

## PART TWO

# SECURITY FOR INDIVIDUALS

CHAPTER 1. INTRODUCTION TO SECURITY FOR INDIVIDUALS

CHAPTER 2. UNDERSTANDING AND ADDRESSING SECURITY

CHAPTER 3. KEEPING YOUR COMPUTER AND DATA SECURE

CHAPTER 4. KEEPING YOUR OPERATING SYSTEM AND APPLICATION  
SOFTWARE SECURE

CHAPTER 5. MALICIOUS SOFTWARE

CHAPTER 6. SECURING SERVICES OVER NETWORKS

CHAPTER 7. TOOLS TO ENHANCE SECURITY

CHAPTER 8. PLATFORM SPECIFIC ISSUES  
ADDENDUM 1. INTRODUCTION TO  
ENCODING AND ENCRYPTION

ADDENDUM 1. INTRODUCTION TO ENCODING AND ENCRYPTION

ADDENDUM 2. TCP/IP

ADDENDUM 3. MINI-GLOSSARY OF TECHNICAL TERMS

## CHAPTER 1. INTRODUCTION

Part 2, Security for Individuals, is aimed at all computer users, from novices to experts and should serve as a primer on how to use your personal computer safely. Safe computing is possible, but it takes knowledge, vigilance, and care. The language in this section will include a certain degree of technical jargon; in general, some technical terms are defined in the mini-glossary at the end of Part 2; they also appear in the full Glossary in Annex 1 of this Handbook.

The first step in devising a security strategy is to understand what “safe computing” means. If you practice safe computing, you are seeking to ensure that:

- your data and programs will not be altered or disappear unless you request it;
- your computer and programs will behave the way their designer intended (with the exception of software bugs which are unintended flaws in program code);
- no one will use your computer, your data, or your network without your permission;
- you will not unknowingly spread computer viruses;
- you will not be annoyed *as much* by unwanted advertisements (spam);
- no one will watch every move you make on *your* computer;
- no one will capture any of the data that goes over your wired or wireless network;
- no one will steal your usernames or passwords on systems or sites that you access;
- if you enter credit card numbers or bank account information online, the data will be reasonably secure (at your end of the connection; obviously you have less control over what happens at the other end of the connection).

In the personal computer context, if you ignore security issues, the results can range from annoying but costless, to time-consuming and expensive. In a professional computing context, the problems caused by unsafe computing could jeopardize your business. In either case, someone must take responsibility for assessing the risks, developing a security plan, and executing that plan. Even with detailed knowledge of information technology (IT) security issues, you will not be able to control all aspects

of your computing environment. However, if you follow the guidelines in this Handbook and survey the resources available to you, you will be able to minimize the risks and develop appropriate responses to the evolving world of information technology.

Covering all of the issues related to security for individuals would take hundreds of pages. Most people do not have the interest or time to read such a complete study. This summary provides the information required for a typical user to understand and implement a reasonable degree of security on his/her computer. At times, the material presented in this Handbook may over-simplify some of the more complex issues. The bibliographies provided in the Annexes offer references to print resources, electronic resources, and organizations that will aid the user in further study of IT security issues.

## CHAPTER 2. UNDERSTANDING AND ADDRESSING SECURITY

### At a Glance

This chapter evaluates why computer and network security are necessary. It addresses the impact of security breaches and it assesses the initial measures required to counter such breaches. The chapter also includes a list of definitions of technical terms; additional terms are defined in Annex 1.

### Why are Security Measures Required?

In the early days of computing on shared systems, there were usernames, but no passwords. Passwords were added once the first malicious (or curious) users began to abuse the ability to logon via username only. Today, there are a number of reasons to think about computer and network security:

- The value of your investment in hardware equipment and software programs. Computers and software packages are expensive. Replacing them may be costly and difficult. Even if you do not lose the actual hardware and software, security problems can require a re-installation of all software programs and then re-configuration to meet your specific needs. This can be time-consuming, if not impossible, for someone with only a moderate degree of technical knowledge.
- The value of your business data. This data could include your customer lists, financial projections, or proprietary programs that you have written.
- The value of your personal data. Your personal data may not have any clear monetary value, but a *loss* could be expensive (see later definition of identity theft), and you should consider how much time it may take to recreate the information.
- The threat of computer criminals. As technology has advanced, a class of people who take advantage of networked computers to steal data has emerged. In some cases, they are operating for benign (or malicious) kicks or to prove to themselves or to their friends that they can do it.

In some cases, they are operating for personal gain (stealing credit card information, engaging in fraudulent transactions). In any case, these people can cause inconvenience and damage; in extreme cases they may create serious problems for individuals and businesses whose data has been compromised. Since the Internet is available to users worldwide, it can be complicated, if not impossible, to trace where the attacks are coming from and to stop the intruders permanently.

### Why is security lacking?

Software programs are often developed without a focus on securing them. This happens for several reasons:

- o Ignorance – the programmer or designer did not know about the need for security;
- o Low priority – until recently, security issues did not have the visibility that they now do. As a result, even people who knew about security issues chose to ignore them;
- o Time and expense – some people think that it is more expensive and time consuming to design, code, and test for security issues during the software development process; and
- o Sloppiness – in some programming efforts, the same mistakes are made repeatedly, some of these mistakes make security breaches possible.
- People are innovative and motivated individuals will find ways to circumvent security or to discover errors that create security exposures.
- Normal users (potential victims of security breaches) are not sufficiently aware of the threats around them and do not make an effort to follow proper procedures for securing their data and their systems.
- Some users may be aware of security issues, but simply do not take them seriously – they assume that an attack will not be launched against them.

## Assessing the Threat and the Cost of Loss

In order to understand how important security is to you, you may wish to consider a number of “what if” questions. Imagine each of the security incidents listed below and try to assess the likely results of the incident.

The key questions that you must answer are:

- Could you recover from the incident?
- How much time would it take?
- How much money would it cost?
- How would it impact your business?
- What hidden costs would there be (including loss of status or authority)?

Here are a few possible security incidents:

What if...

... someone broke into your home or office and stole your computer. For added impact, they might also take the backup disks found near the computer.

... all of the data on your machine was erased?

... all of your data was stolen. This data might include: your bank account information, a list of your usernames and passwords for web sites where you make online purchases, an important report that you are writing for work, or a school assignment that is due tomorrow and is worth 50% of the course grade.

... someone watched and memorized everything that you were doing on your computer? When you type a credit card number, they know it. When you browse a web site, they know it. When you log onto a web site or system, they are able to capture your username and password.

... your computer kept crashing when you were working on an important, time-sensitive project?

... you sent a malicious computer virus to everyone in your address book?

... your telephone bill arrived and showed that you owe the phone company more than your monthly salary for calls that you did not make?

... you received a bill for a credit card that you do not own, but the bank issuing the card is convinced that you applied for it. (And they have proof of “your” application.)

All of these situations highlight why computer security is important. Once you understand that security is important to you, the next step is to assess what a good security plan will entail:

- Will it cost you anything to implement security measures?
- How much time will it take?
- How inconvenient will it be?
- Are there things that you like to do on your computer that will become difficult or impossible?
- Can you put the security measures in place yourself or will you need help from others?

These are important questions because you need to approach security with a solid understanding of the costs in terms of money, time, and inconvenience. Without this knowledge, you might become discouraged in the process of securing your system and perhaps you would abandon the project, leaving yourself unprotected.

### Will it cost you anything to become secure?

Many of the paths to good security do not require specific products and those available commercially are fairly inexpensive. Even virus-checkers, the most common purchased security product, are available as freeware. Some organizations that offer freeware products are listed in the Annexes.

### How much time will it take?

You will need to devote some time to implementing and following security measures, although this commitment should not be overwhelming. In short, you will need to install the proper software and perform some routine maintenance tasks on a regular basis.

### How inconvenient will it be?

How inconvenient it will be depends on your point of view. In a security mindset, you have to think

about what you are doing and you will not presume that everything is safe. For example, if someone sends you an attachment, you will decide whether you should open it or not. However, this level of caution is taken in other aspects of life. It is more convenient to cross a street whenever and wherever you wish. Nevertheless, in many places, it makes sense to check that there are no cars coming before you step into the road.

### **Are there things that you like to do that will be difficult or impossible?**

Yes, you will have to modify your actions to some extent. Opting for increased security will prompt you to be conscious of potential problems and to avoid them whenever possible. Contemporary software packages have many attractive capabilities, however, using certain features, especially those that enhance networking and messaging, can make you vulnerable to attack. For example, you might find a web site that offers a service that you want to use. However, to access the service, you must allow it to download and run a program on your computer. If you are not sure that the people who operate the service are trustworthy, it may be better not to download the program.

### **Can you put the security measures in place yourself or will you need help from others?**

In theory, you can be fully responsible for all aspects of security, but in practice, it may be better to share the responsibilities with others.

- Updating software programs and patches, a necessary part of being secure, is often bandwidth intensive. For someone connected to the Internet with a link running at megabit speeds, this is not a problem. However, in developing countries, bandwidth is often severely restricted and sometimes very expensive. Dialup connectivity, while sufficient for downloads, may result in high costs for connections of a long duration. It may be better to have one person download updates for common software and then to distribute copies locally. Unfortunately, this is often not as convenient as having each user work directly online.

- Many security alerts are aimed at the computer professional (although this is changing as the world becomes more security-conscious). A novice user may not know how to access these alerts. If a new user does receive the alerts, he or she may not be able to understand them or take appropriate actions in response to them. Occasionally, you may receive malicious spam claiming to be a security update from Microsoft that contains an “update” attachment. The mail, of course, is not from Microsoft and the attachment is typically a dangerous virus.
- In environments where there are a large number of machines (businesses, schools, government offices), it makes sense to have a system administrator handle some aspects of security.

If you do choose to share the tasks of securing your systems with others, you should put a good communication plan in place. More information will be provided on systems administration in other parts of this Handbook. However, assigning clear responsibilities for security procedures to a designated individual or group of individuals is an important part of the security plan.

### **Deciding on a personal security plan**

There are many programs that address a range of computer security needs. Once you understand the threats and decide on what kinds of risks you would like to minimize or eliminate, you can take steps to put a personal security plan in place. After assessing the issues of cost, time, and inconvenience, you may decide that there are some types of threats that you will live with, at least for the time being. Your security plan will rely, to a certain extent, on software programs, but it should also include procedures, rules, and self-discipline.

Good security is a result of multiple barriers or layers. Each layer will stop certain kinds of threats. If you use a variety of barriers, you will be more successful in eliminating a variety of problems. You can use the analogy of driving a car; what do you need to do to reduce the chance that you will have an accident? Some of the techniques are:

- Keep the car in good repair;
- Drive carefully;
- If the manufacturer alerts you that there is a safety-related defect in the car, get it fixed quickly;
- Pay attention when you drive, as other drivers may cause problems for you;
- If you read in the newspaper that a bridge is broken, do not drive over it.

None of these techniques alone will keep you safe, but by employing all of them, you will be more likely to avoid an accident. In developing a good security plan, one must take a number of partially redundant steps. Consider how you might protect a valuable piece of jewelry. You keep the jewelry in a locked box, inside your locked house, and you have an insurance policy that will replace the jewelry if it is stolen. So you have several levels of protection. Any one of these *in theory* would protect you from loss, but it is wiser to take all precautions. That way, if one of the methods should fail (perhaps there is an untrustworthy workman in your house – so the locked door will not help), there are still safeguards in place.

The principle that needs to be understood is that virtually all security techniques can and will fail occasionally, either due to design problems, poor implementation, or human error. This applies to tools such as virus checkers, encryption and passwords. Any tool may fail at times and you should never rely on a single method to save you from disaster.

