. It handles the transport of keys and related parameters. There are several versions of EAP. It has many variations, including these:
  - **LEAP:** Lightweight Extensible Authentication protocol was developed by Cisco and has been used extensively in wireless communications. LEAP is supported by many Microsoft operating systems including Windows 7 and later versions. LEAP uses a modified version of MS-CHAP.
  - **Extensible Authentication Protocol—Transport Layer Security:** This utilizes TLS in order to secure the authentication process. Most implementations of EAP-TLS utilize X.509 digital certificates to authenticate the users.
  - **Protected Extensible Authentication Protocol (PEAP):** This encrypts the authentication process with an authenticated TLS tunnel. PEAP was developed by a consortium including Cisco, Microsoft, and RSA Security. It was first included in Microsoft Windows XP.
- **Kerberos:** Kerberos is used widely, particularly with Microsoft operating systems. It was invented at MIT and derives its name from the mythical three-headed dog that was reputed to guard the gates of Hades. The system is a bit complex, but the basic process is as follows:

  When a user logs in, the authentication server verifies the user's identity and then contacts the ticket-granting server. (These are often on the same machine.) The ticket-granting server sends an encrypted "ticket" to the user's machine. That ticket identifies the user as being logged in. Later when the user needs to access some resource on the network, the user's machine uses that ticket-granting ticket to get access to the target machine. There is a great deal of verification for the tickets, and these tickets expire in a relatively short time.

More on Kerberos

Since Kerberos is so widely used, it bears a bit closer look than the other authentication methods. In this section we will look a bit more in depth at Kerberos. If this is your first exposure to Kerberos, you may need to read this section more than once to really digest. While there are variations, the basic process is shown in Figure 9.9.

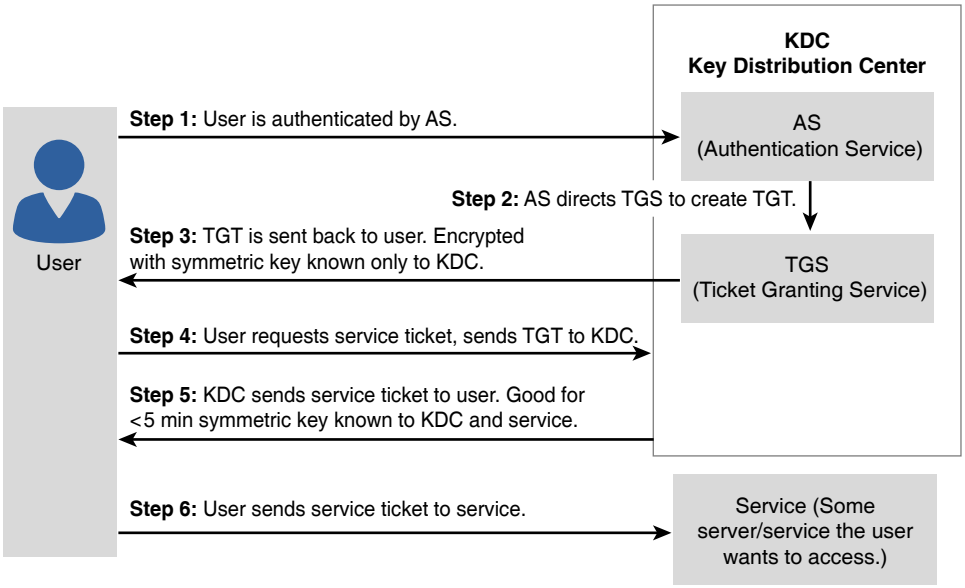

FIGURE 9.9 Kerberos

The elements of Kerberos follow:

- **Principal:** A server or client that Kerberos can assign tickets to.
- **Authentication server (AS):** Server that authorizes the principal and connects it to the ticket-granting server.
- **Ticket-granting server (TGS):** Provides tickets.
- **Key distribution center (KDC):** A server that provides the initial ticket and handles TGS requests. Often it runs both AS and TGS services. It must be noted that Kerberos is one of the most widely used authentication protocols. Europe often uses an alternative SESAME Secure European System for Applications in a multivendor environment.

Digital Certificates

It seems very likely that you have heard the term *digital certificate* previously. The first thing you may wonder is what does a digital certificate do? First recall our discussions of asymmetric cryptography in Chapter 8, "Encryption." We mentioned that the public key can be disseminated widely since it can only be used to encrypt messages to us. Well, how does one provide people with a public key? The most common method is via a digital certificate. The digital certificate contains the user's public key, along with other information. However, a digital certificate can provide much more. It can provide a means for authenticating that the holder of the certificate is who she claims to be.

