9 PROTECTING AGAINST MALWARE

Starting Point

Go to www.wiley.com/college/cole to assess your knowledge of protecting a computer against viruses, worms, and other malicious programs.
Determine where you need to concentrate your effort.

What You'll Learn in This Chapter

- Viruses
- Worms
- Trojan horses
- Spyware
- Web browser security
- Spam
- Email security

After Studying This Chapter, You'll Be Able To

- Identify various types of malicious code
- Mitigate the risk of a malware infection
- Configure web browser security settings
- Mitigate the risk of spam
- Identify safe email practices

INTRODUCTION

As software has become more powerful and users around the world have become more interconnected, the threat of a computer being infected with malicious code has ballooned. In this chapter you will learn about the types of malicious code you need to guard against and some steps for mitigating the threat. This chapter pays particular attention to two venues frequently used to spread malicious code: web pages and email.

9.1 Viruses and Other Malware

Before you can understand how to mitigate the threat of malicious code, you need to understand the types of malicious code being **propagated** (spread from computer to computer) and the methods of propagation. In this section, we'll look at various types of malicious code, which is also known as **malware** or **malcode**.

9.1.1 Viruses

A **virus** is a piece of code that inserts itself into legitimate software. As with a biological virus, the computer virus is not viable without a host. The virus needs the **host** software or file to propagate and carry out its mission. A virus is able to **replicate** (reproduce) itself and attach itself to a host file, a technique known as **self-propagation**.

Early viruses infected boot sectors of floppies and were spread by the sharing of applications on floppies. Today, floppies are too small to be practical for sharing applications, so **boot sector viruses** that are transmitted through floppy disks are not common anymore.

If the virus has attached itself to an application, the code in the virus is run every time the application runs. The virus code will have the same privileges as the host application. A typical example of a host for this kind of virus is a self-extracting video clip. When the unsuspecting user launches the file to extract the video, the virus code runs. This virus spreads by people sending the self-extracting video clip to their friends.

Some viruses are able to attach to data files such as spreadsheets and word processor files. These viruses are **scripts** that execute when the file is loaded. A script is code written in a scripting language, so it does not need to be **compiled** (converted from human-readable source code to binary machine language) into an executable. Instead, it is run by an application that supports such scripts.

One of the first widespread viruses to exploit scripts was **Melissa**, which spread by infecting Microsoft® Word files. When the Word files were opened,

the virus code would run and infect the Normal.dot template file used by the word processor. After Normal.dot was infected, any Word document saved would have the Melissa virus. Melissa used the **autorun macros** in a Word document to run a **Visual Basic® script (VBScript)** when an infected Word document was first opened. Microsoft now has a feature called **Macro Virus Protection** that can stop macros from running. This protection should not be disabled.

Email viruses move from PC to PC as part of the body of a message. When the virus code is executed, a message with the virus embedded is sent to other mail clients. The virus can either be an attachment that must be opened or an embedded script. Scripts can access the user's address book, and can use those addresses to propagate the virus-infected message.

One example of a virus that propagates through email is the **ILOVEYOU virus**. The ILOVEYOU virus first appeared in the spring of 2000 and was simply an attachment that users launched. Once launched, the virus's Visual Basic script sent out an infected message to everyone in the user's address book.

9.1.2 Worms

A **worm** is code able to replicate itself and propagate to other hosts by exploiting a vulnerability in a program. Most worms exploit previously identified vulnerabilities that are correctable with patches or upgrades. Therefore, the best protection against worms is to stay current with patches and upgrades for Windows® as well as for other major applications.

Another way to protect against worms is to minimize the services and applications running on a computer. For example, worms often target common, high-visibility applications, such as the Microsoft web server, Internet Information Server (IIS). If a computer does not need to serve web pages and it is not being used to develop an application that relies on IIS, IIS should be disabled on the computer.

9.1.3 Trojan Horses

A **Trojan horse** is a program that masquerades as a legitimate application, while also performing a covert function. Users believe they are launching a legitimate application, such as a screen saver. When the Trojan horse runs, the user has every indication that the expected application is running. However, the Trojan horse also runs additional code that performs a malicious activity.

The best way to detect a Trojan horse is to identify executable files that have been altered. This is most easily done by creating a baseline of **cyclic redundancy check (CRC)** values for all executable files on a workstation. A CRC calculates the file size and divides by a number, then stores the remainder of the operation. If an executable file is later altered to include a Trojan horse, it can be detected by comparing the current CRC value with the baseline value.

Trojan horses are more difficult to distribute than viruses and worms. They do not propagate on their own. They rely on users accepting questionable executables from untrusted sources.

Trojan horses are very powerful threats to the security of a computer, network, and organization. They bypass most security controls put in place to stop attacks. Trojan horses are not stopped by firewalls, intrusion detection systems (IDS), or access control lists (ACLs) because a user installs them just as they would any other application.

Logic Bombs

A **logic bomb** (also called **slag code**) is a type of Trojan horse that lies in wait until some event occurs. The most common trigger for a logic bomb is a date, in which case the code is known as a **time bomb**. The **Michelangelo** virus was an early logic bomb, created in 1991. Its trigger was March 6, Michelangelo's birthday. It was a particularly destructive logic bomb because it was designed to overwrite the hard disk. The **Nyxem Worm** is a more recent time bomb that activates on the third of each month. It disables file sharing security and virus protection and deletes certain file types, including Microsoft Office files, .zip files, and .rar files. The files with extensions .zip and .rar are compressed files.

The use of Trojan horses to launch **distributed denial-of-service (DDoS) attacks** is common. The attacker installs a logic bomb Trojan horse on a number of computers. When the triggering event occurs, those computers launch a denial-of-service attack against the target. The more computers hosting the Trojan horse, the more devastating the attack. The fact that the packets are coming from a number of locations also makes it more difficult to track down the source of the attack. When a computer is controlled to launch a DDoS attack, it is known as a **zombie**.

9.1.4 Browser Parasites

A **browser parasite** is a program that changes some settings in your browser. The parasite can have many effects on the browser, such as the following:

- Browser plug-in parasites can add a button or link add-on to the user's browser. When the user clicks the button or the link, information about the user is sent to the plug-in's owner. This can be a privacy concern.
- Browser parasites can change a user's start page or search page. The new page might be a "pay-per-click site," where the owner of the browser parasite earns money for every click.
- Browser parasites can transmit the names of the sites the user visits to the owner of the parasites. This can be used to formulate a more directed attack on the user.

9.1.5 Spyware

Spyware is a software application that gathers information about the computer and user. This information is then sent back to the developer or distributor of the spyware and is often used to serve ads to the user.

Targeted marketing has long been a part of a good sales program. The classic example is marketers that use census data to direct more effective mass-mailing campaigns. Census data is used to find certain zip codes that have the best demographics (characteristics such as age, number of children, and annual income) for the particular product being advertised. The use of census data and data compiled by companies that conduct market research is not as controversial because specific names and addresses have been removed, and the data is a summary of statistics for the zip code.

Spyware does not provide the developer with summarized data, but instead includes specifics on a named individual. Therefore, it is a violation of privacy and might make it possible for the person who receives the data to steal the victim's identity.

Typical information that can be reported includes the following:

- **User keystrokes:** User keystrokes can be used to capture passwords and other very sensitive data entered by the user.
- **Copies of emails:** Emails sent or received can be forwarded to the person wanting to monitor the user, unbeknownst to the user.
- **Copies of instant messages:** Essentially, any communications to and from the PC can be copied and sent to the spyware's owner.
- **Screen snapshots:** Even encrypted communications will at some point be displayed in clear text on the screen. At this point, the spyware can take a screen shot and send the image to whoever has developed or distributed the spyware.
- **Other usage information:** Login times, applications used, and websites visited are examples of other data that can be captured and reported back.

9.1.6 Backdoors

A **backdoor** (also called a **trapdoor**) is way for an attacker to access a computer without being detected or blocked by usual security measures. Often, the initial attack on a computer is potentially detectable by a firewall or IDS. So the attacker will install an application that will allow him to get back into the computer quickly and easily. These backdoors are often stealthy and difficult to detect.

If a Windows computer has been connected to the Internet for more than a day without any security protections in place, it most likely has been **rooted** and

FOR EXAMPLE

Social Engineering at Work

In January 2007, a worm called the Storm Worm was propagated through email attachments. The subject line of the email related to current news stories, a social engineering technique used to entice users to launch the attachment. One of the headlines used as the subject was "230 dead as storm batters Europe," giving the worm its name. The worm is a Trojan horse that can be used to launch a DDoS attack and send data.

As you can see, sophisticated attacks often do not fit into a specific category. Instead, they use a combination of methods to launch the attack. In this case, a social engineering attack was used to encourage users to launch malicious code, which then installed a Trojan horse and propagated itself as a worm.

Almost immediately, other variant subject lines began to emerge, making it impossible to identify the worm by subject line alone. In fact, within three days, the worm began appearing in emails with matchmaking subject lines instead of news stories. This example illustrates the fact that malware detection is a moving target. In this example, user education and attachment filtering helped prevent the worm from becoming even more of a problem than it was. The attachments were .exe files, so they were easily identified as executable code.

has a backdoor installed. In such a case, the best thing to do is wipe the system clean and re-install the Windows operating system. Although you can delete the application, you can never be sure that other changes have not been made on the computer. Some operating system and driver modifications are difficult to detect.

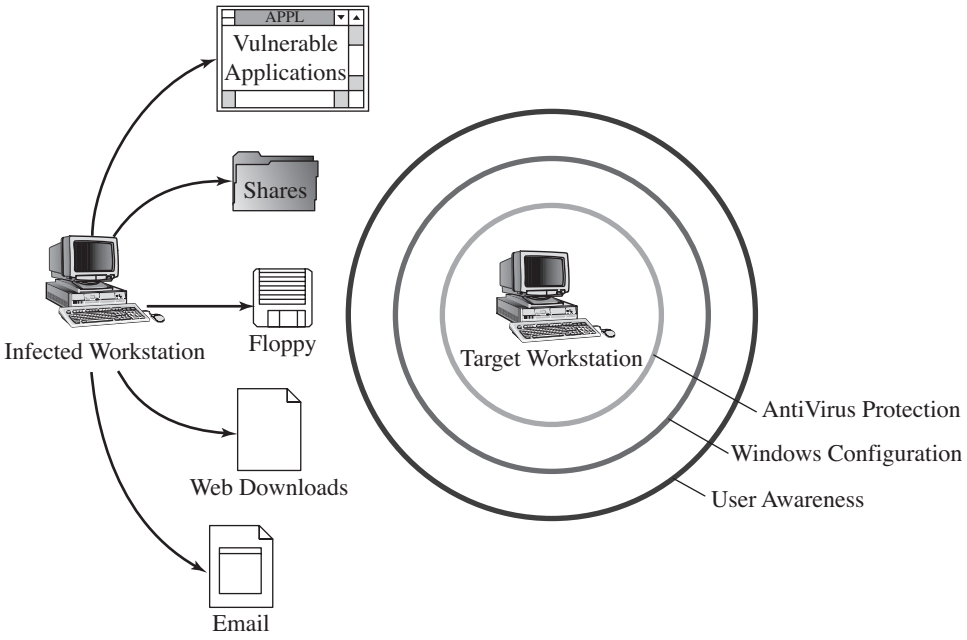
SELF-CHECK

1. Identify the types of malware that are self-propagating.
2. Describe a logic bomb.

9.2 Protecting the Workstation

Now that you have a basic understanding of the types of programs you are up against, let's look at some ways you can protect the computers on the network against these threats. Malware protection should focus on the following:

Figure 9-1



Protecting against malware.