## The Role of the User in Security

The primary user of a computer clearly has a large role to play in ensuring that the computer and its software are set up with a good degree of security. In addition, other users of that computer also have a role to play in ensuring that safe computing practices are followed carefully. As we will see, one of the greatest threats to safe computing is a user who does not understand or is not sufficiently diligent about security.

## Security is an Art, not a Science

There is nothing guaranteed in trying to secure your computer and network. There are always new bugs, new forms of attack, and new opportunities for breaches that

arise from human error. However, if you study and follow a set of *best practices* in security with diligence and care, you are improving your chances at operating your system securely. It also helps to stay current with the field through web site research and the mailing lists of respected computing organizations, some of which are listed in the Annexes. Such research may help guide your security practices, particularly when new or unusual circumstances are present.

## CHAPTER 3. KEEPING YOUR COMPUTER AND DATA SECURE

### At a Glance

This chapter investigates ways in which you can keep your computer physically secure and ensure that its programs and data are protected from loss. Topics include physical security, backups, and authentication through the use of usernames and passwords.

### Introduction

One of the best ways to master the concepts of information security is to take a rules-based approach. Starting with Physical Security, the next few chapters in Part 2 will take you through the basics of setting up security procedures for your personal computer or those of your colleagues, if you work in a small group. Information on the technical aspects of security for larger organizations or more experienced users is featured in Part 5 of this Handbook. If you are comfortable with the concepts introduced here, you may decide to build on your knowledge by consulting Part 5 – Security for Technical Administrators.

### Physical Security

The first step is to ensure that your computer is physically secure. This may be a trivial or non-trivial exercise, depending on what you own, where it is kept, and how critical the computer and data are.

#### Computer Theft

Computer theft is a growing problem. Computers, particularly laptops, are often very easy to steal and difficult to recover. If the thief is not interested in using the computer himself, there is a strong market for used computers, stolen or otherwise. Some thieves do not even bother to steal the entire computer and monitor, but will take certain parts, perhaps the memory or the processor. Both items are marketable, simple to conceal and transport, and very difficult (if not impossible) to trace.

#### Rule 1: Think about computer theft *before* it happens.

Having your computer stolen is certainly inconvenient. It may also be expensive if you have no (or insufficient) insurance. In some cases, the loss of data could expose your business or personal secrets to others. In extreme cases, a stolen computer could put you out of business. Fortunately, by following a number of simple and inexpensive measures, you can dramatically reduce the chance that your laptop or desktop will be stolen. There are two main preventive techniques: make your computer difficult to steal and/or make it less desirable for those who would want to use it.

#### Make it difficult to steal and access

There are several ways to prevent a thief from taking your computer:

- Ensure that the place where you keep the computer is secure. It can be locked up in a room or it can be watched by your colleagues, if you work in an office with many employees. Don't leave your computer unattended in public places such as airports.
- Use an alarm system, if it is likely that someone might break into your office at night, for example, when no one is around.
- Consider securing the computer to a desk or pipe or other immovable object using heavy wire cable or chain. This method is often used in semi-public areas such as libraries or schools. Many computers have a convenient place to connect such a tie-down. Virtually all laptops have a connection point for a security cable; special cables and locks are sold for them.
- If the computer has a lock to prevent the case from being opened, use it. You can also buy special screws that cannot be undone easily.
- If there is potentially valuable information on your computer (business data, personal information), you should consider restricting logical access to it when you leave it in hotel rooms or other unattended locations. *Logical access* means actual use of the computer once you have physical access to it.

Robust logon passwords and password-protected screen-savers are a good start in this direction. (See the section on Authentication later in this chapter).

- Laptops and PDAs (Personal Digital Assistants) are small and easily lost. Get in the habit of putting them away immediately when they are not in active use.

#### **Make it less attractive to take**

Few people will want to buy a used computer if it is obvious that it was stolen. A simple and inexpensive way to make it less attractive to would-be purchasers is to identify your property with non-removable tags or mark the equipment with paint. The markings can include your name or other identifying information. If you use this method, do not get any paint in ventilation slots or other openings. Be aware that marking the computer case can void your warranty.

#### **Computers are delicate**

Computers are particularly sensitive to dust and rough handling. If you operate computers in dusty environments, they should be cleaned regularly, with extra care taken that ventilation openings are not blocked. Computers are also sensitive to drops and bumps.

#### **Other aspects of physical security**

If you open up your computer to install new hardware, don't ignore the warnings about reducing electrostatic shocks – making sure that your body is grounded is essential.

### **Using Backups to Protect Your Data**

In the last section, we addressed physical security. In this section, we will consider a different issue – ensuring that your data and your programs are secure. How do you protect your computer data from corruption or loss?

**Rule 2: Make backups regularly and take steps to ensure that they will survive if your computer is physically threatened.**

Data can be corrupted or lost for a number of reasons. Some of the more common ones are:

- Accidentally deleting a file;
- Accidentally storing a new file under the same name as an old one, wiping out the old one;
- A misbehaving program that alters or corrupts your data;
- A misbehaving program that deletes your data;
- A rogue program (perhaps a virus) that alters, overwrites or deletes your data;
- A hardware failure (perhaps in the hard disk, or its controller, or the processor or power supply) that destroys data;
- A fire burns your computer or the water that is used to put out the fire renders the computer useless;
- The entire computer is stolen.

Creating backups is one solution to all of these problems. A backup is a copy of a file, or set of files, transferred onto a floppy disk or CD-ROM and put away for safekeeping. If the original file is inadvertently deleted or corrupted, the backup can be retrieved and the original file can be replaced.

Backups can be very simple, (e.g. a floppy disk in your desk drawer) or they can be exceedingly complex. Many backup software packages will let you copy every file on your computer onto a magnetic tape or a series of CD-ROMs. If your computer is lost or stolen, you can buy a new computer and the backup system will restore all of your files and applications on the new computer, assuming that the architecture of the new computer is similar to that of your old one.

Bugs, accidents, natural disasters, and attacks on your system cannot be predicted. Often, despite your best efforts, they can't be prevented. However, if you have good backups, at least you won't lose your data and, in many cases, you will be able to restore your system to a stable state. Even if you lose your entire computer, with a complete set of backups you can restore the information after you purchase or borrow a replacement machine. Of course, this will only work if the backups were stored away from the computer and not lost along with the computer.

Here are some reasons why backups are a key element in computer security:

#### **User error**

People accidentally delete their files. With graphical user interfaces, it's all too easy to accidentally drag a file or folder to the wrong place. Creating periodic backups makes it possible to restore files that have been deleted accidentally, protecting you from "finger-failure" mistakes.

#### **Hardware failure**

Hardware breaks from time to time, often destroying data in the process. Disk crashes may destroy the complete disk, but if you have a backup, you can restore the data onto a new drive or system.

#### **Software failure**

Many application programs, including Microsoft Word, Excel, and Access, have been known to corrupt their data files on occasion.<sup>23</sup> If you have a backup and your application program suddenly deletes half of your 500 x 500-cell spreadsheet, you will be able to recover your data.

#### **Electronic break-ins and vandalism**

Computer attackers and malicious viruses frequently alter or delete data. Your backups may help you recover from a break-in or a virus incident.

#### **Archival information**

Backups provide archival information that lets you compare current versions of software and databases with older ones. This capability lets you determine what you've changed, intentionally or by accident. It also provides a valuable resource if you ever need to go back and reconstruct the history of a project.

#### **Theft**

Computers are easy to steal and easy to sell. Not only should you make a backup, but you should also take it out of your computer and store it in a safe place; there are many cases where backups were stolen along with the computer system.

#### **Natural disaster**

Floods, earthquakes, and fires are all effective at destroying the places where we keep our computers. Here too, it is important to keep backups off site.

#### **Other disasters**

Sometimes Mother Nature isn't to blame: gas pipes leak and cause explosions, coffee spills through ventilation holes, computers may get dropped or knocked over. In each case, backups can prevent a misfortune from turning into an irrecoverable situation.

With all of these different uses for backups, it's not surprising that there are many forms of backups in use today. In fact, the perfect backup to recover from one of these problems might be useless for another. It is useful to remember the multi-layered defense concept and employ several forms of backup systems to cover the range of risks that you face in your home or office.

Here are a few types of backup methods to be considered:

- Copy your critical files to a floppy disk or a high-density removable magnetic or optical disk.
- Copy your entire disk to a spare or "mirror" disk or copy a disk to a folder/directory on the same disk if there is sufficient room. Obviously this will not help for catastrophic types of failure, but it does give you a copy in case of accidental deletion.
- Make periodic compressed archives of your important files.<sup>24</sup> You can keep these backups on your primary system or you can copy them to another computer, possibly at a different location.

<sup>23</sup> This statement is not meant to imply that these products have more such problems than others – they are listed only because they are the most popular applications used by users.

<sup>24</sup> Examples of compressed archives include "zip" and "tar" files that can contain very bulky information in a dense form. They are "unzipped" and individual files may be called up through fairly simple procedures. There are a number of vendors and some freeware available for file compression.

- Back up your files over a network or over the Internet to another computer.
- If you want high security against hard-disk failure, you may consider having two hard disks in your computer and use hardware/software that duplicates everything that is on the first disk on the second one as well. If you do this, you *still* need regular backups to protect against other types of problems.

### What Should You Back Up?

There are two approaches to computer backup systems:

1. Back up everything that is unique to your system except the application programs. This primarily includes your data files, but it *should* also include all of the files that tailor your operating system and your applications to you. It may be somewhat challenging to figure out where all of these files are kept and it is difficult to know whether it is safe to restore them later without making other critical changes. However, you may choose to keep all of your *data* files in a few major directories or folders. This way, you can make backups that only contain your unique work.
2. Back up everything. With an *image* backup, depending on the utility you use to make it, you can restore the system in its entirety. You can also restore individual files or directories/folders selectively.

We recommend both approaches.

1. Make a complete image backup as soon as your system is set up and back the system up periodically, perhaps once every several months.
2. On a more regular basis, you should back up your personal data. Depending on the backup utility that you use, there are several basic methodologies:

- a) Unless you have a massive amount of personal data, back up all of your data periodically, (every few months, for example).
- b) If you have a lot of personal data, you may consider backing it all up periodically and, at more frequent intervals, back up only the files that have changed since the last full backup. This is called an incremental backup. In this case, to restore a file or files, you will need the last full backup plus the last incremental backup.

There are other variations of these back up methods. Typically, backup utilities offer advice in their instructions on how to use their products.

### Where should I keep my backup copies?

The answer to this depends on how you may use the backups. If you are trying to protect your system from theft or fire, the backups must not be stored near your computer system. Ideally, they should be located far enough away that natural or man-made disasters affecting the system do not affect the backups. However, if you will use your backups for recovering data that has been deleted or altered accidentally, then you will want to keep them in a more convenient location.

One solution is to keep the full backups off site and incremental backups nearby. Another is to keep the most recent data backup nearby and a less recent copy off site. Some people make two copies of every backup, so they can keep one full copy on site, and one farther away.

Remember, if you have data on your computer that someone may want to steal, they can steal it from the backup as well. So it is important to protect the physical security of your backup, just as you protect the computer itself.

### Will I be able to read the backup?

There are a number of reasons that you will not be able to read a backup when you need it. Among them are:

- The copy is too old or is physically damaged. This is most likely to occur with floppy disks or other magnetic media.

- The device that wrote it was poorly adjusted and therefore what was written cannot be read. In this case, it *may* be readable by the same device that wrote it.
- Media failure. Media failure was common on old floppy disks. It was not unusual to create a disk that could not be read, even few days later. Optical disks (such as CD-Rs) have been thought of as extremely stable. However, a recent study of CD-R reliability has indicated that lower quality CD-Rs may not be readable in as little as two years after they are written.