X.509 is an international standard for the format and information contained in a digital certificate. X.509 is the most common type of digital certificate in the world. It is a digital document that contains a public key signed by the trusted third party that is known as a certificate authority, or CA.

The following are the basic items in an X.509 certificate. There can be other optional information:

- **Version:** This is the version of X.509 that this certificate complies with.
- **Certificate holder's public key:** This is the primary way of getting someone's public key from his X.509 certificate.
- **Serial number:** This is a unique identifier for this certificate.
- **Certificate holder's distinguished name:** This is often a domain name or email associated with a certificate.
- **Certificate's validity period:** One year is the most common validity period.
- **Unique name of certificate issuer:** This is the certificate authority that issued this certificate.
- **Digital signature of issuer:** This field, and the next, are used to verify the certificate itself.
- **Signature algorithm identifier:** Identifies the actual digital signature algorithm used.

Let us see how this works in a common scenario. You visit your bank's website. In order to get the bank's public key, your browser will download that bank's digital certificate. But there is a problem. Could someone have set up a fake site, claiming to be your bank? Could that person have also generated a fake certificate claiming to be the bank? Yes, it's possible. This is one place digital certificates help us out. Your browser will look at the certificate issuer listed on the certificate and first ask if that is a CA that your browser trusts. Assuming it is, then your browser communicates with that CA to get that CA's public key. (Recall from Chapter 8 that a digital signature is created with a private key and verified with the public key.) The browser uses that CA public key to verify the CA signature on the certificate. If this is a fake certificate, the digital signature won't be recognized. This means a certificate not only provides you with the certificate holder's public key, but also gives you a method of verifying that entity with a trusted third party.

It should be noted that unlike X.509 certificates, PGP (Pretty Good Privacy) certificates are not issued by a CA and don't have a mechanism for third-party verification. They are usually only used for email communication. That is because it is assumed that you know who you are emailing, and verifying that identity is not required.

There are some other terms and concepts in digital certificates you need to be familiar with. Let us begin with a CA, the entity that issues you a digital certificate. For example, Comodo, Symantec, Digicert, GoDaddy, Verisign, and Thawte are all well-known certificate authorities. When you purchase a certificate from one of these vendors, they first verify who you are. (That can be as simple as matching your credit card with the domain you are buying the certificate for, or it can be far more involved.)

Since verifying a certificate user can be time consuming, many CAs offload that process to a registration authority (RA), who will then notify the CA to issue the certificate (or not to).

A CRL (certificate revocation list) is a list of certificates issued by a CA that are no longer valid. CRLs are distributed in two main ways: In the push model, the CA automatically sends the CRL out a regular interval. In the pull model, the CRL is downloaded from the CA by those who want to see it to verify a certificate. The problem is that CRL is not real-time checking. Thus, the newer answer is "Online Certificate Status Checking Protocol" OCSP; the idea is to have a protocol that checks in real time if the certificate is still valid.

SSL/TLS

What sort of encryption is used on bank websites and e-commerce? In general, symmetric algorithms are faster and require a shorter key length to be as secure as asymmetric algorithms. However, there is the problem of how to securely exchange keys. Most e-commerce solutions use an asymmetric algorithm to exchange symmetric keys and then use the symmetric keys to encrypt the actual data.

When visiting websites that have an HTTPS at the beginning, rather than HTTP, the *S* denotes *secure*. That means traffic between your browser and the web server is encrypted. This is usually done with either SSL (Secure Sockets Layer) or TLS (Transport Layer Security). SSL, the older of the two technologies, was developed by Netscape. SSL and TLS are both asymmetric systems.

SSL is a technology employed to allow for transport-layer security via public key encryption. SSL was developed by Netscape for transmitting private documents via the Internet. By convention, URLs that require an SSL connection start with https instead of http. There have been several versions:

- Unreleased v1 (Netscape)
- Version 2 released in 1995 but had many flaws
- Version 3 released in 1996 RFC 6101
- Standard TLS1.0 RFC 2246 released in 1999
- TLS 1.1 was defined in RFC 4346 in April 2006
- TLS 1.2 was defined in RFC 5246 in August 2008. It is based on the earlier TLS 1.1 spec
- TLS 1.3 July 2014

The basic process of establishing an SSL/TLS connection is shown in Figure 9.10.