- ▲ The use of antivirus and anti-spyware applications.
- ▲ Hardening the computer's configuration.
- ▲ User training and awareness.

This multilevel defense against viruses and worms is shown in Figure 9-1. Because new viruses and worms are constantly being created, the best protection is to run antivirus software, properly configure Windows, and educate users on safe practices.

It is important to note that anti-malware protection is also important on Linux-based computers, as well as on computers running Windows. Although currently a larger number of viruses and other malware programs are developed to target Windows computers, the number of malware programs that target Linux and Mac® OX is increasing. As these operating systems become more popular, they will become more desirable targets, and the number of malware attacks will increase even more.

This section will focus on protecting the workstation by looking at some general guidelines. The next two sections will look at defending against the two most common methods of propagation: web pages and email.

9.2.1 Antivirus Software

In today's threat environment, virus protection applications (**antivirus** programs) are no longer optional. A number of good antivirus products are available today, such as those from Symantec™, McAffee®, and Computer Associates™.

An organization should have protection on every computer where people are saving files, storing email messages, or browsing web pages. The antivirus software should be configured to provide real-time protection as well as routinely scheduled scanning. Without continuous protection, a virus can spread throughout an organization before the next routine scan is scheduled.

Keep Current with Antivirus Signatures

Because new viruses are always being released, antivirus software relies on periodic updated virus signature files to provide protection against the latest threats. A **virus signature** is the pattern of bits inside a virus that allows the antivirus software to recognize it.

Most signature updates are obtained by accessing the antivirus vendor's site and pulling down the latest update. Most antivirus packages will allow the administrator to choose to have the new signatures downloaded automatically on a regular schedule. Automating the process ensures that critical updates are not missed.

If the new antivirus signature is downloaded to be redistributed throughout a large organization, it should be tested first and deployed from a server within the organization. The local server, in turn, gets its files from a master server that distributes the tested update. There are four key steps to deploying updated signatures in a large organization:

1. Download new signatures.
2. Test new antivirus downloads.
3. Deploy new signatures.
4. Continue to monitor.

Finally, it is important that the computers be monitored periodically to ensure that the new antivirus signatures are being distributed properly. When the next big virus or worm hits is not the time to find a flaw in the system.

9.2.2 Anti-spyware

Anti-spyware software monitors a computer for spyware and allows you to remove it. There are a number of anti-spyware applications. In fact, some companies like Symantec and Microsoft sell an integrated package that includes antivirus and anti-spyware software. A term describing software that protects against a variety of malware is **anti-malware**. As with an antivirus application, you must keep your anti-spyware software up-to-date.

Some Internet service providers (ISPs) are so concerned about preventing malware that they offer security suites to their subscribers free of charge.

9.2.3 Computer Configuration Guidelines

Another important way to guard against malware is to make sure client computers are hardened. Many of the same guidelines apply as for hardening servers, including the following:

- Remove unnecessary services and applications.
- Filter traffic.
- Implement access control.

In this section, we'll look at a few specific precautions: personal firewalls, limiting user rights, and disabling hidden file extensions.

Personal Firewalls

A **personal firewall** is software that runs on the user's computer and blocks incoming and outgoing traffic. When used properly, a personal firewall can be very effective. For instance, a properly configured personal firewall can be very specific to a user's need for LAN traffic. A good way to configure a personal firewall is to start by blocking all traffic in and out of the computer. As the user encounters warnings of attempted activity that has been blocked, the user can choose to permit that traffic. In a short period of time, the user will have unblocked the majority of traffic he or she needs and the firewall will be configured to the user's very specific requirements.

Windows XP Professional with Service Pack 2 includes Windows Firewall. Other personal firewalls are available for Windows, Linux, and Mac OX from a variety of software distributors.

Limiting User Rights

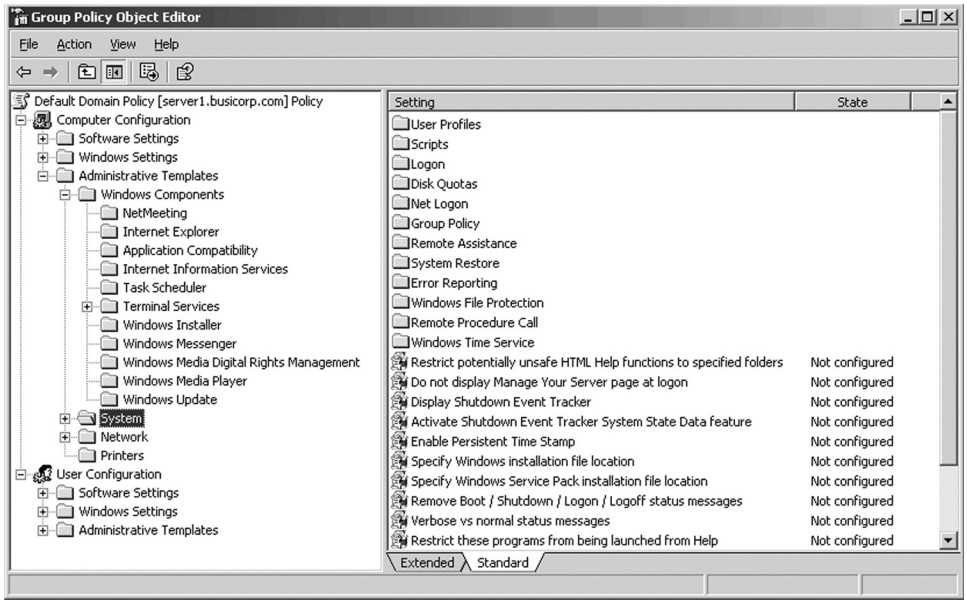
Remember that malware usually runs under the security context of the user who is logged in. Therefore, you should consider the rights and permissions a user has on his or her computer. For example, if a user does not need to install applications, the user should be prevented from doing so.

A **managed computer** is one that is configured through an automated policy. On an Active Directory® network, the policy is deployed through Group Policy Objects (GPOs). A large number of templates are available for configuring computers, as shown in Figure 9-2.

Some things you can do include the following:

- Prevent users from creating automated tasks through Task Scheduler.
- Disable services.

Figure 9-2



Computer configuration policies.

- ▲ Prevent users from installing applications.
- ▲ Set permissions on directories.
- ▲ Restrict the software that can run based on file type, publisher, and location.
- ▲ Restrict membership in specific groups.

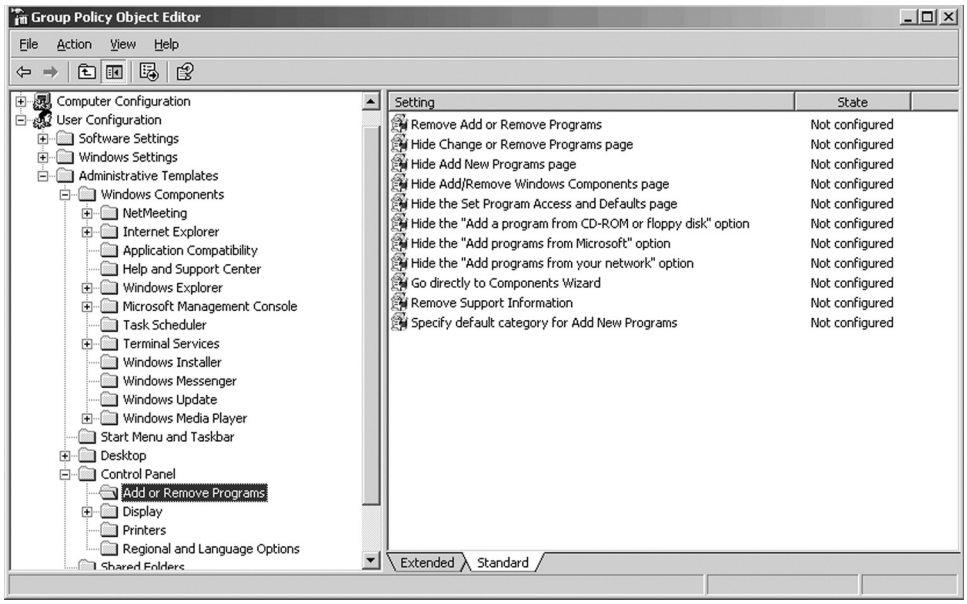
You can also create policies that apply to user accounts, as shown in Figure 9-3. For example, you might want to prevent users from accessing the Control Panel or from running applications unless they meet specific criteria.

On a NetWare® network, you can use ZENworks® to create managed computers and restrict the features available on a desktop. ZENworks patch management can also be used to ensure that desktop computers are kept up-to-date.

File Extensions

Windows has a feature that allows file extensions to be hidden from the user. Although this feature is designed to make the system more convenient and user friendly, this convenience comes at a security price. By hiding the extensions, malicious code is able to masquerade as something benign. For example, a user might be tempted to open a file named readme.txt, knowing that simple American Standard Code for Information Interchange (ASCII) text files cannot

Figure 9-3



User configuration policies.

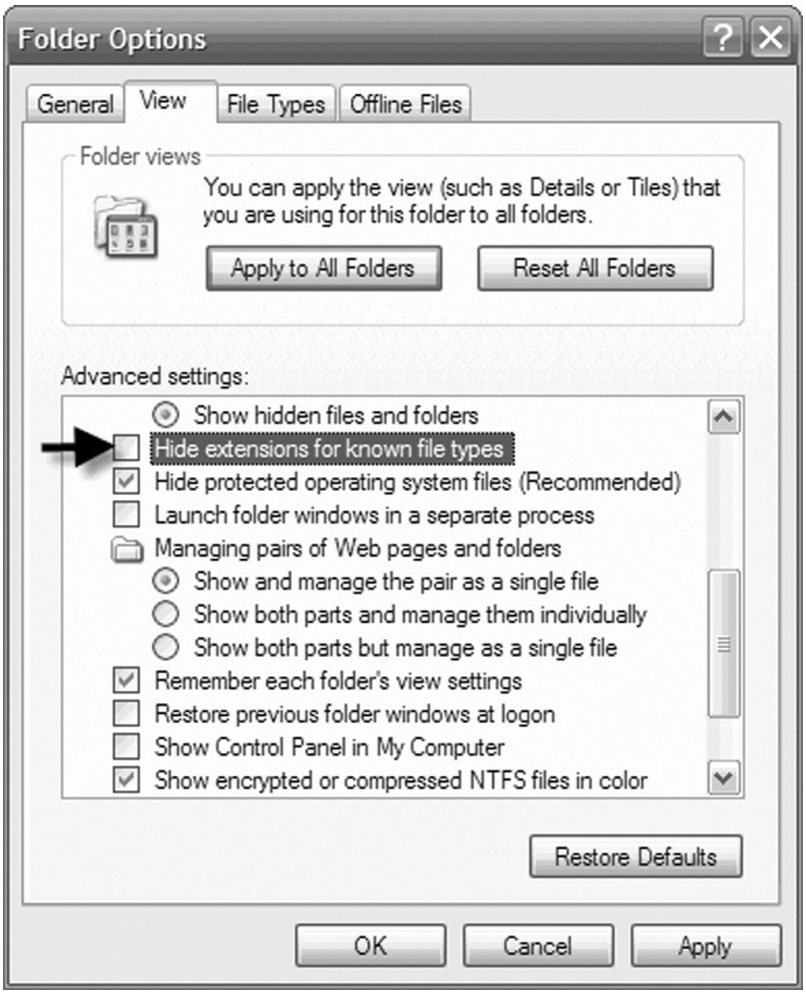
contain malicious code. However, the user will be at risk if the real file name is readme.txt.bat, because Windows hides the true extension, .bat. If the user opens the file by double clicking it, the malicious code in the BAT file will run with the same permissions as the user. File extension hiding should be disabled on Windows systems. You disable file extension hiding by removing the check from the "Hide file extensions for known file types" check box on the View tab of the Folder Options dialog box, as shown in Figure 9-4.