It is always good practice to try to read a backup, preferably on a different device than the one that wrote it, to ensure that it is readable. If you write backups to removable magnetic disks (floppy, zip), make sure they are clean and reasonably new.

Some people keep their backups for a long time. It is amazing how often you really want to reuse a copy of a document or image or program that you had several years ago. If you keep backups for a very long time, you need to consider the possibility of media *obsolescence*. The data stored on a 5 1/4" floppy disk from the 1980s may still be there, but will you be able to find a computer with a 5 1/4" floppy drive?

#### How many copies should I keep?

Let's say that you make a backup once a week, so if you have some catastrophic failure, you will not lose more than one week's work. These backups are good from a security standpoint, but over time they will take up space. How many of these backups should you keep? If you are using CD-Rs as the backup media, there is no reason to discard them quickly, as they are small and cannot be reused. If you are using magnetic disks or CD-RW, then they can be reused. But you should always keep several backup copies. In the above example, you might keep the most recent four copies.

Why is this good practice? What possible reason would there be to keep the copies from the past month when you have the more up-to-date copy from last week? The reason is simple: it is always possible that the copy you made most recently is bad or will be lost, or stolen. The copies from last month are not as complete, but they

are better than nothing. This is another example of how good security is composed of multiple, partially redundant measures.

#### Backing up purchased software

If the license allows it, always make a copy of software CD-ROMs and use the backup for routine installation and maintenance operations.

#### The most important thing about backups

The *most* important thing about backups is that you create them regularly. Many people avoid the trouble of making backups. They may have even suffered previous losses due to insufficient backups, but they feel that they will not get hit again. Avoid risk and make regular backups!

#### Authentication

Authentication allows your computer or a distant web site to know who you are. It also should prevent other people from pretending they are you. Typically, you will be known by a user identification and password, although there are many variations on this theme. The challenge is to make your user identification and password combination hard to guess, so that attackers cannot figure it out. At the same time, it should be memorable enough so that you don't forget it or feel the need to write it down next to the computer. If you use computers and the web frequently, you will have *many* usernames and passwords. If they are all written in an obvious place near your computer, the usernames and passwords are not very secure.

#### User Identification

Most systems that want to identify you will either assign or ask you to select a "User Identification." It goes by many names: username, userid, member number, member name, etc. In this discussion, we will use the term *username*. Some systems will use your e-mail address as your username. In fact, your e-mail address is a specific example of a *username*. Systems often have rules about how the username should be composed.

- Some systems limit the length of the username, for other systems, the length is effectively unlimited.

- In some cases, any printable character is allowed in the username. In others, you may be limited to letters and numbers and perhaps a few punctuation marks.
- Some systems ignore upper and lower case, while others treat them as different characters (an “A” is not the same as an “a”).

If the system or web site does not give you a choice, then it will decide what your username is and you will be required to use this name. However, in the cases where you can select your own username, what are the criteria that you should consider? Sometimes, there are competing criteria, not all of which can be met at the same time.

- Do you want your username to reveal who you really are? Will this username be used to help your friends and colleagues recognize you? An e-mail address is often such a username.
- Do you want the username to help conceal your true identity? If you are using this name to participate in some group activity (such as an online game or chat group), you might not want people to know who you really are.
- Do you want this username to be easy for you to remember? If it is a username for some online service that you visit infrequently, you might want to pick a username that you will not forget. Some people use the same username for many services, if there is not critical or valuable information associated with these services.
- Do you want this username to be difficult for other people to guess? If it is the username to access your bank account, you might want to make it difficult for others to guess what it is (this goes back to the concept that effective security is made up of multiple, partially redundant layers; if you use your publicly known e-mail address to access your bank, it makes it easier for a thief to “guess” your bank username).

## Passwords

**Rule 3: Select passwords that you will be able to remember but will be very difficult for someone else to guess.**

Although usernames are often given to you without offering you a choice or are likely to be publicly known (such as your e-mail address), passwords can nearly always be set by you. Their form should make it difficult for an unauthorized person to access your account.

When passwords are stored on the host system, they are usually encrypted, so someone looking at the disk cannot see your password. In some cases, they can be decrypted by someone who knows the key. In other cases, it is not possible to decrypt the password (one-way encryption); when you enter a password while logging on, it is encrypted and compared to the version on disk (see Addendum 1 on Encryption for more details).

Due to poor security on some host systems, at times it may be possible for attackers to access the entire password table and find the encrypted passwords for all users. Even if these passwords use one-way encryption and cannot be decrypted, it may still be possible for the attacker to determine what your password is. The encryption algorithm used for these passwords is typically documented and known. The attacker could use this algorithm to encrypt all the words in a dictionary, as well as other commonly used passwords. So if you used the word “birthday” as your password, when the attacker encrypted the word “birthday,” he would find that the encrypted version is the same as what is on disk and would now know your password!

Since the whole idea of passwords is to make it difficult for someone to guess, but to allow you to sign on at will, one can state a number of criteria and techniques associated with robust passwords. Like usernames, each system has certain rules regarding the password formats (minimum and maximum size, what characters are valid, etc.)

- Never use single words in your native language (or English) as a password. A phrase or a sentence, or several word fragments is much better.
- If the system treats upper and lower case as different letters, use both, and do not place them where they would be used in normal writing.

- Mix numbers, allowed punctuation, and blank spaces, if the system allows it.
  - If the system allows blank spaces and your password is a phrase, consider omitting some of the spaces (that is, have the words run together).
  - To make your passwords easy to remember, you may be tempted to use the same password for many systems. If you do this, remember that once an attacker discovers your password on one of these systems, he or she can make a pretty good guess that it is the password on your other systems, so *only* do this for systems where you have absolutely nothing to protect. For example, some newspapers require a username and password to read articles on their web site. No money or confidential information is involved, they just want you to log on, and so it may be all right to use the same password for newspapers and similar reading material.
  - Some people replace letters in words with similar looking numbers or punctuation. They use the digit “1” for the letters “l” or “L”, the number “3” or the symbol “#” for the letter “E”, the digit “0” for the letter “O”, the symbol “@” for the letter “A” and the digit “5” for the letter “S.” This is a useful artifice, but remember, a good attacker knows about these tricks and they make his job a little bit harder, but not impossible.
  - Replace the letter “I” with the string “eye” or “aye” or whatever makes sense in your language. This works particularly well with words like “icon” which is now “eyecon.”
  - Use acronyms (the first letters of the words in some familiar expression). For example, “tgbwc” is an acronym for the Coca Cola slogan “Things go better with Coke.”
  - Spelling words backwards slightly obscures the words but does not make them much harder to crack.
  - Never use:
    - o Your username, or some variation of it
    - o Your name
    - o Your maiden name
    - o Your spouses name or maiden name
    - o Your children’s names
    - o Your parent’s names
    - o Your pet’s names
    - o Your co-worker’s, boss’s or friends names
    - o Your birthday, or the birthday of any of your friends or relatives
    - o Your address, phone number, license plate number or similar identifiers
    - o Your favorite color
    - o Your job title or rank
    - o Your company name or school name
    - o Anything else that is commonly identified with you
    - o Classic passwords such as “xyzyzy” or “plover” (passwords used in the first computer game), “abracadabra” and “open sesame”
    - o Words in popular movies, news or literature. Examples are “Harry Potter”, “Lord of the Rings”, and “Gone with the Wind”.
    - o Letters on the keyboard in order (such as “SDFGHJ”)
    - o Adding a single digit before or after any of the above.
    - o Repetitions of the same letters or numbers, or in sequence (“aaaa9999”, “123456”, “ABCDEFGG”)
  - Some systems require a minimum number of characters in a password or a certain number of letters and/or digits. Although long is good, as is mixed case, if you are not a very good typist, think about whether some one looking over your shoulder will be able to figure out what you are typing.
  - Whatever the password is, you will have to remember it, preferably without writing it down. If you need to write down a password, never write it near where it will be used, or with a label on it identifying it as a password.
  - Never keep an unencrypted list of passwords in a computer file.
- The best password is a very long string of random numbers and letters. However, for most of us, this would be impossible to remember and a password that is written on a note on your computer screen or under your keyboard is not secure.
- Here are some examples of reasonable passwords (for a system that accepts letters, numbers, special symbols and blanks, and treats upper and lower case as different letters) along with variations of each. They are memorable and yet not easily guessed or found in a dictionary.

<u>Password</u>	<u>Comment</u>
Computers Are Useful	Something many computer users will agree with.
Computers aReuseFul	One blank missing, funny capitalization.
C0mputer5@reus#fv1	Digit 0 for letter O, 5 for s, @ for a, # for E, V for U, 1 for L, no blanks.
comp9uter8sare7usef6ul	The original expression, with no blanks and with digits interspersed every four characters.
comutrsareusful	The original expression with a few letters missing.
onupatithwa	In many countries where there is a tradition of story telling, there are standard forms for beginning the story. In English speaking areas, children's stories often began: "Once upon a time, there was ...." In this example, each word is truncated to two letters to limit the length, which makes it less recognizable than "onceuponatimetherewas".
oNup@T-1thuua	The same thing, but with some substitutions, upper case letters, and arbitrary punctuation inserted.

### Changing your Password

Passwords should be changed periodically. The frequency of changes is the subject of debate. Some security specialists recommend changing passwords very often, but others argue that making changes too frequently increases the need to write passwords down or pick simplistic passwords. For *typical* applications, the following recommendations are realistic:

- Change your password immediately if you think that it may have been compromised.
- If you give your password to someone else for any

reason, change it immediately after they are finished. Sharing passwords is generally a bad thing, and should be avoided unless there is no alternative.

- Change you password periodically, just in case it has been compromised. "Periodically" is subjective, but between six months and a year is reasonable.
- If you belong to an organization that has a more stringent policy, follow it.

### Restrict Privileges

Most systems allow users to be given a restricted set of privileges; this set may not include all the privileges granted to the person who administers the computer. For computers where the user is also the administrator (as is the case for many personal computers), the user often does all of his/her work using the full set of privileges (often called root or administrator privileges). It is good practice to use a separate username when non-administrative work is being done. This reduced the chances that the user will damage the system by accident. It also reduces the chance that if the system is penetrated, the attacker will have full administrator privileges.

## CHAPTER 4. KEEPING YOUR OPERATING SYSTEM AND APPLICATION SOFTWARE SECURE

### At a Glance

This chapter investigates techniques you can use to reduce the chances that your operating system and applications software are vulnerable to security breaches.

### Introduction

Principle 1: Computers run programs.

Principle 2: Programs have bugs.

Principle 1 is obvious. Given that people write programs and people are not perfect, Principle 2 is expected. It is not clear, however, why there are so many security-related bugs. Problems such as *buffer overflows* (see definitions in Addendum 3) are easy to avoid; nevertheless, they seem to be involved in almost half of all known security bugs.

### Commercial Software

#### How does it normally work?

Several years ago, when you bought PC-type software, that was it; no updates were available until you bought the next version. Now most software is updated regularly, particularly for security problems. For some software such as operating systems, “regularly” means almost daily.<sup>25</sup> For most products, there is no charge for updates.

Many companies that offer commercial software also *provide* some updates to address bugs in general, and security vulnerabilities in particular. In the case of larger vendors, you can go to the corporate web site, click on a “support” or “downloads” tab and find any available fixes for their products.