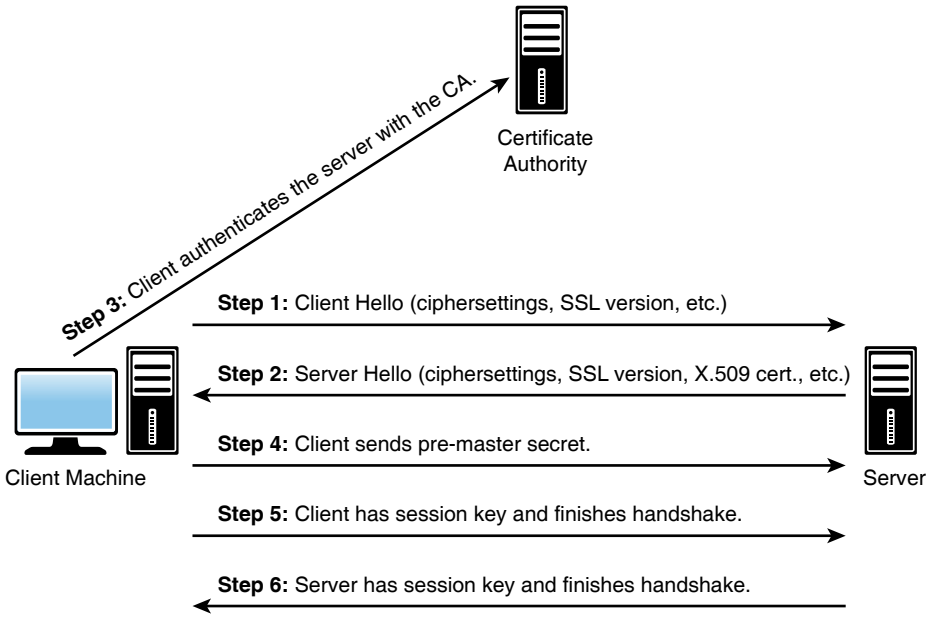

FIGURE 9.10 SSL/TLS.

The process involves several complex steps, as defined here:

1. The client sends the server information regarding the client's cryptographic capabilities. That includes what algorithms it is capable of, what hashing algorithms it can use for message integrity, and related information.
2. The server responds by selecting the best encryption and hashing that both client and server are capable of and sends this information to the client. The server also sends its own certificate, and if the client is requesting a server resource that requires client authentication, the server requests the client's certificate.
3. The client uses the information sent by the server to authenticate the server. This means authenticating the digital certificate with the appropriate CA. If this fails the browser warns the user that the certificate cannot be verified. If the server can be successfully authenticated, the client proceeds to the next step.
4. Using all data generated in the handshake thus far, the client creates the pre-master secret for the session, encrypts it with the server's public key that it received from the server's X.509 certificate, and then sends the encrypted pre-master secret to the server.
5. If the server has requested client authentication, then the server will also authenticate the client's X.509 certificate. This does not happen in most e-commerce and banking websites.

6. Both the client and the server use the master secret to generate the session keys. These are symmetric keys (such as AES) that will be used throughout the session to encrypt information between the client and the server.
7. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key.
8. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key.

This process provides a process to not only securely exchange a symmetric key, but also to verify the server and (optionally) verify the client. This is how secure web traffic is accomplished.

Virtual Private Networks

A *VPN* is a *virtual private network*. This is essentially a way to use the Internet to create a virtual connection between a remote user or site and a central location. The packets sent back and forth over this connection are encrypted, thus making it private. The VPN must emulate a direct network connection.

There are three different protocols that are used to create VPNs:

- Point-to-Point Tunneling Protocol (PPTP)
- Layer 2 Tunneling Protocol (L2TP)
- Internet Protocol Security (IPsec)

These are each discussed in more depth in the following sections.

Point-to-Point Tunneling Protocol

Point-to-Point Tunneling Protocol (PPTP) is the oldest of the three protocols used in VPNs. It was originally designed as a secure extension to Point-to-Point Protocol (PPP). PPTP was originally proposed as a standard in 1996 by the PPTP Forum—a group of companies that included Ascend Communications, ECI Telematics, Microsoft, 3Com, and U.S. Robotics. It adds the features of encrypting packets and authenticating users to the older PPP protocol. PPTP works at the data link layer of the OSI model (discussed in Chapter 2, "Networks and the Internet").

PPTP offers two different methods of authenticating the user: Extensible Authentication Protocol (EAP) and Challenge Handshake Authentication Protocol (CHAP). EAP was actually designed specifically for PPTP and is not proprietary. CHAP is a three-way process whereby the client sends a code to the server, the server authenticates it, and then the server responds to the client. CHAP also periodically reauthenticates a remote client, even after the connection is established.