9.2.4 User Training

Windows users can also take steps to minimize the spread of viruses, worms, and Trojan horses on their systems. Remember, many malware schemes depend on social engineering to propagate. Some good practices for users include the following:

- Configure the email client to not download graphics and audio automatically. Some malicious email includes links that will let the attacker know that the file was opened. An attacker is more likely to continue to send malicious emails to addresses on which they are opened. A graphic or audio file embedded for this purpose is known as a **web beacon**.

Figure 9-4



Showing file extensions.

- Don't open any email from strangers that contain attachments.
- Only accept or open expected attachments. The user should have prior knowledge that the attachment was going to be sent. Many viruses today will at first appear to be legitimate messages. However, upon scrutiny, a user will be able to catch unexpected messages.
- Do not open attachments on email that seems vague or out of character. Watch out for nondescript messages such as "Check this out." The sender should include information in the message that helps the recipient

trust the attachment. For example, instead of "Check this out," the message should be "Bobby's graduation pictures," where the sender, Bobby, and a graduation are familiar or expected.

- If questionable email must be received and read, use a separate email client that is less susceptible to viruses. There are circumstances when a user is part of a public mailing list in which the members routinely share files. If attachments from this mailing list are routinely opened, an email client that does not run scripts should be used.
- Use macro protection in spreadsheet applications and word processors. Macros should be enabled on a case-by-case basis.

In many cases, the preceding procedures require a judgment call on the part of the user. Security awareness training is essential to help users recognize dangerous code before installing it on their computers.

FOR EXAMPLE

Adding a Computer to a Network Securely

When adding a new computer to the network, it is important that you harden the computer before attaching it to the network. Otherwise, you run the risk of installing malware before the computer has been completely configured and the latest security updates installed. Consider this scenario. You install Windows XP Professional on a computer and add it to the network. Next, you browse the Web to locate an anti-malware program. While browsing the Web, you unintentionally download spyware. When you reach the website distributing the anti-malware, you are asked to type in your email address, mailing address, phone number, and credit card information. The spyware monitors the data you enter and sends it to an advertiser, who then knows too much information about you. You install the anti-malware program and it detects and removes the spyware. However, your private data has already been compromised.

When setting up a new computer, you should always install the latest service packs and security updates, along with anti-malware software *before* connecting the computer to the Internet.

Another important consideration is to make sure you use a trusted source for anti-malware applications. A particularly annoying browser parasite called **SpyBlast** claims to locate and eradicate spyware, but in actuality it displays pop-up advertisements. A type of malware that displays pop-up ads or other advertisements is known as **adware**.

SELF-CHECK

1. What should malware protection focus on?
2. What is a *virus signature*?
3. What is the function of a personal firewall?

9.3 Web Browser Security

Web browsers today provide a lot more features than simply rendering images and HyperText Markup Language (HTML) code. Their convenience is greatly enhanced by their capability to do the following:

- Run Common Gateway Interface (CGI) scripts on the web server.
- Run scripts written in JavaScript® or VBScript on the web browser.
- Run executables such as Java™ and ActiveX® on the web browser host.
- Launch various plug-ins such as an audio player or movie player.

The convenience, productivity, and popularity of web browsers make them a prime target for hackers and would-be attackers. The hacker who develops an attack for a common web browser is sure to find many susceptible targets. This section looks at some of the risks associated with web browsers and how to mitigate them by configuring browser security settings and educating users.

9.3.1 Web Browser Risks

The security risks associated with browsing the Internet can be grouped into several categories:

- The web server might not be secure. All the data that users enter into their browsers is ultimately processed on the web server. In most cases, this information is stored in a database of some sort. Most typical users trust businesses to make sure their data is secure. However, the opposite is probably true. The best defense a user can have against an unsecure web server is to limit the sensitive data that is transmitted to the server.
- The browser runs malcode in the form of scripts or executables. Web applications often have dynamic elements that execute on the client. When used legitimately, these technologies improve the browsing experience. However, when used maliciously, client-side code can do a lot of harm to a computer.

- An attacker might eavesdrop on network traffic. Users should be aware that the security of the data transmitted to and from the web server is no more secure than the security of the network on which it travels. This risk can be reduced when the web server uses Secure Sockets Layer (SSL) to encrypt the data transmitted and received.
- A website might add browser parasites to your browser.
- An attacker might employ a man-in-the-middle attack. Web-based applications are **sessionless**, meaning that each request is sent independently, and are potentially susceptible to man-in-the-middle attacks, such as hijacking and replay.

9.3.2 Web Browser Technologies

Browsers support several types of client-side code. These features come with security risks, but are essential to many web-based applications. Therefore, it is important that you understand what these features are and why they present a risk so that you can configure the browser to limit or provide support for them according to a user's requirements and the organization's security policy.

Plug-ins

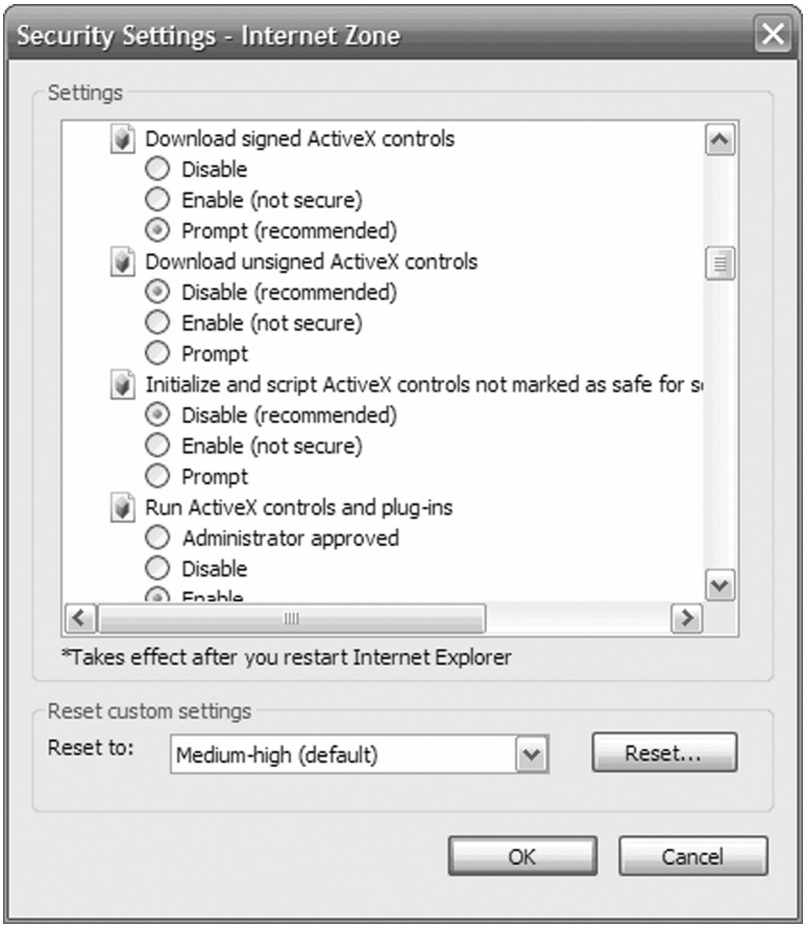
A **plug-in** or **add-on** is an addition to the browser that allows you to open a specific type of content. For example, the Adobe® Reader® plug-in allows you to view **Portable Document Format (PDF)** files in the browser window. The Macromedia® Flash® plug-in allows you to view and interact with Flash animations.

ActiveX

ActiveX is a technology developed by Microsoft for creating reusable content that can be distributed over the Internet or through an application installation. **ActiveX controls** are user interface elements that are embedded in a web page and must be downloaded to the client.

The use of ActiveX is a security risk because the browser places no restrictions on what an ActiveX control can do. To mitigate the risk of using ActiveX, each control can be digitally signed. The digital signatures can then be certified by a trusted certifying authority, such as VeriSign®. The developer can also indicate whether the control is safe for scripting and/or safe for executing. A control that is **safe for execution** should not modify files on the computer or perform other harmful tasks when it is added to a web page. A control that is **safe for scripting** should not allow parameters that could be used for harmful purposes to be set in script. For example, if a control is safe for scripting, it should not allow a web page developer to set a path to a file. It is up to the developer to be honest about the safety level of the control. Figure 9-5 shows some of the security settings you can configure to control how ActiveX controls are downloaded and executed by Internet Explorer® 7.

Figure 9-5



ActiveX control settings.

If the browser encounters an ActiveX control that hasn't been signed (or that has been signed but certified by an unknown certifying authority), the browser presents a dialog box warning the user that this action might not be safe. At this point the user can elect to accept the control or cancel the download. Few users that accept an unsigned control appreciate the risk involved. This is an area where user education is essential to preventing the installation of malevolent code.

Java

Java applets are programs written in the **Java** programming language that are run on the user's workstation. Java has a large number of security safeguards intended to avoid attacks. A **security manager** monitors what the applet does

and prevents it from performing tasks that are known to be risky. The following security features are part of the Java design:

- The security manager does not ordinarily allow applets to execute arbitrary system commands, to load system libraries, or to open up system device drivers such as disk drives.
- Applets are generally limited to reading and writing to files in a user-designated directory.
- An applet is only allowed to make a network connection back to the server from which it was downloaded.
- The security manager allows Java applets to read and write to the network and to read and write to the local disk but not to both. This limitation was created to reduce the risk of an applet spying on the user's private documents and transmitting the information back to the server.

JavaScript

JavaScript is a scripting language that can be executed by most browsers. The designers of JavaScript built security into the language itself. The basic approach was to eliminate the possibility of JavaScript code doing insecure activities by not providing commands or objects for those activities. The following are some examples of the security protections with JavaScript:

- JavaScript cannot open, read, write, create, or delete files. The language does not have any objects for managing files. A script cannot even list files and directories.
- JavaScript cannot access the network or network resources. The language does not have any objects for connecting or listening to the network interface.
- JavaScript can only access the domain from which it was downloaded. The script cannot access any other domain other than the one from which it originated.

Over the years, a number of security vulnerabilities have been discovered when using JavaScript. Patches and updated browsers have eliminated most of the security problems. However, the general concept that JavaScript is a potential avenue for the loss of private data still exists. For instance, JavaScript can access information available to the browser, such as URLs, cookies, names of files downloaded, and so on, and can make Hypertext Transfer Protocol (HTTP) requests. Scripts can request URLs and send other HTML information such as forms. This means the scripts could hit CGI programs that run on the web server.

Cookies

A **cookie** is an ASCII file created by a website to store information about the user visiting that site. This information is stored for the convenience of the website or for the convenience of the user. When a new request is made, the server can ask the browser to check if it has any cookies and, if it does, to pass those cookies back to the server. The browser can potentially pass any cookie to a web server. This could include cookies from completely different websites.

The contents of the cookie are under the control of the web server and might contain information about you or your past and present surfing habits, and when used maliciously can present a threat to a user's privacy.