Typically, when you go to a software supplier’s web site, you identify what software packages and versions you

have and they will list what updates are available. In some cases, it is completely clear what updates are relevant for your computer; in other cases the choices are less obvious. Once you have decided what updates you need, you download them onto your computer. The next step is to apply the update. Depending on the software, this may mean running the program that you have just downloaded or following the steps outlined in the accompanying documentation or instructions. In some cases, once the update is downloaded, it will install itself automatically.

In recent years, there have been three new trends:

1. For complex programs such as Microsoft Windows, Microsoft provides software via their web site (“Windows Update”). An applet inspects your computer and gives you a list of updates that apply to your system. You can then download and install these updates as described above.
2. The update that you find and install as described is not really the actual update, but a program that will, while it is running, download and install the actual update. So, for instance, you might find that there is a major update to one of your programs. When you look at it you will see that it is only 500,000 bytes – really small for a software update. In fact, this is just the program that will download the *real* upgrade and install it – the real upgrade consisting of perhaps 30,000,000 bytes.
3. Some programs have built-in functions that will dynamically check to see if updates are available and may even install them (with your permission).

These capabilities were designed to make your life easier. In all cases, the task of selecting exactly what updates you need (a complex task for operating systems and certain applications) is completed for you by the programs.

<sup>25</sup> In October 2003, following a severe security problem related to a problem in Microsoft Windows, Microsoft decided that it was unreasonable and unrealistic to have users apply patches weekly, and that in the future, they would only issue monthly updates unless a problem was severe and urgent.

### The developing country conundrum

As you can see, many of these processes are designed to run online and typically involve downloading many megabytes of updates. That works well if you have a high-speed connection to the Internet (greater than 1 megabyte per second), or a dialup connection where you can remain connected for several hours. In developing countries, however, this is often not the case.

There are two alternatives to address this problem:

1. Don't update your system and applications.
2. Have someone else download the update and provide detailed instructions for how to install it. The update can be distributed on CD or via a local area network, if there is one.

The first alternative is not acceptable given the rise in security risks. So, the only reasonable alternative is to work cooperatively to download and share the updates.

There are several vehicles for doing this:

- If an organization owns multiple machines, a local technical support person should take responsibility for downloading updates and installing them or making them available to others.
- Computer clubs or other groups could download updates and make them available to their members.
- For individual users, Internet Service Providers (ISPs) could offer a service whereby they get the updates for popular products and common operating systems and distribute them locally. This could also reduce the ISP's requirement for international bandwidth, reducing their costs.
- Computer stores that sell the machines can make the updates available to their customers.
- During a flurry of computer worm vulnerabilities in 2003, Microsoft began distributing some updates on CDs locally in various countries. Perhaps this practice will be continued.

The last three types of software update distribution are not prevalent, but given the increased need to keep software up to date, they may become a sensible commercial strategy for ISPs and vendors in the developing world. Although this will be a welcome support strategy for users, they will need to ensure that the source of these local updates is reliable and trustworthy. If they are not

reliable and trustworthy, they could become a way to distribute Trojans and viruses.

### Should you install updates as soon as they are available?

This has been a debate among computer professionals for decades. The two arguments are:

**Pro:** If you install updates immediately, you protect yourself from failures that are already known. In the case of security-related updates, you will protect yourself from penetrations and exposures that the original system allowed.

**Con:** Anytime programmers write code, they can make mistakes or break some other part of the program. This applies to updates as well as to the original programs, so there is a chance that the update will introduce new problems that are unrelated to the problems it is designed to fix.

The problem of attackers and criminals using security flaws to penetrate systems and alter or destroy data has changed the scope of the problem. Once a security flaw is announced, even if the announcement comes with a patch, attackers will immediately create viruses and other tools to exploit the problem. Those who do not implement security fixes *quickly* may be compromised.

Today's conventional wisdom:

- Novice users and those who use their computers for non-critical tasks should apply all updates soon after they are available. The risk of introducing new problems through the updates is lower than the risk of having a seriously out-of-date machine.
- Sophisticated users and technical administrative staff should install security-related updates immediately, but they can defer larger overall upgrades that may have multiple functional changes in them. Delaying for a few weeks or months may allow more adventurous users to install the upgrades, discover the problems, and report them, giving the manufacturer an opportunity to fix the flaws before you install the overall upgrade on your system.

If your computers are used for business applications, it is always a good policy to test *all* changes and new software on an identical, but non-critical computer before applying them to your production machines. You can never tell when a change will stop an existing application from working properly.

## Non-traditional and non-commercial software

The previous discussion focused mainly on commercial offerings including operating systems and major applications that are common to many computing environments. How does the situation change with other types of software?

### Shareware and small-supplier commercial software

There is a vast amount of software that is offered for free, or for a modest cost. The level of support offered by suppliers varies enormously. In general, upgrades are offered periodically, either for free or for a small fee. These programs do not tend to have security exposures, so their upgrades are aimed at fixing non-security flaws or adding functionality; as such they are beyond the focus of this book. However, some freeware applications, such as firewalls and virus checkers do fall in our domain and will be discussed later in this book.

If you use programs that have clear security implications, make sure you understand what the supplier's upgrade policy is. You do not want to be in a position where you are using security-sensitive software and the upgrade support suddenly disappears or you cannot afford to buy it. Deploying software such as a virus checker that is not *regularly* (daily or weekly) updated may be more dangerous than not using one at all, because if you use it, you may be working under a false sense of security.

### Open Source software

Open Source software that is in active development tends to be well supported. In some cases, there may be fee-based services available for upgrades and

support, even though the original software was free. Red Hat's version of Linux, which is available both for free and through commercial vendors, is a good example. Organizations that desire a higher level of technical support may find it worthwhile to purchase the package or at least the services to support it.<sup>26</sup> It is important to note that, as with some free software, if you decide to use the software at no charge and without paid support, the period for which security fixes are available may be quite short. Therefore, if you select non-support software for your operating system or other critical sub-systems, you may need to upgrade to new versions very often (perhaps as every six months).

The update processes for Open Source products tend to be more difficult than those for Windows, but are in line with other Unix products and the installation procedures for the original Open Source products. There are Open Source Windows-based products that distribute binaries and use simple installers as well.

As with Windows-type systems, updates and patches for large Open Source systems are sizable themselves. It is important to identify local sources of these updates to reduce Internet download times for individual users.

One final issue related to Open Source software is worth some discussion. There is an ongoing debate between advocates of Open Source and advocates of traditional proprietary software regarding which product is more secure.

Proprietary software advocates say:

- since the source is available for Open Source products, attackers can easily analyze the code and locate all of the flaws which they can exploit;
- since a large number of people in different locations and without organizational ties may be working on a given Open Source product, standards may be lax and the uneven integration of the various components may cause security vulnerabilities;

<sup>26</sup> See selected links on Linux and other Open Source projects in the Annex on Electronic Resources.

- since the people working on proprietary products are paid by the manufacturer, they follow instructions and the quality is uniform (and high);
- since no single authority is responsible for some Open Source products, security could be ignored if it does not happen to be important to any of the individual developers.

Open Source advocates say:

- since so many people are working on the source, problems tend to be recognized by the “good guys” and fixed quickly;
- the people working on proprietary products may generate uniform quality code, but it may not be secure if the manufacturer does not value security highly;
- with proprietary programs, you are at the mercy of the manufacturer to fix problems, and that may cause long delays.

In fact, each of these arguments has some validity to it. There is no way to ensure that either proprietary or Open Source software is secure or that problems will be discovered and fixed in a timely manner. In both types of software, there are examples of exemplary behavior and of careless behavior on the part of their respective designers and support organizations.

### **Pirated Software**

Neither the authors nor the publisher of this book advocate software piracy, but it would be foolish to pretend that it does not exist. Software piracy is a problem throughout the world, but it is particularly relevant in countries where the relative cost of legitimate software compared to wages far exceeds that in developed countries and where local laws and law enforcement make punishment highly unlikely.

Aside from the potential for legal liability due to violating the product owner’s property rights, there are two issues related to security and pirated software that must be addressed. Neither is very common, but both are possible.

- 1) It is possible that pirated software may not be updateable, or that an update may stop it from working.
- 2) Some pirated software includes other “goodies” that you may not have expected. These can include backdoors, keyboard loggers or other malicious software.

## CHAPTER 5. MALICIOUS SOFTWARE

### At a Glance

The concept of malicious software is introduced. The various types of malicious software (such as viruses, worms and Trojans) are discussed and the mechanisms used to spread them are investigated.

### Introduction

#### Malware

**Definition:** Short for **malicious software**. Software designed specifically to damage or disrupt a system.

The first known microcomputer virus dates back to 1981. The concept of a computer worm was introduced in a science fiction book in 1975, and the first actual implementations were in the early 1980s. Interestingly, these worms were designed to do good things instead of malicious things. Computer Trojan Horses date back to the early days of time-sharing (1960s). Despite their long history, it is only in recent years that their impact on normal users has been so severe and potentially dangerous.

To begin, we should first define what these terms mean.<sup>27</sup>

---

**Virus** A virus is a program that is attached to or inserted into another program. When that program runs, the virus also runs and it inserts copies of itself into other files or disks. In this way, it replicates itself. When the program it infected runs, the whole process starts over again. The virus may or may not do other things.

---

**Worm** A worm is similar to a virus, in that it replicates itself, but it does not need a host program. Like a virus, a worm may only replicate itself or it may take other actions as well. A worm can only work if there is some capability in a system that will allow an external source to send

it a program and run that program. Some malware detection vendors consider a worm a type of virus.

---

**Trojan** This type of software is named after the (perhaps mythical) Greek conquest of Troy, where the Greeks presented the city of Troy with a large wooden horse. When the horse was brought into the city, it was found to contain Greek soldiers who proceeded to take over the city. Since then, a “Trojan Horse” has meant something that looks benign, but contains some hidden and potentially dangerous content.

A Trojan horse program is one that can do something malicious in addition or instead of what the person thinks it is doing. The term has recently also come to mean any malicious program that is added to your system without your knowledge or authorization.

---

**“Bonus” software** This is software that is included in some other package without your knowledge. It is common for commercial software to include other packages. For instance if you install a web browser, it may also include Adobe Acrobat® or software that plays music or videos.

These are included because they enhance the original package and usually the install process asks you if you want them, or at least informs you that they are being installed. Bonus software is different because it is not really related to the original package in function. Given a choice, you probably would not install it.

The terms Trojan, Virus and Worm are not mutually exclusive. Attackers can write software with the characteristics of more than one, such as a self-replicating Trojan. Software that has the characteristics of more than one form of malware is often called a *blended threat*. As you can

<sup>27</sup> See [www.rbs2.com/cvirus.htm](http://www.rbs2.com/cvirus.htm) for further information on viruses and other potentially malicious programs.

see, the terms generally refer to how the malware is spread, and not what it does. This chapter describes what malware does and the specific ways in which it is propagated. The following chapters discuss ways in which your computers and networks can be secured against such software.

## What do they do?

There is no limit to how malware acts once it is running on your computer, but the programs do have some common characteristics in their activities:

### Send e-mail

Sending e-mail is one of the most common actions of malware programs. The e-mail may include a copy of the program itself (a virus or a worm) as an attachment. The content may be specific to the malware (such as falsely claiming it is an alert from Microsoft warning you about a security problem) or it may even be random parts of your previous e-mails that it finds lying around your computer. If there is a malicious attachment included, the text of the message may be something that will encourage the recipient to open the attachment. The Subject: and the From: line are similarly set according to the whim of the malware; they too may be set to encourage you to open the attachment (as in the famous worm that said "I LOVE YOU" in the subject line). The messages are typically sent to people it finds in your address book or to people whose e-mail addresses are in other types of files on your computer. Sometimes when messages have been sent to all possible recipients the program stops and sometimes it will start all over again! Note that if someone else's computer is infected with a virus or worm that sends e-mail and it puts *your* address in the From: line (because it found your address somewhere on the infected machine, perhaps in its address book), *you* may be accused of distributing this virus.