PPTP uses Microsoft Point-to-Point Encryption (MPPE) to encrypt packets. MPPE is actually a version of DES. DES is still useful for many situations; however, newer versions of DES, such as DES 3, are preferred.

Layer 2 Tunneling Protocol

Layer 2 Tunneling Protocol (L2TP) was explicitly designed as an enhancement to PPTP. Like PPTP, it works at the data link layer of the OSI model. It has several improvements to PPTP. First, it offers more and varied methods for authentication—PPTP offers two, whereas L2TP offers five. In addition to CHAP and EAP, L2TP offers PAP, SPAP, and MS-CHAP.

Besides more authentication protocols available for use, L2TP offers other enhancements. PPTP will only work over standard IP networks, whereas L2TP will work over X.25 networks (a common protocol in phone systems) and ATM (asynchronous transfer mode, a high-speed networking technology) systems. L2TP also uses IPsec for its encryption.

IPsec

IPsec is the latest of the three VPN protocols. One of the differences between IPsec and the other two methods is that it encrypts not only the packet data (recall the discussion of packets in Chapter 2), but also the header information. It also has protection against unauthorized retransmission of packets. This is important because one trick that a hacker can use is to simply grab the first packet from a transmission and use it to get their own transmissions to go through. Essentially, the first packet (or packets) has to contain the login data. If you simply resend that packet (even if you cannot crack its encryption), you will be sending a valid logon and password that can then be followed with additional packets. Preventing unauthorized retransmission of packets prevents this from happening.

IPsec operates in one of two modes: Transport mode, in which only the payload is encrypted, and Tunnel mode, in which both data and IP headers are encrypted. Following are some basic IPsec terms:

- Authentication Headers (AHs) provide connectionless integrity and data origin authentication for IP packets.
- Encapsulating Security Payloads (ESPs) provide origin authenticity, integrity, and confidentiality protection of packets. These have encryption-only and authentication-only configurations.
- Security Associations (SAs) provide the parameters necessary for AH or ESP operations. SAs are established using the Internet Security Association and Key Management Protocol.
- The Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange.
- Internet key exchange (IKE and IKEv2) is used to set up a security association (SA) by handling negotiation of protocols and algorithms and to generate the encryption and authentication keys to be used.

Essentially during the initial establishment of an IPsec tunnel, security associations (SAs) are formed. These SAs have information such as what encryption algorithm and what hashing algorithms will be used in the IPsec tunnel. Recall that we discussed encryption in some depth in Chapter 8. IKE is primarily concerned with establishing these SAs. ISAKMP allows the two ends of the IPsec tunnel to authenticate to each other and to exchange keys.

Wi-Fi Security

With wireless networks being so prevalent, it is important to consider wireless network security. There are three Wi-Fi security protocols, ranging from the oldest and least secure (WEP) to the most recent and most secure (WPA2). They are each briefly described here.

Wired Equivalent Privacy

Wired Equivalent Privacy (WEP) uses the stream cipher RC4 to secure the data and a CRC-32 checksum for error checking. Standard WEP uses a 40-bit key (known as WEP-40) with a 24-bit initialization vector (IV) to effectively form 64-bit encryption. 128-bit WEP uses a 104-bit key with a 24-bit IV.

Because RC4 is a stream cipher, the same traffic key must never be used twice. The problem with WEP is that the committee who created it was composed of very good computer professionals who thought they knew enough about cryptography, but did not. They reuse the IV. That defeats the entire purpose of an IV and leaves the protocol open to attacks. A simple search of YouTube for "how to crack WEP" will yield a deluge of videos on techniques for cracking WEP.

Wi-Fi Protected Access

Wi-Fi Protected Access (WPA) was definitely an improvement over WEP. First, WPA uses AES, which is a very good encryption algorithm. Then WPA uses Temporal Key Integrity Protocol. TKIP dynamically generates a new key for each packet. So even if you crack a WPA key, there will be a different key for the next packet.

WPA2

This is the most modern form of Wi-Fi security, and if it is at all possible, this is what you should be using. Thus, we will give it a bit more attention. WPA2 is based on the IEEE 802.11i standard. It provides the Advanced Encryption Standard (AES) using the Counter Mode-Cipher Block Chaining (CBC)-Message Authentication Code (MAC) Protocol (CCMP) that provides data confidentiality, data origin authentication, and data integrity for wireless frames. Some of these terms you should recall from Chapter 8. The Cipher Block Chaining prevents known plain text attacks.

The MAC preserves message integrity and ensures that packets were not altered in transit, either accidentally or intentionally. This means that WPA2 uses very strong encryption along with message integrity.