There are two general types of cookies: persistent and nonpersistent. A **persistent cookie** is one that will survive reboots and last for a predetermined period of time. Persistent cookies are traditionally stored on the hard drive in a file such as "cookies.txt." This file can be read and edited by the user or system administrator. Some marketing companies have attempted to exploit user behavior by trying to capture these persistent cookies.

More and more people are becoming wary of cookies, especially those that can be used to track users over time. Therefore, many sites are starting to use nonpersistent cookies. A **nonpersistent cookie** (also called a **session cookie**) is stored in memory, so when the computer is turned off or rebooted, the cookie information is lost. Some browsers delete nonpersistent cookies when the user closes the browser, navigates to a different website, or after an elapsed period of time. There is no assurance that every browser will handle nonpersistent cookies the same way. The web server has no control over how the browser stores or disposes of the cookies; it merely tells the browser whether the cookie is meant to be persistent or nonpersistent.

9.3.3 Specific Threats to a Browser Session

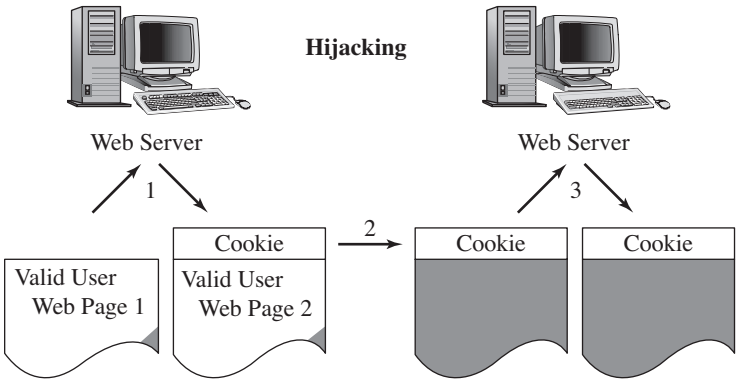
Now that you are familiar with some of the web technologies and the general risks they present, let's look at some specific attacks that target web browsers and sessions with web servers.

Hijacking Attack

Session hijacking occurs when an HTTP session is observed and captured by a network sniffer. The attacker modifies the captured traffic to allow the attacker to impersonate the client. All future traffic in the session is then channeled between the web server and the attacker.

The hijacking is usually done after the legitimate user has authenticated to the web server. Therefore, the attacker does not have to re-authenticate (usually for the remainder of the session). In this way, the attacker bypasses one of the major security features of the web-based session, the initial authentication.

Figure 9-6



1. A valid user does some web activity that results in him or her acquiring a cookie.
2. The cookie is stolen or captured by an attacker.
3. The cookie is transmitted with the attacker's attempt to access the application. The cookie authenticates the attacker as a valid user. The attacker gets access to the application.

Hijacking when cookies maintain state.

The hijacking attack exploits a weak method of maintaining **state** (information about the current session). If the attacker can understand how state is maintained, they might be able to inject themselves into the middle of the session by presenting the intercepted authentication cookie or credentials. This is illustrated in Figure 9-6.

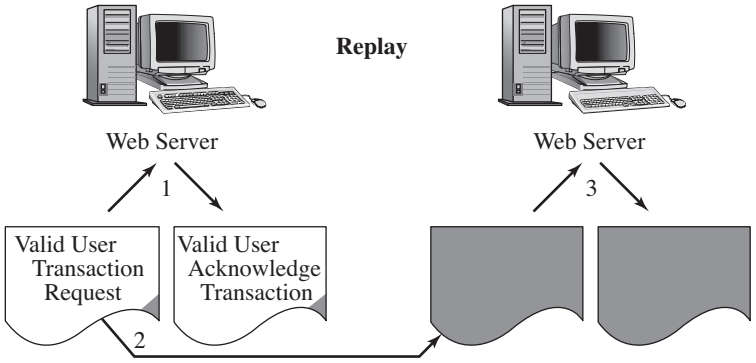
The attack involves the following steps:

1. A valid user performs some web activity that results in him or her acquiring a cookie.
2. The cookie is stolen or captured by an attacker.
3. The cookie is transmitted with the attacker's attempt to access the application. The cookie authenticates the attacker as a valid user. The attacker gets access to the application.

Replay Attack

As with a hijacking attack, the first step in a **replay attack** is to capture HTTP packets for a session. Some aspect of the session is then modified (certain replays, such as transferring bank funds, might not require modifications). The modified session is then fed back onto the network. If the replay is successful, the web server will believe the replayed traffic to be legitimate and will respond accordingly. This could produce a number of undesirable results. Figure 9-7 illustrates session replay.

Figure 9-7



1. A valid user does some web activity such as "Transfer $5,000 from account A to account B". There might or might not be a cookie.
2. The web page holding the transaction request is stolen or captured by an attacker.
3. The web page is retransmitted. The transaction is repeated—an additional $5,000 is transferred. The attacker can retransmit numerous times.
4. Depending on whether the attacker had to do spoofing, the final acknowledgment transaction might go back to the valid user's IP address where it is dropped beccause no session is open.

Replay attack.

The responsibility is on the web server to prevent replay attacks. The web server should be able to recognize replayed traffic as no longer being valid.

The following steps are involved in a replay attack:

1. A valid user performs some web activity such as "Transfer $5,000 from account A to account B." There might or might not be a cookie.
2. The web request is stolen or captured by an attacker.
3. The web request is retransmitted. The transaction is repeated; an additional $5,000 is transferred. The attacker can retransmit numerous times.
4. Depending on whether the attacker had to do spoofing, the acknowledgment might go back to the valid user's IP address, where it is dropped because no session is open.

9.3.4 Browser Configuration

Web browsers, like most Internet applications, respond to emerging security threats. In the early years, web browsers were very vulnerable. They had features making them convenient and productive but had no means for the user to make them more secure. Web browsers have evolved (due to the security threat), and users are now able to set various configuration settings to improve the security of their web browsers.

The problem with relying on the user to make security configuration decisions is that most users are not sophisticated and savvy when it comes to securing a web browser or even understanding the threat. Often users will not change any of the browser's security configuration settings, or even know they exist. The customization, for security purposes, is then left to the system or network administrator. However, as discussed earlier, browsing has become such an accepted norm for convenience and productivity that few users will tolerate less than total functionality. As a result, administrators who initially attempt to secure browsers are often beaten back by the onslaught of complaints and requests for help. In the end, the administrator must relax the web-browsing security settings.

In this section, we'll look at some areas where you can configure a browser to help mitigate the risks.

Secure Socket Layer

The Secure Socket Layer (SSL) protocol provides for the encryption of traffic between the web browser and server. SSL uses public-key encryption to exchange a symmetric key between the client and server; this symmetric key is used to encrypt the HTTP transaction (both request and response). Each transaction uses a different key. If the encryption for one transaction is broken, the other transactions are still protected.

Some of the settings related to digital certificates in Internet Explorer 7 are shown in Figure 9-8.

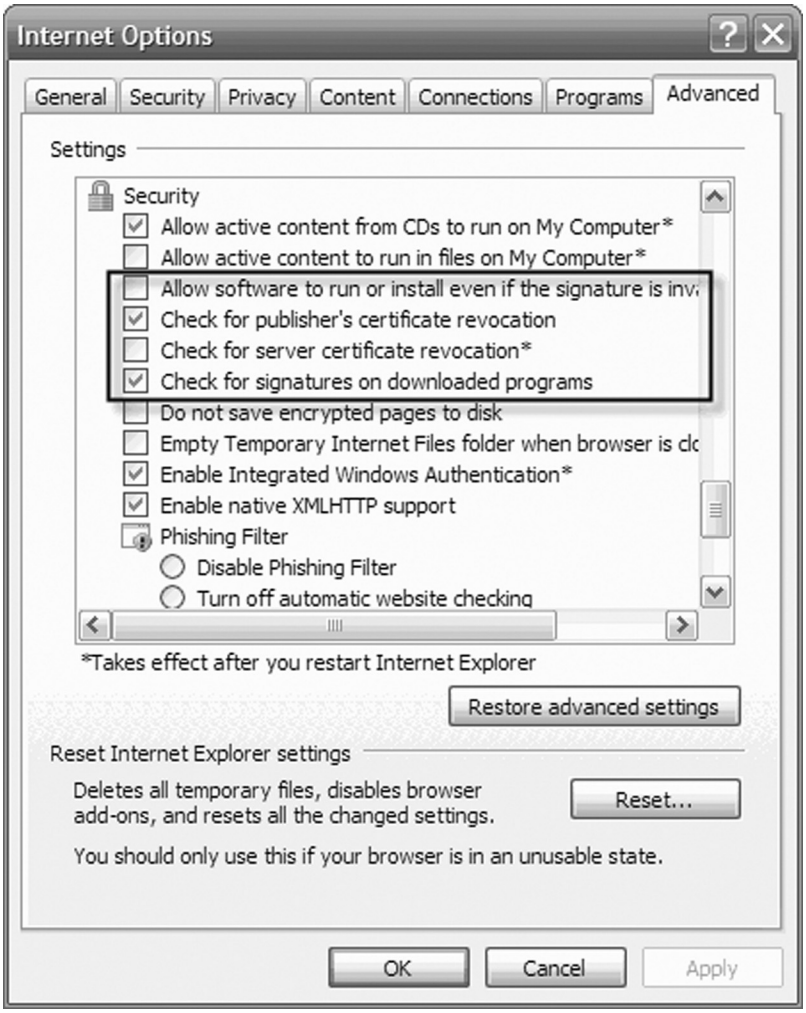
Settings related to digital certificates when using the Firefox® web browser on a Linux computer are shown in Figure 9-9.

A properly configured web browser will warn the user of a certificate problem if any of the following occur:

- The certificate was not signed by a trusted certificate authority.
- The certificate is currently invalid or has expired. Legitimate websites will keep their certificates up-to-date. This could indicate that the certificate has been stolen and is being used by a third party.
- The certificate or the certificate of the server that issued it has been revoked.
- The common name on the certificate does not match the domain name of the server. The host name of the web server is a fixed part of the site certificate. If the name of the web server doesn't match the name on the certificate, the browser will report the problem.

If a problem has been identified with the certificate, the user is prompted whether or not to accept the certificate.