### Gather information

Malware may gather information about your computer and its files and send this information back to its author. Since it can read any files on your computer (often including encrypted files), whatever you have is fair game. If you store information about your bank accounts or credit cards on your computer, this data may be of interest to an attacker. If you have a scanned

image of your signature to allow you to print or fax letters, this may also be useful. Together these pieces of information could allow the attacker to assume your identity. Alternatively, if you operate a small business and store other people's credit card numbers on your computer, it will be a serious problem for you if these numbers are stolen.

### Over-write or erase data

Some malware programs are truly malicious; upon entry to your computer, they can immediately begin to erase all the files on your hard disk or overwrite the files with garbage. Sometimes they change things in less detectable ways including:

### Installing a Trojan

This aspect of malware is becoming increasingly common. One or more programs may be installed on your computer. The program may replace some common program that you or the operating system normally use (the original meaning of Trojan). Alternatively, it may insert some other program that will be invoked either at some pre-determined time or whenever your computer is started. The following section on Payload Software describes many of these programs.

### Scheduling something to happen later

Any of the previous actions may happen immediately or they may be triggered at a later date. Malware writers seem to like the suspense that comes with the announcement that a certain worm will do something nasty on January 1, 2000, for instance

## Payload Software

Malware often comes in the form of programs left on your computer that run when you start your machine or when you start a particular program. The type of program is only limited by the imagination and programming skill of the attacker.

### Web tracking/modification software

This class of programs watches what sites you visit, can display pop-up ads in addition to those you would

normally see, and can display ads replacing those that the site you are visiting is sending. They can send information about your computer and what you are doing back to its developer. In many cases, the software will also have full control over your browser, watches what you enter, and may alter what you see. When it watches what you enter, it can report these entries to its developer. For Internet Explorer, this capability is designed into the product and called a *Browser Helper Object* (BHO) - <http://msdn.microsoft.com/library/en-us/dnwebgen/html/bho.asp>. Although one can build very useful and legitimate BHOs, there are also clearly opportunities for less than ethical applications.

### Backdoor Software

Normally to access a computer system, you need to give it a username and password, although this security is often by-passed for systems that are thought of as being physically secure and used only in front of their own keyboard and monitor. Backdoor software allows a *remote* user to access your computer bypassing all of your security. It may even install its own security to allow only that attacker to use it. Although the details vary from case to case, this remote user will now have full control of your system; they could even lock *you* out if they wished. In essence, your computer has been hijacked and you will not realize it. Why does this attacker want access to your system? The reasons vary, but they may include:

- No reason other than to prove to himself or his friends that he could do it;
- To be malicious – in general;
- To be malicious – he has some specific reason to target you;
- To use your computer for some other activity such as sending spam or launching a denial of service attack later;
- To steal something of value from your system.

Note that this same type of software, under names such as remote access or remote administration tools has very legitimate, practical applications as well. If you use these tools for work, make sure that you have proper security measures employed, including usernames and passwords.

### Keyboard loggers

Keyboard loggers do just what the name implies. They trap all keyboard input and log it to disk. The file can be inspected later, perhaps via backdoor access, or it can be sent back to the person who installed the program via e-mail or web delivery.

It is important to note that keyboard loggers watch what you are actually typing, not what is sent over the network. So if you enter a credit card number on a web page that is secure (uses encryption when the data is transmitted), the logger *still* sees exactly what you typed in unencrypted form.

### Financial Theft

Most thefts that are the result of personal computer attacks involve information that is taken from the computer. However, there are cases where payload programs actually spend your money automatically. The simplest example is if the program detects a modem on your computer and uses it to place long distance calls. Since the program cannot talk, there is no benefit to the attacker, other than the malicious satisfaction in knowing that at the end of the month, you will get an outrageous bill from the phone company.

In other cases, the attacker can benefit personally. In many countries, it is possible to arrange to have a special telephone number – when this number is called, the phone company will charge the caller a specific amount per minute and part of that money goes to the person being called. It is used for a variety of businesses, but examples are software companies that want an easy way of charging you when you call them for out-of-warranty support. In that case, the phone company collects the money from the caller and sends part of it to the company being called to pay for the support call. If an attacker had such a number, they could program *your* computer to call the number and just hold the line open for a while. Your telephone bill would reflect this charge.

## How do you get them?

A number of years ago, the only way a PC or Macintosh user could be the recipient of a virus or other malware was to use an infected diskette. If you didn't trade files with people who were infected, you were safe. Unix systems were not particularly prone to viruses, but with their superior connectivity capabilities (even in those days), security holes in operating systems and some common applications occasionally allowed attackers to access systems and install backdoor software. The Internet's first *major* security incident was a worm that attacked Unix systems in 1988. Today, you can be attacked in a number of ways. All of the following apply to Windows machines. Unix and Macintosh systems are somewhat less prone to these types of attack, not necessarily because they are more secure, but rather because the vast number of Windows machines makes them more interesting targets.<sup>28</sup> Unix systems are next in line, with Macintosh exhibiting the fewest exploited vulnerabilities to date.

### e-mail

A few years ago, rumors would spread periodically that you could be infected with a virus by receiving e-mail. System managers and helpdesk people would have to reassure their users that this was *impossible*. As long as a user did not run a program that he or she received in an attachment without verifying that it was safe, the machine and the user were OK.

It is no longer impossible to be infected via e-mail, in fact, it is highly likely. Two enhancements brought this about. The first change is that we now have e-mail programs that can run attachments automatically.

Originally, a user would have to save the attachment and then run it. Now, automatically running attachments makes things easier, particularly for the novice user who wants to see what was sent without taking additional actions. The second change is that in an effort to make e-mail prettier and more powerful, we now allow HTML programming within the body of the

e-mail, however, that HTML can include instructions that cause problems. For example, the HTML can also direct a web browser to go to a specific web site that may not be appropriate for you or your children. It should be noted that the people who send these e-mails can be very innovative. Recently, there have been a number of virus-loaded e-mails that claim to be from Microsoft and say that they are providing the latest patches to protect you from viruses and worms. They contain logos and images that could easily convince someone that they are authentic and that the attachments should be run immediately. Needless to say, anyone who does run such an attachment is in for trouble.

### Web sites

When the World Wide Web was launched, web pages contained text and images. Now they can contain far more, including dynamic programs that are downloaded onto your machine and executed (Javascript, Java, ActiveX). If you allow your browser to run these programs without determining that the sending site is completely trustworthy, then there is a good chance that the program may do something objectionable. *Javascript* is generally safe, but Java and ActiveX are potentially quite dangerous. Browsers can usually be set to refuse these programs or to ask the user before executing one.

### Plug-ins and Add-ons

Web browsers and many other programs (including word processors and spreadsheets) allow other programs to be loaded and executed from within the main program. A common example is the Adobe Acrobat Reader<sup>®</sup> which allows you to view PDF files while browsing the web. Once these add-ins or plug-ins are installed, they can do anything that the base program can do, including (usually) read and write on disk, or use your network connection. Add-ins and plug-ins should only be installed if the source is known to be trustworthy.

---

<sup>28</sup> Typically, a virus, worm or Trojan written for Unix may work only on the variant (Red Hat, Solaris, etc.) that it was written for, because the libraries that interface applications to the operating system differ on each type of Unix. As Linux becomes more popular and standardized, this advantage will be reduced.

### Security holes

Security holes are bugs in parts of the operating system or other system components that allow an attacker to access information on your system, or to gain control of the system. In recent years, most suppliers are reasonably quick to respond to security problems that are discovered in their systems, so if you apply patches to your system regularly, you may plug the holes before would-be attackers build and distribute software exploiting the known bugs.

### File sharing

File sharing is available in one form or another for all operating systems. It is very convenient to share files among co-workers. If you have several machines of your own, sharing files between them is a great feature. However, if you allow file sharing over the Internet and you don't apply adequate security measures (such as robust usernames and passwords and limiting write and update privileges) then any attacker in the world can also share your files. Further, if you allow others to write to your disks, then the attacker can set up your machine to do anything they want!

### Drive-by downloads

Drive-by downloads occur when you innocently go to a web site and the HTML statements on the page automatically invoke a Java or ActiveX program that downloads another program and either executes it or schedules it for later execution. The HTML code can also arrive in e-mail. If you allow Java or ActiveX programs to execute, they can download and install whatever they want, without asking your permission and without telling you what is happening.

### Piggy-back on pirated software

Pirated commercial software is not new. Counterfeit CDs have been sold for years and copies on the Internet (called Warez) are common. There has long been a problem that the CDs could have a virus, but there is now an increasing chance that the software may deliberately include altered code giving access to your computer to an unauthorized person over the Internet. Since administrator privileges are needed to install most software, it is an ideal opportunity to add a few more programs that you had not requested.

### Piggy-back on legitimate software

Although most software that you download is probably legitimate, it is increasingly likely that downloaded software (particularly freeware) will install other programs as well. Peer-to-peer file sharing programs have been particularly prone to this. They often include other programs, many in the Web tracking/modification category, which monitor your web activity, display advertisements, and report on your activities to their masters. Some of these programs are particularly insidious in that they try to disguise themselves and they are almost impossible to remove. One such program includes a *uninstall* utility; if you run it, it deletes the uninstaller but the original program is still alive and running!

### Non-resident Malware

Not all malware runs on your computer. It is becoming increasingly common to send e-mail that somehow entices the user to visit a web site. The traditional form of this trap is when an e-mail offers you something that is of interest to you (just as with any of the common spam sales e-mails), but once you go to their site, some sort of malicious software takes over, perhaps downloading software (what is referred to as a *drive-by* download) or taking other actions.

In the newer form, the e-mail claims to be from e-Bay (the Internet auction site) or PayPal (Internet payments) or from your bank. The e-mails are crafted to *really* look like they are authentic. They point you to a web site to (typically) re-validate your credit card numbers. The URLs that they point you to *look* exactly like an authentic URL to the casual user. For instance, the real URL for PayPal is [www.paypal.com](http://www.paypal.com). The URL which displays in the e-mail might be exactly that. However, what is shown on the screen is *not* the actual URL that will be used to access the web. The actual URL pointed to is often hidden and might be something like: <http://www.paypal.com:user=32454329:transaction=43293:code=4333033.33@218.5.79.162>.

If one is not very familiar with URL formats, it really looks like it is going to [www.paypal.com](http://www.paypal.com), so it must be authentic. In fact, all of the data prior to the @ sign is ignored, and this goes to site 218.5.79.162. At that site, you would see a page that looks *exactly* like the

PayPal site, asking you to log in and re-enter your credit card number. In fact, this site is not connected to PayPal at all, but rather belongs to someone who is trying to steal your credit card information. These ploys have been very successful. Note that e-mails *similar* to this may be legitimate. A legitimate e-mail will usually include some information unique to you (and not included in your e-mail address) in the mail, such as your full name or the last 4 digits of your credit card. If they direct you to a web site, they will either tell you where to go, but not include a hyperlink, or the resultant web page will also include information that no spammer/fraud artist could know. If in *any* doubt, contact the company via telephone at their normal telephone number (not one included in the e-mail).

## CHAPTER 6. SECURING SERVICES OVER NETWORKS

### At a Glance

E-mail and the Web are the primary applications on the Internet. This chapter describes them in detail, investigating how they work and how careless use can result in security breaches. Other security-sensitive network-related topics covered include wireless communications, file sharing, and instant messaging.

### General Issues

You should update security patches for your software regularly. Although security problems can hurt you in many ways, you are most vulnerable when connected to the Internet. If there is a security hole in your operating system or application, you can be sure that the attackers know about it and will design ways to use it to infiltrate your computer.

**Rule 4: Keep your operating system and key application software up-to-date.**

By up-to-date, we do not necessarily mean the latest version of the software. Most companies and developers will issue fixes to bugs (at least security-related bugs) for older versions as well. Note that for free software, it is common for the developer to provide fixes only for the most recent version; this means that to stay security bug-free, you must regularly upgrade to the latest version of the software.

### E-mail

#### Evolution of e-mail

If you go back into network ancient history (10-30 years ago), e-mail was used for sending text messages. Most of the systems that deployed e-mail also had some way to transfer files. Typically though, the file transfer mechanisms were somewhat arcane and difficult to use. This did not matter much when the main users of networks were technology experts. However, as the use of e-mail spread to the greater public, the application had to become easier to understand and to use.

The problem was that traditional e-mail allowed only printable text, and most files such as word processing files or executable programs contain non-printable characters. The solution was to “encode” the non-printable information so that it was now printable. (Encoding is described in more detail in Addendum 1). This printable file was inserted into the e-mail message, preceded by a signal that what followed was an encoded file. When the e-mail message was received, this encoded file would be “decoded” back into the original form. Later, the concept of attachments was generalized to allow encoding more types of file. The new methodology was called MIME (Multipurpose Internet Mail Extensions). Once attachments became common, e-mail programs were changed to open these attachments automatically, so that the recipient could see what had been sent to them readily.

At about the same time, the World Wide Web was becoming popular and it used HTML to format web pages. HTML became one of the MIME encoding techniques, allowing e-mail to be formatted (changing fonts, colors, inserting images, pointing to web pages, etc.) as needed. E-mail programs executed HTML automatically.

#### Impact of enhanced e-mail

These enhancements made e-mail much more useful. Users could exchange all sorts of files easily. With skillful use of fonts, color, and images, mail could be more pleasing to the eye and relatively simple formatting could be employed without a word processing program. However, these enhancements had some negative aspects as well. As mentioned previously, in the days before these enhancements were available, you could *not* get infected with a virus/worm directly through e-mail. As long as you did not run a program that you received in an attachment without verifying that it was safe, you were OK.

Now, programs that you receive could execute automatically. HTML also executes automatically, which means that it can send you to web sites that take malicious actions, including directly downloading malicious software into your computer. In addition, specific HTML commands could give the attacker control of your machine, due to bugs discovered in the programs that ran that HTML.

### E-mail is NOT Authenticated

In most cases, the From: address of e-mail that is sent over the Internet is *not* authenticated. This is a capability that has been heavily exploited by spammers. When you Reply to e-mail, it normally goes back to whoever is listed in the From line. Sometimes, but not always, if you look at the *full* headers (all of those almost incomprehensible "Received from" lines), it may be possible to roughly identify where the mail came from.

### How to protect yourself

Anyone who knows your e-mail address, or is able to guess it,<sup>29</sup> can send you an attachment. This attachment could be relevant and useful to you or it could be a virus, a worm, or a Trojan, any of which could do a great deal of damage. Most current e-mail programs will not open attachments without your explicit request (typically by clicking on the attachment), but if your program will open attachments automatically, turn the option off.

**Rule 5: Configure your mail program not to open attachments automatically.**

**Rule 6: Before opening *any* attachment, look at the name to verify that it is not an executable program.**

Virus writers are cunning. One often finds an attachment with a name like *budget.xls.vbs*. To the casual observer who does not know what *vbs* is, this looks like Microsoft Excel spreadsheet named *budget*. In fact it is an executable Visual Basic program named *budget.xls*. The *xls* is just part of the name and unrelated to the Excel extension. The program could do anything it wished including erase your hard disk.

**Rule 7: Never open an attachment from someone you do not know unless you are *very* sure that it is a type of file that cannot contain malicious code.**

Remember that programs such as Microsoft Word (word processing) and Microsoft Excel (data spreadsheets) and

all of their equivalents contain macro-capabilities that can include a virus. Even PDF files can contain malicious programs, although these programs are dangerous only when viewed with the Adobe Acrobat program and not the Adobe Reader which most people use. You should check your user manual or help screens to see what capabilities may be turned off, especially if they are rarely used.

**Rule 8: Do not open an attachment from someone you *do* know and trust unless you are sure that they sent it deliberately.**

It is possible for a colleague's machine to have a virus that causes this machine to send infected files to all of the people in his or her address book.

**Rule 9: Consider configuring your e-mail program to not process "fancy" HTML and not to send it to other computers.**

This means that you will miss some images and other decorative things, but it also means that you will be in better control of your e-mail activities. Note that in some e-mail programs, you don't even have to open a message to execute the HTML code, having it in the preview screen is sufficient. Even though e-mail may contain HTML, many browsers and e-mail programs allow you to disable cookies, JavaScript, and plug-ins for pages that are received as part of e-mail messages.

**Rule 10: Check with your ISP to see if they are checking e-mails for viruses and similar threats before delivering e-mail.**

Due to recent increases in the virus/worm activity, more and more ISPs are doing this. Note that this does not alter any of these rules, as you cannot presume that your ISP filtering will be 100% effective, but your ISP's preventive actions will help in your security efforts. If your ISP is not aware of security issues, you may be able to work with them to deliver better service to you and their other customers. Feel free to share this Handbook with them!

<sup>29</sup> In the west, there is a children's story about a magical dwarf who promises to give a large reward if someone can guess his name. The person tries guessing many names, and eventually does guess the correct one – "Rumplestiltskin". To guess e-mail addresses, attackers repeatedly try many, many name variations in the hope that one of them will be correct. This is known as a Rumplestiltskin attack.

## SPAM

Spam is the name we use for unwanted e-mail, and in particular, unsolicited commercial e-mail sent out in massive numbers with no specific reason to believe that the recipient will be interested in the product. In recent years, the amount of spam has grown dramatically. In 2003, it is estimated that over 50% of all e-mail transported over the Internet is spam! Many people currently receive over ten spam e-mails for every valid one.

It would be nice if all spam would contain something like "\*\*\*SPAM\*\*" in subject line, so that we could delete it easily. In fact, laws being passed in some jurisdictions mandate that any unsolicited commercial e-mail sent from within their territory contain just such a warning. However, this type of legislation is not practical at the present time, for reasons of volume, extraterritorial spam, and enforceability. One must have a reasonable way of recognizing and eliminating spam without reading each message or notifying a potentially overburdened complaint system.

### Understanding Spam

To understand the problems associated with spam, one must look at three issues: a) how do the spammers get your address, b) how should spam be defined (in detail), and c) why do the spammers send these messages at all?

- a) If you engage in any of the following activities, there is a good chance that a spammer will obtain your address:
- Send mail or subscribe to a semi-public mailing list
  - Reply to a spam message saying that you should be removed from their mailing list
  - Post messages to a Newsgroup
  - Register for something on the web, giving your e-mail address (when you are not *absolutely* sure it is a reputable organization)
  - Use a computer that has an *Ident* daemon running (on many Unix systems, an *Ident* daemon will tell anyone who asks what your username is).
  - Let your web-browser know your address
  - Use IRC, instant messaging, or chat
  - Play games over the Internet

- Use an e-mail address that is a common given name, or an initial plus a common surname
- Put your e-mail address on a web page, or, in fact, allow your e-mail address to appear in print anywhere
- Register a domain name or be listed as the technical contact for a web site
- Use a "guessable" e-mail address
- Have your e-mail address on any system that has been maliciously penetrated previously

If any of these apply to you (and you will not necessarily have control or even knowledge about previously penetrated systems), there is a good chance that your address was *harvested* and sold to spammers. If you use the Internet to any extent, you are likely to be on some spammer's list of recipients.

b) Some commercial spam is obvious and by nature of its volume and irrelevance, virtually everyone will agree that it is spam. For other mailings, the distinctions are less clear. In some cases, it depends on the recipient whether a particular e-mail is considered spam, rather than on the actual mailing. Several examples will help illustrate the point.

- Is an e-mail considered spam if it contains information on how to change the size of certain sexual body parts? Answer: Yes. Unless you are a plastic surgeon or a urologist and the e-mail was an academic paper, not a commercial advertisement.
- A Call-for-Papers requesting people to submit papers for an academic conference on some obscure topic is sent to *many* mailing lists. Is this spam? Answer: Perhaps. Unless by some coincidence the subject was of interest to you and you will submit a paper.
- A company that sold you a product sends you information about a follow-on product at your request, along with a million e-mails to other customers who asked to be notified. Is this spam? Answer: No, but any spam filtering programs at your ISP may have a hard time understanding this, as it looks *like* spam.
- An e-mail contains content that is spam by any definition. Is it spam? Answer: Yes, when it was originally sent. But if it was then forwarded to this author by a trusted colleague as an interesting example to include in this book, it is not spam and should not be filtered.

c) Why do spammers send spam? The simple answer is because it works. If you look at spam, you quickly see a pattern.

Most spam is about:

- Making or saving money
- Improving your love-life or sex life
- Improving your health

These topics have one very important thing in common. Most of us care about these issues to some extent and many of us are deeply concerned about them. So even though a very small percentage of recipients respond to spam messages related to these topics (estimated at about 1 purchase for every 100,000 e-mails sent), spammers who send out many millions of messages per day might make a lot of money.

### What can you do about spam?

There are many ways that one can attempt to control and limit spam. Some governments are enacting legislation prohibiting spam mailings from within their jurisdiction. Most ISPs say that using their facilities to send spam is a violation of their usage agreement. Rules such as these can be effective, but to date, most spam-related rules have proven difficult and costly to enforce.

Some large (e.g. corporate) users of e-mail refuse to accept mail from ISPs that are known to allow spammers to operate. This can be effective, because it may force the ISP to clamp down on spamming activities. However, more often this method simply hurts the enterprises' innocent customers who can no longer send e-mail to some locations. There are a number of programs that try to recognize spam and either delete it or warn the recipient that the mail *looks* like spam. These programs can be run at an ISP's site or in your own mail client. The programs will look at the content of e-mail and/or its point of origin. These criteria are difficult to evaluate, and such programs often will generate false negatives or false positives.

---

**False negatives** A false negative is produced when the scanning program decides that an e-mail is not spam, but it really is. This means that it lets

some spam through and thus is not 100% effective.

---

**False positives** A false positive means that the scanning program decides that some innocent mail is spam. This can be very dangerous, particularly if the mail is discarded instead of being delivered. False positives may mean that good mail is lost and unrecoverable through electronic means.

The target in spam scanning programs is to minimize false negatives and to have no false positives. Unfortunately, reducing false negatives usually increases false positives. People who, for whatever reason, need to receive mail that looks like spam can be hurt, in particular. A recent case involved an academic electronic newsletter that discussed spam. Since the newsletter included examples of spam, it was viewed as spam by some spam scanners, and was deleted by several ISPs.

In addition to spam scanners, there are also spam-filtering techniques which involve the sender in the process. One spam filtering technique is a challenge-response process. When mail is received from an unknown sender, it is intercepted before the recipient can see it. A challenge is sent to the sender requesting a confirmation that the mail was sent by an individual and not a program. The form of the confirmation is such that a human must reply; it cannot be handled automatically, at least not in a manner that is effective for the would-be spammer. If no confirmation is received after a few days, the mail is discarded. There are provisions for accepting mail from known mailing lists and other desired automatic mailings. The problem with this technique is that it requires manual intervention by the sender. If you send mail and then are unable to quickly respond to the confirmation request, your mail will not be delivered. If two people both use this type of service, it is possible that they would never get any mail from each other, because the first receiver will not see the mail unless it is confirmed and the request-for-confirmation will not be passed on because it's sender is also unknown. Some spam-filters put suspected spam into a low priority folder, rather than deleting the messages. Then you may periodically review the spam folder to make sure that it doesn't contain any false positives.