Figure 9-8



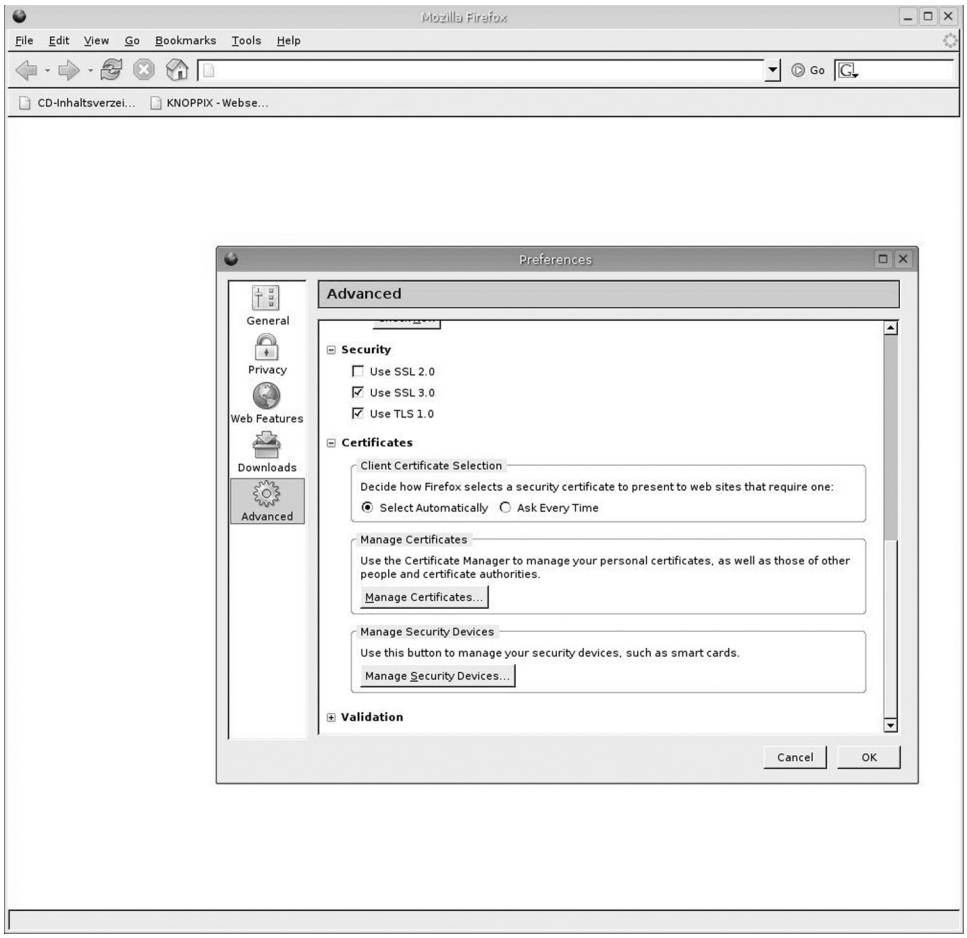
Certificate settings in Internet Explorer 7.

Configuring Support for Cookies

Some configuration settings that can be set on the web browser to mitigate the risk of a loss of privacy due to cookies are as follows:

- Turn off all cookies.
- Limit the websites that can download cookies. The browser can be set to ask the user if any particular cookie should be accepted. In this way, the user can decide in each case if the information put into the browser for

Figure 9-9

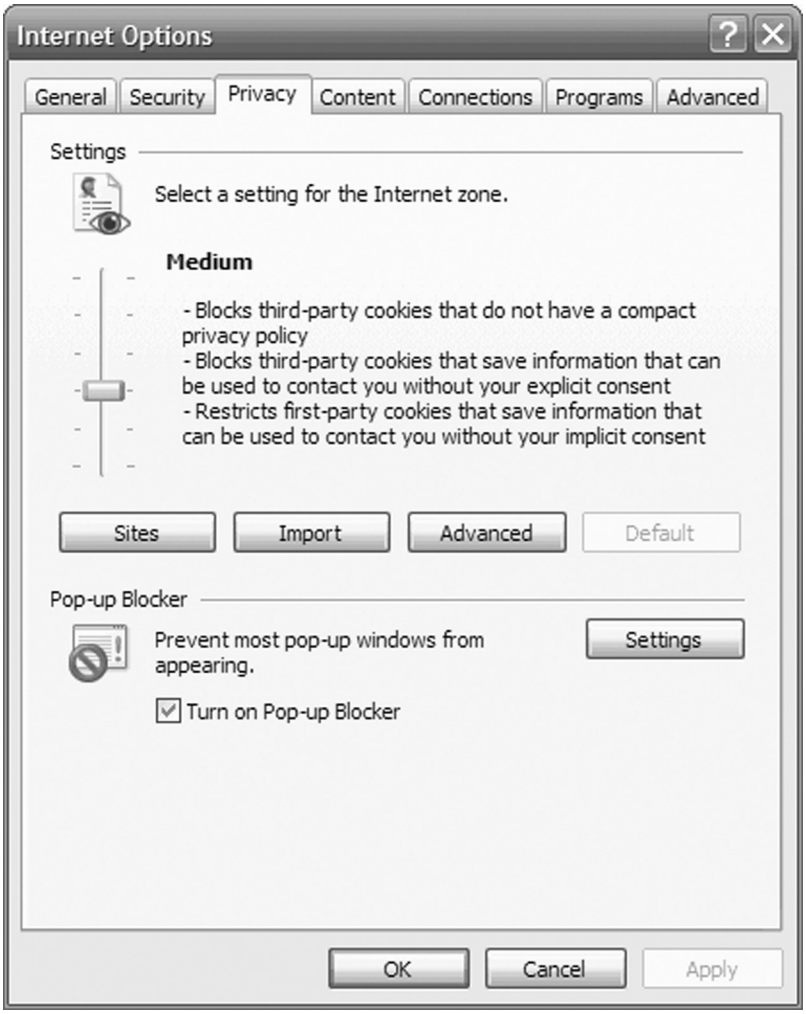


Certificate settings in Firefox.

that particular site poses a privacy risk. In most cases, when prompted to accept or reject a cookie, the user has the option to accept all future cookies from this website.

- Only return cookies to the **originating domain**. Cookies originate (are sent to the browser) from a web server. The browser can refuse to send these cookies back to any website other than the one that created the cookie in the first place. This will mitigate the risk of a third-party site trying to get private data about a user. A cookie that sends data to a different domain is known as a **third-party cookie**.
- Force all cookies to be nonpersistent.

Figure 9-10



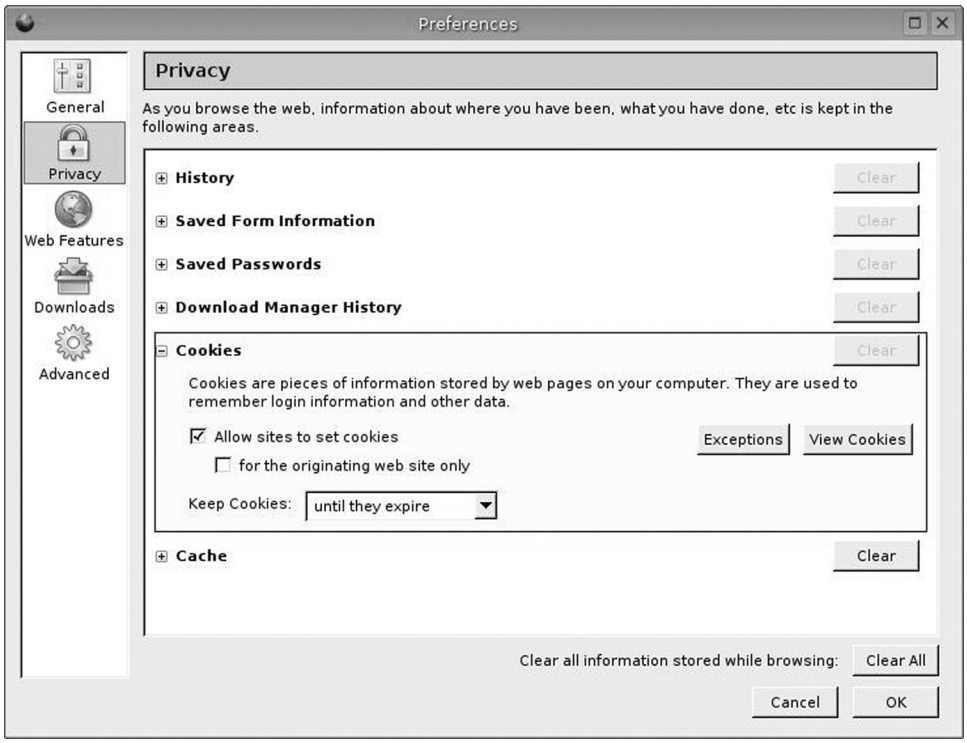
Configuring cookie support in Internet Explorer 7.

▲ Clean out persistent cookies. Periodically, go into the browser settings and delete any persistent cookies.

In Internet Explorer 7, cookie policies are defined on the Privacy tab of the Internet Options dialog box, as shown in Figure 9-10. As you can see, you can set a general policy level, but configure individual sites as exceptions.

Firefox allows you to configure cookies by displaying the Preferences dialog and clicking Privacy. The Cookies options are shown in Figure 9-11.

Figure 9-11



Configuring cookie support in Firefox.

9.3.5 Internet Explorer Security Zones

Internet Explorer orients the Security settings around the web content zone of the site to be accessed by the web browser, as shown in Figure 9-12. In other words, the security settings the browser uses will depend on which zone the website being requested resides in. The zones are as follows:

- Internet
- Local intranet
- Trusted sites
- Restricted sites

Internet Zone

The **Internet zone** contains all websites the user hasn't placed in any other zone. In a sense, this is the default zone. Unless security is relaxed for a particular site, it will be put into the Internet zone and have default security settings. This is one zone to which you cannot add sites. By default, all websites that are not

Figure 9-12

Internet Explorer zones.

added to the local intranet zone, the trusted sites zone, or the restricted sites zone, are placed into the Internet zone.

Local Intranet Zone

The **local intranet zone** is intended to contain all websites that are on the intranet of the user's organization. These sites are considered to be more trusted than those that default to the Internet zone. The local intranet zone contains local domain names, as well as the addresses of any proxy server exceptions you might have configured. To be effective, the local intranet zone should be set up

FOR EXAMPLE

Flaw in Acrobat Reader Plug-in

Many companies and government offices post PDF files on their websites to provide users with information and even to allow users to fill out and print or submit forms. It has long been regarded as a secure way to distribute information on the Internet. However, in December 2006, researchers discovered a flaw that will allow attackers to execute malicious code when a user clicks a PDF document link.

The attack is performed by modifying the link to a PDF document on a website. The modified link includes code that exploits the vulnerability in the plug-in. According to researchers, this vulnerability could be used to create a number of attacks, including installing Trojan horses and accessing data. The flaw exists when the plug-in is used in Internet Explorer 6.0 (and earlier) and in Mozilla's Firefox browser.

What do you do to mitigate the attack? If you are using one of these browsers, you need to avoid opening PDF documents within the browser. Instead, download them to the hard disk and open in them in Acrobat Reader. Or better yet, upgrade your browser to one that does not have the flaw.

in conjunction with a local area network (LAN) proxy server or firewall. The intent is that all sites in the local intranet zone are on the local network and inside the firewall.

Trusted Sites Zone

The **trusted sites zone** contains websites that the user trusts will not damage the computer. The user should also trust this site with sensitive or personal data. The Security settings can require SSL for all the sites in this zone. This zone should rarely be used. Few websites need the added features of this zone. Most web sites that might be put in this zone will probably operate equally well in the local intranet zone. Be cautious when adding sites to the trusted sites zone. If the site is compromised at some point in the future, your computer will be vulnerable.

Restricted Sites Zone

The **restricted sites zone** contains websites that could potentially damage your computer or data. The default security level for the restricted sites zone is High.

9.3.6 Configuring Web Features in Firefox

The Firefox browser allows you to configure feature support for pop-ups, installing software, loading images, Java, and JavaScript. As you can see in Figure 9-13, you can allow pop-ups, installed software, and images from only specific sites.

SELF-CHECK

1. Compare persistent and non-persistent cookies with regards to the security risk.
2. Identify and describe the four security zones in Internet Explorer.

9.4 Email Security

Along with web browsing, email has made the Internet popular, widespread, and indispensable for most users. Despite its critical role in the typical Internet user's life, email is comparatively insecure.

Email is widely used and has well-defined and universally implemented protocols. Therefore, it is a prime target for hackers developing attacks. Attacks on email focus on two areas: the delivery and execution of malcode and the disclosure of sensitive information. In this section we'll look at how to mitigate both types of attacks.