A promising new anti-spam technique is *Bayesian Filtering*. In this method, the filter's rules improve by learning what you consider spam; these rules can be changed by each recipient. These rules tend to learn who your trusted colleagues are and, at your request, will allow content that would normally be spam, but is of interest to you for some reason. Bayesian filters also employ linguistic techniques to allow mail containing certain words that rarely appear in spam, but do appear in your real e-mail, based on prior experience with your e-mail habits. Bayesian filters are being made available for many e-mail programs.

If spam is a problem for you, you should see if your ISP offers any spam identification or filtering capabilities. You should also look into software programs that can filter out spam as it arrives at your computer.<sup>30</sup>

## Using the World Wide Web

As this is written in 2003, the web has been available in varying degrees for about ten years. For those who use it regularly for work, school, and recreation, it has become indispensable. Since the web has become such a common and useful tool, there is a tendency to forget that it can also be a hostile place.

### Safe Browsing

In general, the web is relatively safe, but there are potential dangers. Web sites usually house content, including static text and images, but they can also house dynamic programs that are intended to run on your computer.

**Rule 11: Do not allow web sites to download and execute potentially malicious programs on your computer unless you know that the site is trustworthy.**

Dynamically downloading programs can be very useful. This capability allows you to use online services, including those needed to check your computer for viruses and security problems. It also enables software to be installed and updated easily, without requiring the user to select

technically appropriate modules or perform complicated multi-step procedures.

Unfortunately, dynamically downloaded programs can also be malicious. All browsers allow you to control whether you can download and run JavaScript, Java, ActiveX and other programming tools on your machine. If you want to be completely safe, then you will not allow these tools to run. Of course, by disabling these features, you will find that many web sites cannot function without them.

Instead of blocking your access to so many sites, you may wish to follow a reasonable intermediate path:

- Enable the relatively safe and very commonly used capabilities such as Javascript. This will allow the vast majority of web sites to function properly.
- Either disable the less common and much less safe capabilities such as Java and ActiveX, or set the browser to ask your permission prior to using the capability. Disabling these capabilities means that the functions will not work; some sites may warn you about this, others will simply not work properly or will hang. If you request prompting, however, the browser should detect the requirements of the site and will ask for your permission to download and run a program needed to view that site's content.

**Rule 12: Display the web site address you are visiting and the address you are linking to, and pay attention to them while visiting an unfamiliar web site, especially if you are allowing the site to execute programs on our computer.**

Web browsers can be configured to show what web site is being visited (often called the Navigation or Address Toolbar). When your cursor is pointing to a link, they will also display where that link will take you (Status Bar). Watching these will tell you when you are being transferred to another site, perhaps one you do not want to visit, or perhaps one that is not trustworthy. On a practical level, you are probably not going to look at the Navigation Bar and the Status Bar every time you *click*, but when you are at an unfamiliar site, particularly if

<sup>30</sup> See Annexes 2-4 for web sites and other resources on anti-spam software and techniques to avoid spam.

you have enabled Java or ActiveX, you can use these tools so that you know that you are being redirected to a new site without your permission.

### Cookies

A *cookie* is information written to your hard disk by your browser at the request of a remote web site. When you visit the site later, the cookies owned by that site are sent back. Cookies are typically sent back to the originating web site only, although there have been browser bugs that allowed other sites to see them as well. A cookie reminds the web site who you are, what your preferences are, and what you have done before on this site. For instance, when you log onto a site with your username and password, the site can store this information in a cookie on your computer. When you return a week later, it can automatically log you onto the site based on the information in the cookie. Cookies may also allow a web site to track what you are doing in a single session.

Although a cookie normally can only be retrieved by the originating web site, it is important to understand that the web site that you are visiting may contain images and other objects from a second web site (called a *foreign* or *third-party* site). That foreign web site can also store and retrieve cookies. Since images can be transparent, you may not even know that this is happening. Such invisible images may be used for advertising purposes,<sup>31</sup> tracking what web sites you visit.

**Rule 13: Consider controlling under what situation you allow cookies to be stored on your computer. If you cannot control them (such as when using a computer in a public location), consider not entering private information.**

All web browsers give you a certain degree of control over whether cookies are allowed or not. In some cases, the browser may differentiate between cookies that stay on your computer, cookies that disappear when you close your browser, and those that are stored by the web site you are visiting and foreign web sites. Typically, you can allow all cookies, disallow them, or

have the browser ask for your permission before storing a cookie. You are never informed when a cookie is sent back to a web site.

Cookies can be viewed, since they are in text format, but typically the information has been encoded or encrypted by the web site so it is not intelligible. Some browsers allow you to display and delete cookies, and there are third-party programs that allow you to manage cookies.

If you wish to control what web sites know about you, you should control how and when cookies are being stored on your computer. Note that some sites *require* that cookies be stored to allow the site to function at all. Generally these sites will tell you if they find cookies disabled.

If you use web browsers from public locations (Internet cafés, libraries, schools), note that cookies containing information about you are still being stored on that computer. In many cases, the computer owner may not allow you to control, view, or erase these cookies. So information about you may be left on these computers and used when someone else visits the same site. If you logged onto a site and your authentication information is remembered in a cookie, another user going to that site may automatically be logged on as you! That web site may then give that user stored information about you (such as your name, address, credit card information, etc.).

Even with a private computer used by several people, this can be an issue. In these cases, cookies are not only a *privacy* issue, but also a *security* issue.

### Web Browser Caches

When a browser retrieves a page or an image from a web site, the browser displays the site and usually stores a copy of that page on your hard disk. This set of stored pages and images is called a *cache*. If you visit that site later and the page has not changed, the browser may not download the full page from scratch, but instead will use the one in the cache. In some

<sup>31</sup> Consider what happens if web sites A, B, C and D all include an invisible image from web site Z. When the invisible image from Z is displayed, Z is told which site pointed to them (A, B, C or D), and Z retrieves and restores a cookie remembering what web sites you have been to. Z now has a good idea of what types of things interest you, and can arrange for targeted advertising to be sent to you.

cases, web pages that are in a cache can also be viewed offline, when you are no longer connected to the Internet. This means that anything that you display with a browser may be stored on the computer's hard disk as well. So if you use the web for financial transactions, information about your purchases, credit cards, and bank accounts may be stored on that computer in fully readable text. Depending on how much browsing is done on the machine and the size of the cache that is configured, these pages and images can stay on the computer for a very long time.

**Rule 14: If there is any sort of private information displayed on a web page, clear the cache after the session is over. If you cannot clear the cache (such as when using a computer in a public location), you may decide not to use this particular computer for the task.**

All browsers allow you to clear the cache (called Temporary Internet Files by Internet Explorer), but some public machines, such as those at Internet cafés, do not allow you to access the control windows that clear the cache. Although clearing the cache after entering sensitive information is very important, no browser so far has put an icon on its main toolbar to allow this to be done with one click.<sup>32</sup>

### Secure Transmission

Normally when you are using the web, all the messages that you send and receive are in clear text. That is, if someone were to intercept them and print them, they would be readable and understandable. There are times when this is undesirable. Interception is of particular concern if any part of your Internet connection goes over wireless services or if the ISP at either end of the connection is untrustworthy.

To address this, browsers and web servers support encryption. Encryption changes the messages so that they are difficult or impossible for unauthorized people

to read. (See Addendum 1 for details). The name of the encryption capability is SSL for Secure Socket Layer. You can tell if SSL is being used for messages sent to you because there is (for most browsers) a picture of a small padlock on the screen that is open for normal transmissions, and closed (locked) for SSL transmissions. Also, the URL will start with "https" instead of "http". You should always use the strongest encryption possible – 128 bit is best if it is available in your country.

Note that this padlock does not tell you that your message going back to the server is using SSL, but it is normally assumed that if the screen you received is encrypted, the web site will ensure that your return message is also encrypted.

SSL can only work if your browser knows who it is talking to. This is accomplished by means of "security certificates" and "digital signatures". In general, if a web server wants to be trusted, they must obtain a security certificate from some recognized authority. If the authority is doing their job properly, they verify that whoever is requesting the certificate really is who they say they are. This authority then signs the certificate digitally and your browser has built-in tables to recognize these authorities.

Occasionally, you will get a message that a web site has sent you a certificate that:

- has expired, or
- is someone else's certificate

In the former case, it is usually the case that the certificate has just recently expired, and the site needs to get their paperwork in order. In the latter case, it is usually the case that the site has been recently renamed and that is not reflected in the certificate. However, in both cases, you may want to play it safe and terminate the connection until the problem is rectified.

<sup>32</sup> For Internet Explorer on Windows, Select Internet Options on the Tools pull-down menu. On the General tab, under Temporary Internet Files, hit the Delete Files button.

For Internet Explorer on a Macintosh, Select Preferences on the Explorer or Edit menu, go to Web Browser and then Advanced, and in the box marked Cache, hit the Empty Now button.

For Netscape/Mozilla, Select Preferences on the Edit pull-down menu. Expand the Advanced entry and select Cache. Hit Clear Disk Cache. For Safari on a Macintosh, Select Empty Cache from the Safari menu, and hit Empty to confirm.

### Is secure transmission sufficient?

The little locked padlock is designed to tell you that the web transmission is secure, and it accurately reflects that. However, *transmission* is not the only issue to consider. Only a very small percentage of cases of fraud or identity theft occur due to insecure transmissions. The vast majority of cases are due to:

- unscrupulous web sites,
- the web site has been compromised, or
- your computer has been compromised.

The one major exception to this is for wireless transmissions, which will be covered next.

### Privacy Policies

Many web sites publish a *Privacy Policy*. A privacy policy should describe what kind of information the site collects, what they will and will not do with that data, and how they protect the data. *All* web sites that collect personal or financial data should have a suitable privacy policy.

## Wireless Transmission

Wireless technology of various sorts is increasingly being used in developed countries and in developing countries. It is often less expensive than wired technologies, easier and faster to install, particularly in less populated areas, and subject, at least at the moment, to less regulatory oversight. However, wireless technologies have two potential problems:

- It may be possible to intercept transmissions, and
- Transmission quality may vary with location, weather, time of day, nearby radio equipment, transmission speed, quality of the installation, and malicious interference.

There is little that can be done about the second group of problems. They are characteristic of wireless technology and may be seen as the price that is paid for connectivity without wires. Interception can be addressed through

various levels of encryption. (See Addendum 1 for details on encryption techniques). If the server you are communicating with supports encryption, it should be used (secure SSL web sites, for example). If you use POP e-mail, you should select the "APOP" option that will encrypt your password before sending it, instead of sending it in clear-text. This will give you end-to-end security regardless of the transmission medium. If the server does not offer encryption, you should be aware of the technology limitations and adjust how you use the connection, if necessary.