9.4.1 Attacks that Disclose Data

For many years, the popular email protocols **Post Office Protocol (POP3)** and **Simple Mail Transfer Protocol (SMTP)** have transmitted email in clear text. Figure 9-14 shows a captured IP packet from an SMTP session. The text of the email can be clearly seen in the raw packet.

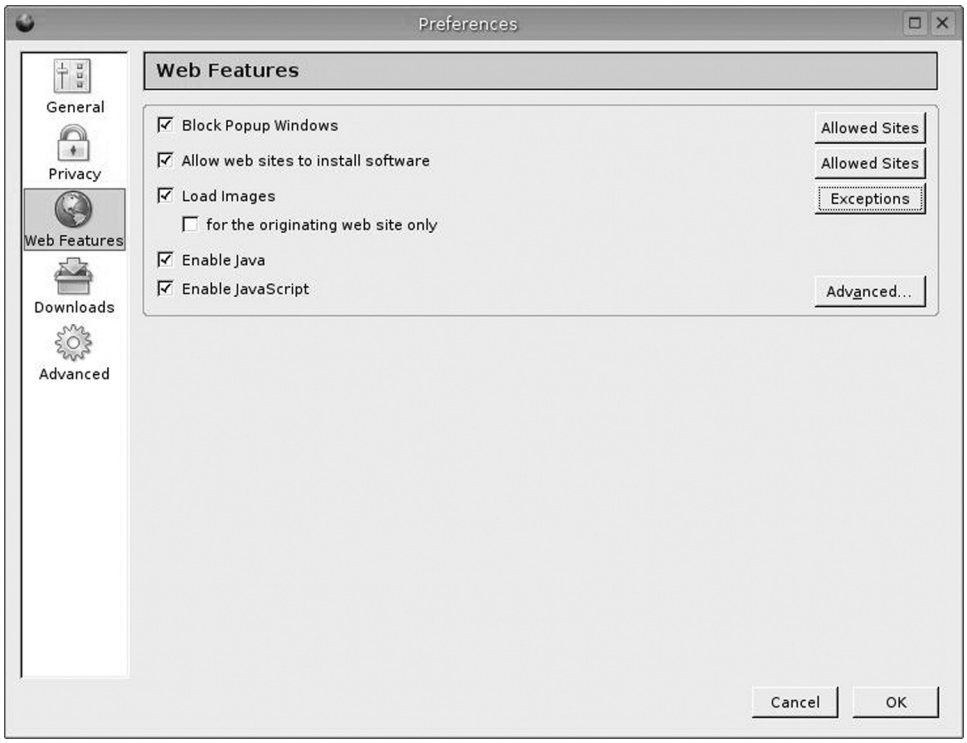
Because email is transmitted in ASCII text, the words typed into an email message are easily viewed and read at the IP packet level. In the preceding sample packet, the text "My passcode is S0nnyB0y" can clearly be read. It is only slightly more difficult to modify the text in the email by modifying the packets.

The capturing and modifying of email can be done via either a man-in-the-middle, replay, or phishing attack. In this section, we'll look at each of these types of attacks.

Email Man-in-the-middle Attacks

In some man-in-the-middle attacks, the attacker must have control of one of the many firewalls, routers, or gateways through which the email traverses. Other man-in-the-middle attacks do not require control of a device; rather, the attacker merely needs to reside on the same local area network (LAN) segment

Figure 9-13

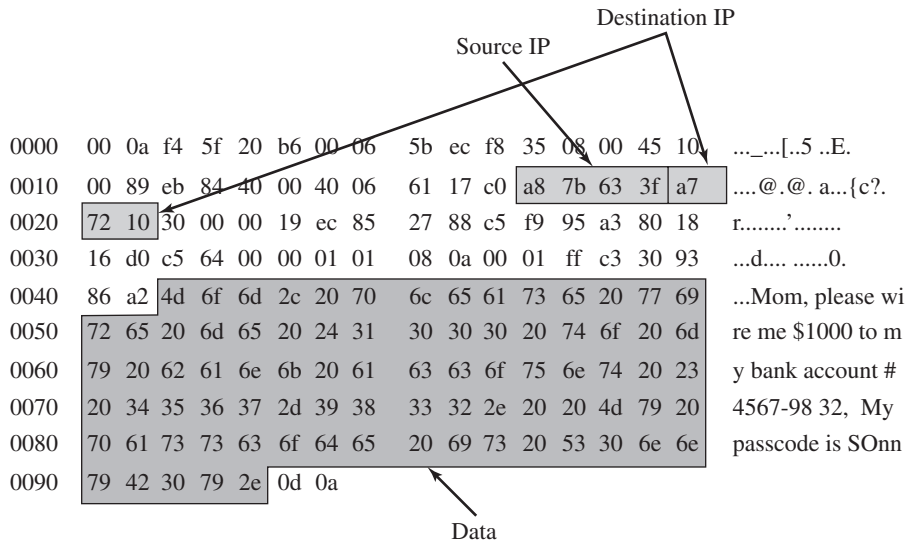


Web Features settings in Firefox.

as one of the computers sending or receiving the email. In this case, the attacker can use an **Address Resolution Protocol (ARP) spoofing tool**, such as **ettercap,** to intercept and potentially modify all email packets going to and from the mail server or gateway. In an **ARP spoof attack**, the attacker gets between any two hosts in the email transmission path. There are four possible locations to attack:

1. **Between the email client and server:** This situation assumes that the client and server are on the same LAN segment.
2. **Between the email client and the gateway:** The gateway must be in the path to the mail server.
3. **Between two gateways:** The gateways must be in the path between the client and the server.
4. **Between the gateway and the mail server:** This option assumes the client and the server are not on the same LAN segment, and therefore the email traffic must reach the server via a gateway.

Figure 9-14



A captured IP packet clearly shows email text.

Figure 9-15 illustrates the network configuration for the ARP spoofing attack. In the ARP spoofing man-in-the-middle attack, the email's IP packets are intercepted on their way to or from the mail server. The packets are then read and possibly modified. The attacker has some minor limitations when modifying the packets. For example, the total length of the packet cannot grow to a size larger than the maximum allowable for transmission on the network.

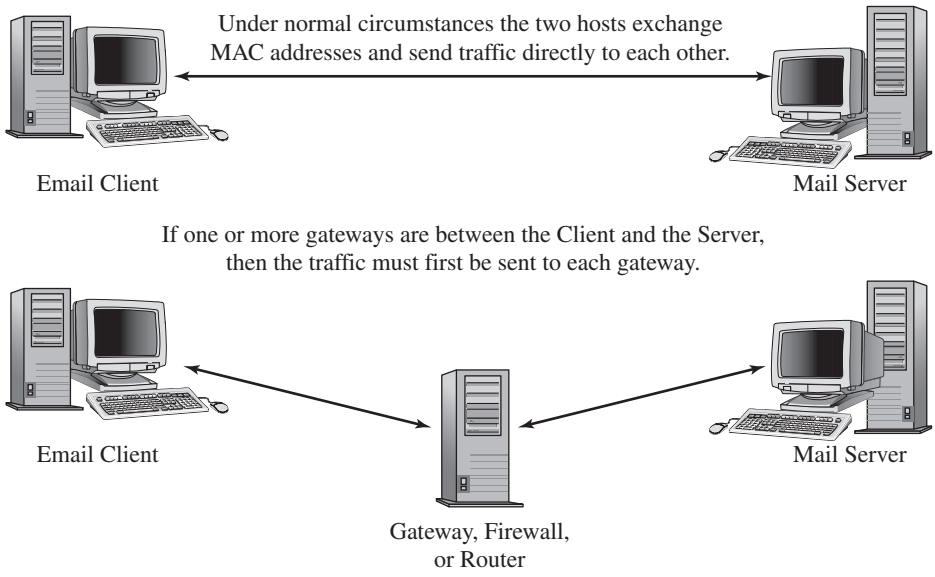
Man-in-the-middle attacks can be avoided by using encryption and by digitally signing messages. If the encryption is sufficiently strong, the attacker will not be able to decrypt and alter the email. Digital signatures ensure the integrity of the body of the email message. The recipient is able to decrypt the hash with the sender's public key and verify the email to have been unaltered. An attacker could not alter the message or the hash (digital signature) without being detected. Figure 9-16 illustrates how a digital signature is created and attached to the email.

Email Replay Attack

An **email replay attack** occurs when an email packet (or set of packets) is captured, the email message extracted, and the message put back on the network at a later time (replayed). This causes a second, identical email to be received. The danger or damage occurs when the second email is accepted as legitimate and causes unforeseen consequences.

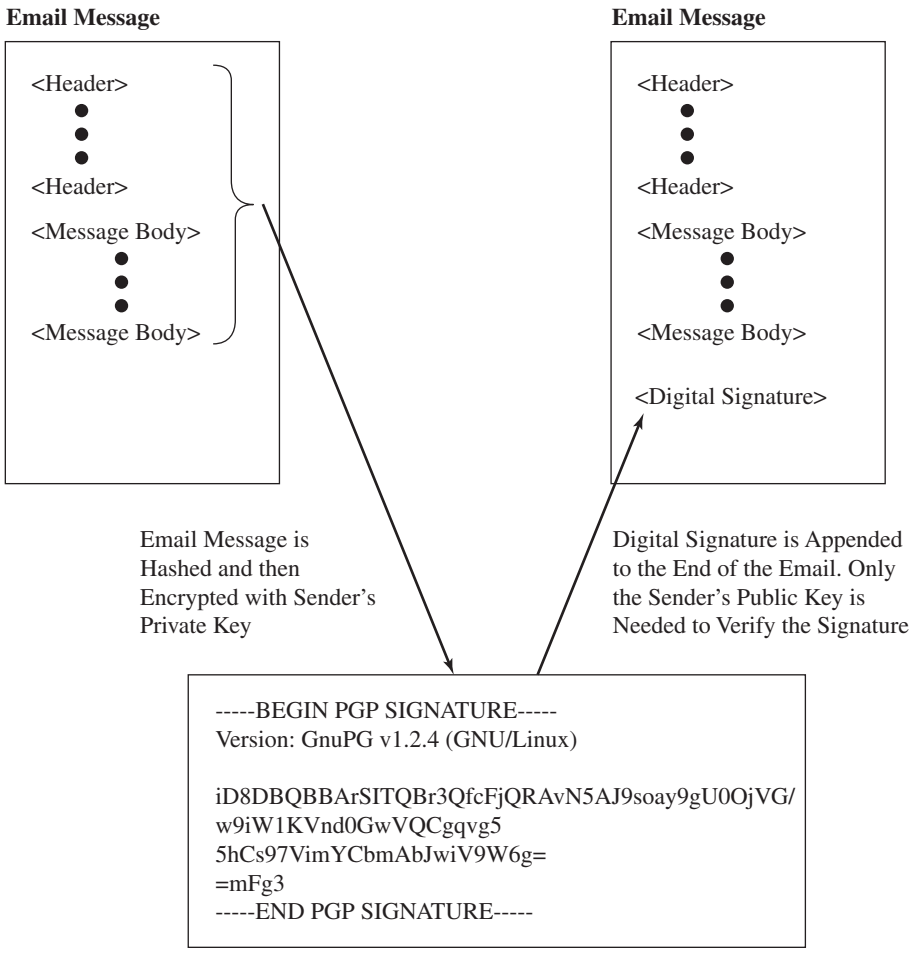
Replay might be used if an attacker discovers a business that sends financial transactions over email. The attacker then arranges for a nominal transaction

Figure 9-15



ARP spoofing attack.

Figure 9-16



Attaching a digital signature.

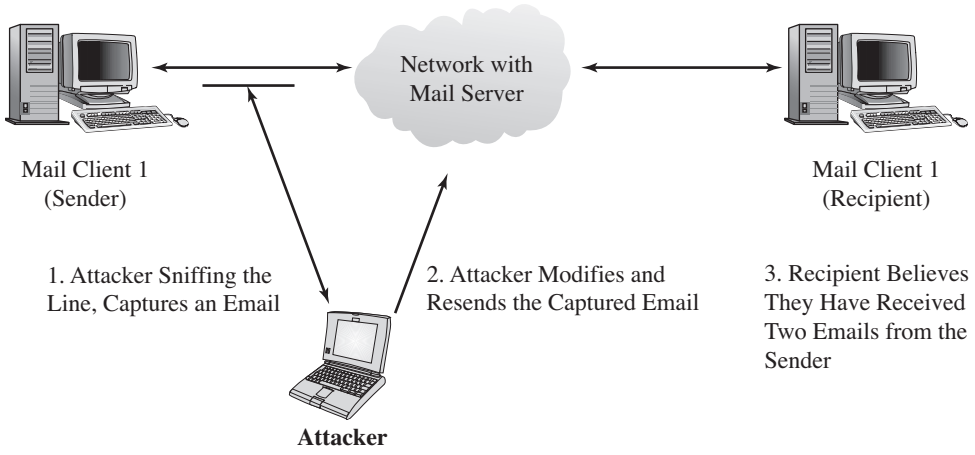
(perhaps a $100 refund). The attacker captures the email authorizing the refund and replays it several times, causing several refunds to occur.

In the case of a replay attack, shown in Figure 9-17, the attacker does not have to use the gateway or ARP spoofing. The attacker merely needs to be on one of the many segments that the email packets traverse on their way to or from the mail server.

Phishing

A **phishing attack** is one in which a user is tricked into clicking a link in an email and divulging confidential information, such as a bank account number or logon credentials for an online banking website. To launch a phishing attack,

Figure 9-17



Email replay attack.

an attacker sends email that pretends to be from a legitimate company, such as a bank, PayPal®, or eBay®. The email includes a link that appears to be to the legitimate site, but that actually goes to an imposter site.

Internet Explorer 7.0 provides a **phishing filter** that attempts to determine whether a site is legitimate. However, the best way to mitigate the risk of phishing attacks is to train users to never click on a link in an email or to verify the actual address of the link before clicking it. Figure 9-18 shows an example of an email sent in a phishing attack. Notice that when you mouse over the link, an address appears that is different from that of the legitimate website.

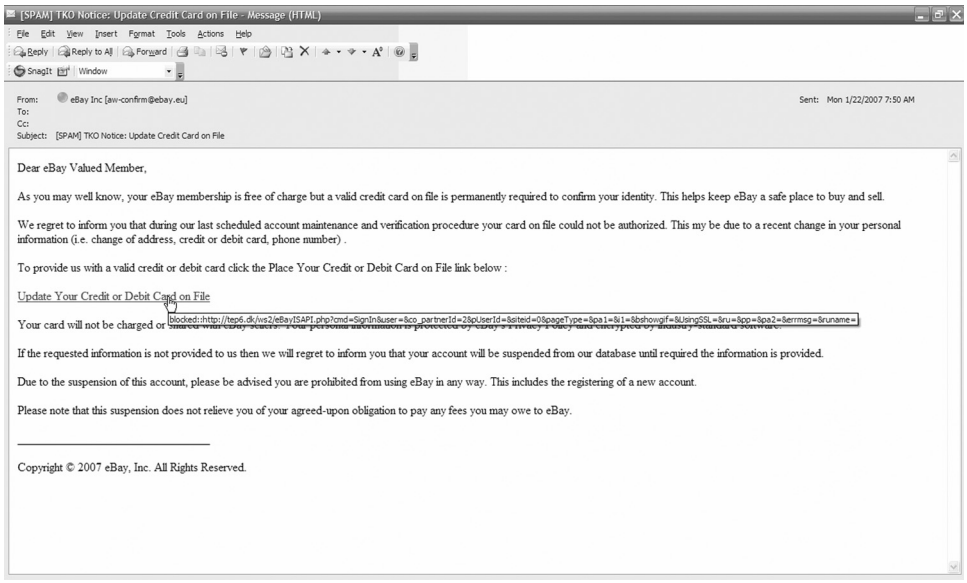
In this case, the spoofed email does a good job of impersonating an actual email that might be sent by eBay. However, notice that the email does not originate from ebay.com, but from ebay.eu. Also, the ISP's spam filter was able to identify the message as spam and marked it as such in the Subject line. We'll talk about spam next.

9.4.2 Spam

Spam is an unwanted email. Spam has become a serious problem in today's networking environment. It is a major irritant and consumer of resources. It has been estimated that for some of the large email providers, over half of the email they service is spam. In gross terms, this means that these providers could get by with half of their current resources to handle their customers' email. From a security perspective, spam is a potential denial-of-service (DoS) problem.

Spammers (people who send spam) make money by getting their advertising message out to thousands or millions of people. Very few will respond positively

Figure 9-18



Phishing attack email.

to the message, but even a very small percentage of responses will produce enough activity to make the spamming profitable because it is very cheap to send email.

Spammers put their advertising message into the body of the email and view email headers as a necessary encumbrance needed to get the body delivered. Spammers view email headers as a possible Achilles heel that can hurt them. If users and ISPs are able to trace the spam back to the source, the spammers could be tied up in legal proceedings, fined, or blacklisted. Blacklisting is discussed a little later.

Spammers take steps to hide their originating (From:) address. This is easily done if spammers run their own email servers. This address can be either fake (such as "yourfriend.spam") or a legitimate address that is not owned by the spammer. Some spam even uses your own email address as the sender.

Spam DoS Attacks

Spam DoS attacks can be launched by spammers using false domains in the emails they send. If a spammer does not use a valid domain, the spam can be blocked by testing that the email was sent from a legitimate domain. In this case, a domain is legitimate if it returns a value when a domain name system (DNS) lookup is done.

The most prevalent DoS attack that occurs due to spam is when a spammer forges an address on thousands or millions of mail messages. The result is tens of thousands of bounces, complaints, and a few responses. This results in a flood of email traffic to the forged address, essentially shutting down the address for legitimate use.

Another DoS situation occurs when the spammer forges a valid email address, and this address then gets blacklisted. When this occurs, the user of the valid email can experience obstacles to sending legitimate email to users whose ISP uses blacklists.

Blacklisting

A **blacklist** is a database of known Internet addresses (by domain names or IP addresses) used by spammers. Often, ISPs and bandwidth providers subscribe to these blacklist databases to filter out spam sent across their network or to their subscribers. Lists of IP addresses to be added to the blacklist are collected in different ways, including the following:

- The email user community sends samples of spam to the blacklist site. The site parses out the offending originating email IP addresses and adds them to the blacklist.
- The blacklist provider runs its own mail server and fake email address. Any email received is automatically unsolicited and therefore spam.
- Blacklist providers exchange lists.

Some blacklists are implemented by placing offending IP addresses in a DNS database. When a spammer's email arrives, a DNS lookup is conducted to verify that the sender's email address is legitimate. However, blacklisted addresses return invalid responses so the server rejects the email.

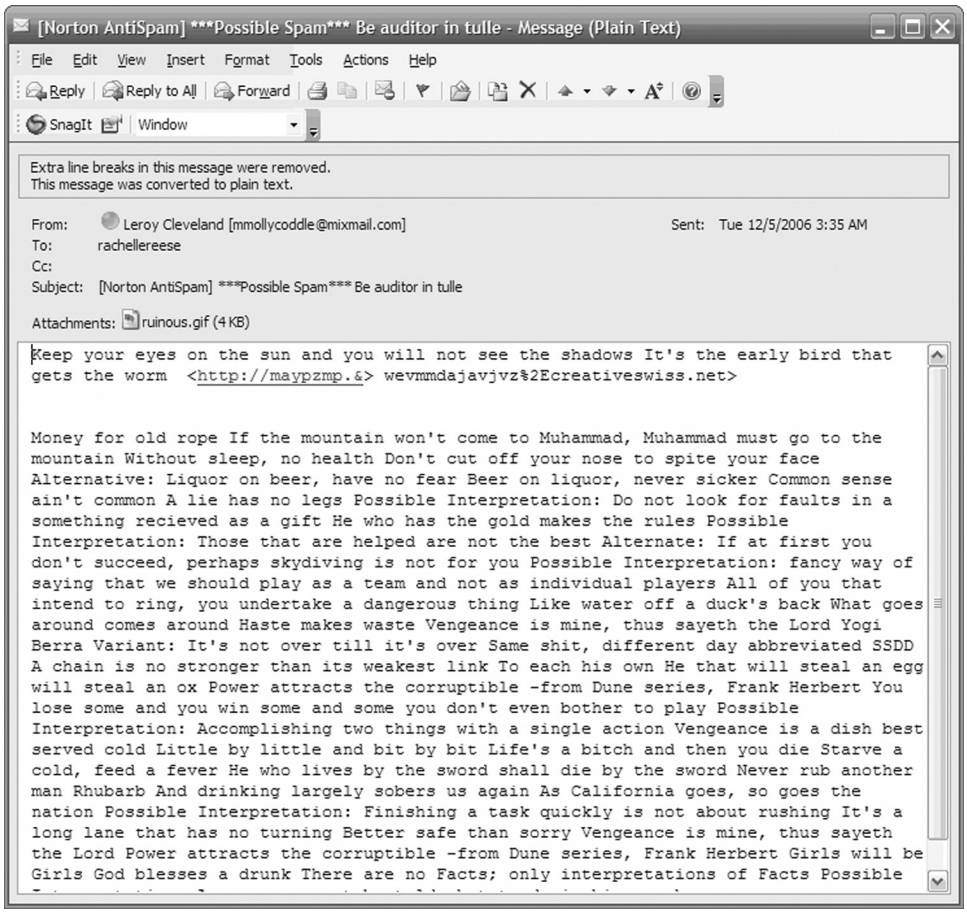
Spam Filters

Spam filters attempt to identify spam from the content of the message subject and body, based on identifying words that frequently appear in spam. This is known as the **naive Bayes classifier**. However, this method of classification often results in non-spam being classified as spam. Also, spammers have begun to use various methods of preventing spam filters from working. One method is to insert **word salad** (a set of random or pseudorandom words) into the text of the message. Another is to use **letter salad** to disguise words in the subject that are frequently associated with spam. For example, here are two letter salad subject lines:

- Why seek? Choose any love pi11 you want!
- Re: primar VIAttGRA

Figure 9-19 shows an example spam email that uses both word salad and letter salad. It includes an attachment, which should not be opened under any circumstances. In this example, the ISP did not identify it as spam, but the Norton AntiSpam™ application did. This is a good illustration of why multiple layers of defense should be used to protect against malware.

Figure 9-19



Sample spam.

9.4.3 Protecting Against Malcode Propagated by Email

Most malcode transmitted by email is not activated unless a user opens the email or the attachment. Therefore, user education can go a long way toward thwarting such an attack. The following are some guidelines users should keep in mind:

- ▲ Be paranoid. You can avoid most email-propagated malcode attacks by properly using your email.
- ▲ Keep your email address private. Avoid providing it whenever possible on websites and other interactive forums such as chat rooms.
- ▲ Set up one or more sacrificial email addresses. When an email address must be provided to a website, the user should have a sacrificial email

address to use. When an email is received on this account, the user knows that there is a high likelihood that it will be spam or malicious in nature. The user should resist the temptation to browse through the emails received on this account.

- Keep email for different organizations separate. In most cases, this will mean one account for work and a separate account for home. The ramifications of receiving and propagating malicious code in a work environment might be more damaging than at home.
- Do not open any email that is not expected. Common sense can be a strong deterrent to the spread of malicious code. An unexpected "Read This" or "Try This Game" should be ignored until the user can verify the sender's intentions. The verification can be in person, by phone, or by a second email (initiated from the user).
- Never save or open attachments from strangers. All curiosity must be resisted.
- Never save or open attachments that are not absolutely needed. The fact that a friend wants to send a user an attachment does not obligate the user to open it. Some users would be surprised to find how easily life proceeds without opening risky emails and attachments. If it is really important or of special interest, the friend will follow up and explain what is in the attachment.
- If supported, turn off the preview function on your email client.

9.4.4 Mail Client Configurations

Microsoft Outlook® uses the same security zones as Internet Explorer to allow users to customize whether scripts and active content can be run in HTML messages.

Scripting capabilities of the email clients should be disabled whenever possible. As discussed earlier, if scripts are executed by the email client, the user will be vulnerable to worms and viruses. If scripts must be passed around, they should be compressed when they are sent, saved to a file, and examined before being executed.

You can also determine whether elements in HTML pages, such as pictures, are downloaded automatically. The default is to not download automatically unless the site is included in the trusted sites zone or if the sender has been added to the Safe Senders or Safe Recipients list, as shown in Figure 9-20. You should not turn on automatic download for all senders.

Some mail clients, including Outlook and Outlook Express, are configured to block attachments with risky file extensions. This protection should be kept enabled and you should add file extensions to the list as attacks emerge.

You can configure email encryption and digital signatures to protect the confidentiality and integrity of the messages you send. These settings are shown in Figure 9-21.

Figure 9-20

Automatic Picture Download Settings dialog box.

9.4.5 Architectural Considerations

A number of system- and network-related architectural considerations ensure safe use of email:

- Check for viruses. Every computer should have virus protection installed.
- Use a mail relay or mail proxy. Medium- to large-size organizations benefit from having all their mail received first by a **mail relay** or **mail proxy**. The mail relay will usually sit in the perimeter network. If configured properly, the relay can check for unwanted scripts, viruses, and questionable attachments. Mail relays are also a good place to put spam protection, such as blacklist monitoring and spam filtering.
- Buffer against attacks. If possible, risky email should be read on computers that can better afford to be attacked. Generally, this would be a computer that has little or no personal and sensitive data. This computer also should not contain critical applications that can't be lost or re-installed. It should

Figure 9-21

Outlook security settings.

be expected that a computer that is buffering this way might have to be rebuilt every 3 to 6 months.

- Back up frequently. Even the best security measures will occasionally fail to stop a new and emerging threat. To minimize the impact when that happens, backups should be done frequently. The frequency of backups depends on the level of critical data involved. A book author will back up a few times a day, although the typical home user might get by with backing up once a week or once a month.
- Control scripting capabilities. Some mail clients will provide collaboration capability and run scripts automatically. Usually, this feature can be disabled to reduce the risk of worm and virus attacks.

FOR EXAMPLE

Microsoft Exchange 2007

In response to the increasing prevalence of spam, phishing schemes, and malware distributed through email, software developers are creating solutions that can help protect against email-propagated threats. One such enhancement is the **Edge Transport** server role in Microsoft Exchange 2007. When a server is installed as an Edge Transport server, it can perform spam filtering based on the sender's reputation and usage patterns, the safe sender lists compiled by users, content filtering, and an Outlook postmark. The **Outlook postmark** is a puzzle of varying complexity that is attached to the email. The Edge Transport server can also implement virus scanning and attachment filtering, including examining the contents of a .zip file, to determine whether email contains file types that should be filtered.

The Edge Transport role is meant to be assigned to a dedicated server on the perimeter network. It does not need to be a member of the Active Directory domain because it can receive encrypted directory data through **Active Directory Application Mode (ADAM)**.

The Edge Transport server can quarantine suspect messages, allowing the administrator to selectively send them to users or even to convert them to plain text before sending them.

- Limit attachments. Attachments can contain scripts and executable code. When the user runs these scripts or executables, they will have all the privileges and access that the user enjoys. Unless the user is diligent and fully appreciates the risk, it is not safe to allow attachments on email.
- Quarantine attachments. In many cases, an organization can benefit from **quarantining attachments**. A mail relay or mail proxy strips attachments off of emails before they are delivered to users. If the user needs the attachment and can verify that it has been sent by a legitimate sender, the user can recover that attachment.

SELF-CHECK

1. Compare ARP spoofing and replay attacks.
2. Compare blacklisting and spam filtering.

SUMMARY

In this chapter you learned about various types of malware, including viruses, worms, Trojan horses, and spyware. You also learned how to protect yourself against these threats. Threats, such as spam, phishing attacks, man-in-the-middle attacks, and replay attacks were also covered.

KEY TERMS

Active Directory Application Mode (ADAM)
ActiveX
ActiveX control
Add-on
Address Resolution Protocol (ARP) spoofing tool
Adware
Anti-malware
Anti-spyware
Antivirus
ARP spoof attack
Autorun macro
Backdoor
Blacklist
Boot sector virus
Browser parasite
Compiled
Cookie
Cyclic redundancy check (CRC)
Distributed denial-of-service (DDos) attack
Edge Transport
Email replay attack
Ettercap
Host
ILOVEYOU virus
Internet zone
Java
Java applets
JavaScript
Letter salad
Local intranet zone
Logic bomb
Macro Virus Protection
Mail proxy
Mail relay
Malcode
Malware
Managed computer
Melissa
Michelangelo
Naive Bayes classifier
Nonpersistent cookie
Nyxem worm
Originating domain
Outlook postmark
Persistent cookie
Personal firewall
Phishing attack
Phishing filter
Plug-in
Portable Document Format (PDF)
Post Office Protocol 3 (POP3)

Propagated
Quarantining attachments
Replay attack
Replicate
Restricted sites zone
Rooted
Safe for execution
Safe for scripting
Scripts
Security manager
Self-propagation
Session cookie
Session hijacking
Sessionless
Simple Mail Transfer Protocol (SMTP)
Slag code
Spam
Spam DoS attack
Spam filter
Spammer
SpyBlast
Spyware
State
Third-party cookie
Time bomb
Trapdoor
Trojan horse
Trusted sites zone
Virus
Virus signature
Visual Basic script (VBScript)
Web beacon
Word salad
Worm
Zombie

ASSESS YOUR UNDERSTANDING

Go to www.wiley.com/college/cole to assess your knowledge of protecting a computer against viruses, worms, and other malicious programs.

Measure your learning by comparing pre-test and post-test results.

Summary Questions

1. Which of the following requires a host file to propagate?
   (a) worm
   (b) spyware
   (c) virus
   (d) logic bomb
2. Which of the following uses the autorun macro to attach itself to the Normal.dot file?
   (a) Michelangelo
   (b) Melissa
   (c) Nymex worm
   (d) SpyBlast
3. A logic bomb can be used to launch a DDoS attack. True or false?
4. A browser parasite is an annoyance, but it cannot do any actual damage. True or false?
5. SpyBlast is an effective anti-spyware program. True or false?
6. An effective antivirus program with updated signatures is the only protection you need against viruses and worms. True or false?
7. Which of the following is a computer that is centrally configured through an automated policy?
   (a) managed computer
   (b) rooted computer
   (c) host
   (d) zombie
8. Which statement best describes the dangers of automatically downloading graphics in an HTML message?
   (a) The graphics might contain macros that will perform a malicious task.
   (b) The graphics might be web beacons.
   (c) The graphics might be Trojan horses.
   (d) There is no danger involved.
9. An ActiveX control that is marked safe for execution has been certified by Microsoft not to do anything harmful. True or false?

10. Code written in JavaScript cannot access any file on the hard disk. True or false?
11. What type of cookie sends data to a different website than the one from which it originated?
    (a) nonpersistent cookie
    (b) persistent cookie
    (c) session cookie
    (d) third-party cookie
12. Which Internet Explorer zone contains any computer not included in the other zones?
    (a) internet
    (b) local intranet
    (c) restricted sites
    (d) trusted sites
13. What type of attack can be mitigated by using digital signatures?
    (a) ARP spoofing
    (b) email replay
    (c) phishing
    (d) spam DoS
14. Which type of attack relies on social engineering techniques?
    (a) ARP spoofing
    (b) email replay
    (c) phishing
    (d) spam DoS
15. Which of the following attempts to identify spam by looking at the content of the message?
    (a) blacklist
    (b) anti-malware program
    (c) spam filter
    (d) web beacon
16. A mail proxy should be installed in the perimeter network. True or false?

Applying This Chapter

1. Busicorp is concerned about the possibility of malware propagating through the organization. They are currently using their ISP to manage email. Users connect to the Internet using a shared Internet connection

on the company network. Some users have laptop computers and also connect to the Internet from home or when traveling. There are three file servers on the network. You have been asked to devise a plan for mitigating the risk of malware.

(a) Identify the potential sources of viruses.

(b) What questions should you ask the ISP?

(c) What additional protections would be offered by an anti-malware program that are not offered by an antivirus program?

(d) Why might you want to limit ActiveX controls to only trusted sites?

(e) What benefits are provided by issuing digital signatures to be used on email?

(f) Why should security awareness training be a necessary part of the plan?

(g) Describe how the company could be a victim of a Spam DoS attack.

YOU TRY IT

Recognizing Malware

Understanding how to identify a risky download, attachment, or phishing email is an essential part of mitigating the threat of malware. Automated scanners have limitiations—they can only identify known attacks. Identifying new or unpublished attacks requires a sharp eye and a keen nose for trouble. Users can develop those over time, but they need training. Of course, before you can train users in what to look for, you have to know yourself. Think about the following situations and determine whether the action is safe, moderately safe or moderately risky, or risky. Explain why.

1. You access an online shopping site. A dialog is displayed that reports the site's SSL certificate has expired. How risky is it to provide your credit card on this site?
2. You access an online shopping site. A dialog is displayed that reports that the site's SSL certificate was not issued by a trusted certificate authority. How risky is it to provide your credit card on this site?
3. You access a vendor's website and a yellow bar appears asking you to download an ActiveX control. When you attempt to install it, you receive an error that the control is not signed. You have been working with this vendor for a few years and have not had a problem. How risky is it to download the control?
4. You receive an email from your bank asking you to verify your address and phone number. The email contains a link with a different domain name than your online banking site. When you click the link, you are prompted for a username and password. How safe is it to enter the information?
5. You receive an email from a former business acquaintance who you haven't heard from in several years. The subject of the message is Hello. The message contains an attachment. How risky is it to open the attachment?
6. You have antivirus software installed, but you are connecting to the Internet through a dial-up connection until your broadband service is restored. How risky is it to ignore the virus signature update message?
7. An online training website uses a nonpersistent cookie to track your progress in a session. How risky is it to accept the cookie?
8. You are creating a website for your business and need to publish your email address so that customers can contact you. How risky is it to use your regular email address?