### 802.11 "Wi-Fi"

802.11 is a set of developing IEEE standards for wireless local area networks (WLAN).<sup>33</sup> 802.11, (often called "Wi-Fi" – short for **Wireless Fidelity**) is becoming popular as an alternative to wired Ethernet for connecting computers and laptops. On the positive side, it is inexpensive and relatively fast. Unfortunately, there are several vulnerabilities in most implementations:

- Typical base stations are shipped with no security enabled.
- Unless you want to share your network connection with someone in the neighborhood, you should change the network name (SSID) from the default one and set the configuration not to transmit it. If you do this, only those people who already know the SSID will be allowed on.
- The encryption mechanism (WEP) is weak and can easily be broken. Nevertheless, in the absence of a better mechanism, you should enable it. Remember that it is vulnerable to attack if anyone really wants to look at your transmissions, including passwords.
- A new encryption mechanism, WPA, resolves the problems in WEP and it is available in newer equipment. It is strongly recommended for all Wi-Fi installations.

<sup>33</sup> For Internet Explorer on Windows, Select *Internet Options* on the *Tools* pull-down menu. On the *General* tab, under *Temporary Internet Files*, hit the *Delete Files* button.

For Internet Explorer on a Macintosh, Select *Preferences* on the *Explorer* or *Edit* menu, go to *Web Browser* and then *Advanced*, and in the box marked *Cache*, hit the *Empty Now* button.

For Netscape/Mozilla, Select *Preferences* on the *Edit* pull-down menu. Expand the *Advanced* entry and select *Cache*. Hit *Clear Disk Cache*. For Safari on a Macintosh, Select *Empty Cache* from the *Safari* menu, and hit *Empty* to confirm.

### Mobile Telephones

Mobile telephones (often called cellular or hand-phones) are widely used for voice transmissions. At times, they are also used for data. Many mobile telephone technologies allow eavesdropping and are not secure.

### Long-haul Lines

Long links, particularly to remote areas, are often built using wireless technologies. Typically the link will serve many users simultaneously. If the transmission method is highly directional (using dish or yagi antennas), it is relatively difficult to intercept transmissions without specialized equipment. These links may be encrypted with the addition of hardware encryption devices if necessary.

### Local Loop Wireless Telephones

Wireless local loops to homes and businesses are used in many countries, as they allow telephones to be installed without the cost and trouble of building wired infrastructure, and because wireless equipment is not as easy to steal and resell as is copper wire. As with a wired telephone, when a modem is connected to these lines, it becomes a data link. The wireless technology used may be interceptable. Depending on your location, your country's regulations, and local practices, you may want to check with your service provider to see if the link is encrypted, and thus protected, at least to a certain extent.

## Other Internet Issues

### File Sharing

File sharing is one of the most useful networking tools if you have more than one computer. In the simplest situation, it lets you access, change, create, or delete files on one system while working on another system. The two systems could be in the same room or they could be half a world apart. Among other things, file sharing allows you to copy files to and from a laptop prior to traveling or while you are away on a trip. At the other extreme, a single computer acting as a *file server* can take the place of the hard disk for a large number of computers. In this case, most or all of your files reside on the file server and you access them over the network.

The obvious vulnerability is that if you can access your files remotely, someone else can do so as well. A less obvious vulnerability is that if you share files with another user, you become vulnerable to security problems that may be present on their computer – if they become infected with a virus and have write-access to your files, you may now be infected. If you read an infected file from their disk, you may now be infected.

**Rule 15: If you are not using file sharing, disable it. If you are using it, to the extent possible, limit the kinds of things that can be done to those functions that you need.**

**Rule 16: If you use file sharing, set robust usernames and passwords and limit the access permissions to the least possible that will allow you to do your work.**

**Rule 17: If you share files with another user, make sure that they take security seriously.**

Virtually all file sharing and remote file access capabilities allow you to set up usernames and passwords to control access. Generally, they also allow you to control what a user can do (read-only, write, create, erase). Many systems allow you to control what *any* user can do. For example, you could restrict the entire remote access facility so that it only allows read-access; if you do not need write access, disable it if you can.

Typically, systems that support some form of file sharing also support the sharing of printers. Although giving someone remote access to your printer is typically not hazardous, it is better to restrict such services unless they are needed. It is possible that a bug will be detected that allows malicious actions through an access that *should* have been used for printing only.

### Instant messaging

Instant messaging is a facility that allows a message typed on one computer to be displayed on one or more other computers virtually instantaneously. Unlike e-mail, both sender and recipient must be online at the time. Instant messaging goes under many names on various systems. Among them are: Chat, ICQ (an acronym-like

homonym for “I Seek You”), IRC (Internet Relay Chat), Talk, AIM (AOL Instant Messenger), and Messenger. Internet communities such as AOL, MSN, Yahoo, game-playing hosts, and many others all have their own Messenger and Chat variants. Some of these interoperate with others, and some do not.

Many messaging systems allow you to select a name that will be displayed with your messages and that allows other participants to send messages to you. They often allow your real identity to be disguised, although the system administrators can identify who you are, at least by your IP address.

**Rule 18: Instant messaging can be very helpful, but use it with care and knowledge.**

Instant messaging plays a very useful role for several reasons:

- it is much faster and easier to use than mail and has almost no delay – this makes interactive conversations much more practical than e-mail,
- messages can usually be sent and received in a small window on your screen while you are doing other work, and
- you do not have to reveal your e-mail address (and identity) to other participants.

For certain types of uses, messaging is far preferable to e-mail. In some people’s minds, it is also more secure, as the messages are not copied to disk at various places, as is the case for regular e-mail. However, users are cautioned that messaging is still not particularly secure. The major problem with messaging systems is that some of them have been expanded to allow file transfer. This makes them vulnerable to the same problems as other types of file sharing, including e-mail attachments. Some messaging systems also allow remote execution of commands, potentially allowing attacks on your computer.

**Improperly Enabled Services**

Operating systems and applications have become very powerful and functional. In most cases, a typical user does not need or want all of the capabilities that their software offers. Services that are not needed should be turned off (disabled). Unfortunately, some software suppliers ship their software with all services enabled and

it is up to the user to turn them off. Often the user is not even aware that the services are there. For many years, some Unix systems were designed so that every installed user machine could act as an unrestricted mail hub if they did not explicitly turn the capability off. This allowed spammers to use these machines to send spam, without the machine owner’s knowledge.

**Rule 19: Disable all Internet services that are not needed and used regularly.**

Increasingly, suppliers are becoming aware of the problem. So, despite their pride at developing feature-rich systems, they are shipping their programs with extraneous services disabled; the user may enable them, if they are needed. In either case, it is important for users to make sure that unused services are not enabled. Such services include file and print sharing, web servers, mail servers, file transfer protocol (FTP) servers, Remote Procedure Call (RPC) servers, and others.

## CHAPTER 7. TOOLS TO ENHANCE SECURITY

### At a Glance

In this chapter, software tools and techniques to enhance computer and network security are investigated. These software packages include virus checkers, firewalls and remote access tools.

### Virus software

**Rule 20: Every computer that is vulnerable to viruses should run anti-virus software and should check for up-to-date virus signatures daily. A full scan of the machine should be performed periodically as well.**

**Rule 21: Computers that are not particularly subject to viruses such as Unix-based systems should nevertheless ensure that the mail that they send out does not contain a virus that may harm the recipient.**

**Rule 22: Keep your operating system and key application software up-to-date and remember that virus checkers only check for infestations in files. Vulnerabilities in operating systems and applications programs can leave you open to attack in other ways.**

Virus checking software attempts to keep your computer free of viruses, worms, and Trojans in a number of ways:

- Whenever you access, copy, save, move, open, or close a file, the virus checker makes sure that it is not infected with any known virus (and other similar pests).
- Whenever you insert a foreign disk in your machine, it is checked for certain types of viruses.
- Whenever a mail file is received, it (and attachments) is scanned for malware.
- Whenever a file is downloaded from the web, it is scanned.
- In many cases, when a web page with embedded software is downloaded, it is scanned.
- You can explicitly request that any file, set of files, or entire disks be checked for viruses.

- If a virus, worm or Trojan is detected, the program will either remove it (disinfect) or it will tell you that the problem cannot be fixed and will “hide” the bad file so that it cannot cause any damage.

A virus checker with up-to-date virus signatures (a signature is the specific characteristic of each virus that is recognized by the checker) is an essential part of any computer, whether it is Internet-connected or not. Note that there are few known Unix *viruses* at the time this is being written but Unix worms and Trojans certainly do exist.

As of the end of August 2003, one of the popular PC/Macintosh virus programs (Norton AntiVirus™) checked for almost 65,000 different viruses. That these programs can do this as fast as they do, without perceptibly slowing down your computer, is quite amazing. August 2003 was a particularly interesting month for malware, with the release of several worms (Blaster and SOBIG being the most common ones) that took advantage of a vulnerability in Windows computers. A month earlier, Microsoft had released a patch for this vulnerability, but relatively few people installed this patch, and so these new worms hit new records for the number of machines infected and the speed at which they spread. They may have also set new records for the number of “copy-cats” – the same basic worm, but with various modifications. On the busiest day, Norton added fifty-one new virus signatures (defining characteristics of those viruses) to their list. For the whole month, 520 new signatures were added.

### Firewalls

A firewall watches all network activity going into or coming out of your computer. Based on a set of rules, it can allow the traffic to pass or it can block it. A firewall can be either a program running on your computer or a separate piece of equipment between your computer (or a cluster of computers) and its network connection. Sometimes firewalls are included in other equipment such as routers. There are free or pre-installed firewalls available for many operating systems.

**Rule 23: All computers should be protected by a firewall of some sort, either software within the computer, or an external firewall protecting that computer or an entire local network of computers.**

To fully understand what a firewall does, and how to set up the rules that govern it, you need an introductory understanding of TCP/IP – the protocol (set of rules) governing all messages sent over the Internet. If you are already familiar with the TCP/IP protocol, you should go directly to the next section. If you are not already familiar with TCP/IP, you should first read Addendum 2. TCP/IP. Note that a firewall can be used even if you do not want to learn these technical details. In that case, here is all you need to know about TCP/IP:

- Machines on the Internet all have an “IP address” that has the form 12.222.103.43, that is, four numbers separated by periods. The Internet uses your address to route messages to you, and your computer says where to send out-going messages by providing the address of the destination.
- Within each machine, different programs are identified by the “port” number (sort of like a telephone extension number within a large company – there is just one telephone number, but each person has their own extension number).
- Information sent to or from your computer is enclosed in “envelopes” called *packets*.
- Ignore the words TCP and UDP in the following discussion.

### Why do we need firewalls?

If your computer is not connected to a local network or to the Internet, you do not need a firewall. Once you use the network, you are subject to all sorts of abuse. For example:

- If you use file sharing, print-sharing or any other inter-computer services, your computer is probably listening on certain ports. Although you may be doing this so that the computer in the next room can share your resources, it is possible that a computer anywhere else in the world could as well.
- If you are listening on a port for (for instance) file sharing, it is possible that due to bugs in the program, someone could send you a message that would take some other action – perhaps malicious. Unfortunately, such bugs are quite common.
- Even if you are not listening on any port, computers elsewhere can send you floods of messages. Even

though they will all be ignored, they can keep your network connection so busy that you cannot do any real work (only hardware firewalls will help you in this case).

- If, despite your best efforts, you do end up with a virus, worm or Trojan on your computer, it can send anything on your computer to the malware creator. This could include any of your data or logs of what you are typing (including passwords).

### How do firewalls work?

A firewall watches every packet that is received by or sent from your computer, and verifies whether it violates any of the rules that you have set for it. If a packet violates the rules, it is blocked (discarded). For both software firewalls and external (hardware) firewalls, the rules might include:

- Do not allow any packets to TCP/UDP ports 135, 137, 139, 445. These ports are used for Windows file sharing and a selection of other Windows services. By discarding these packets, you are ensuring that no one on the Internet can contact your computer for these services.
- Do not allow any packets to TCP/UDP ports 135, 137, 139, 445 *unless* they come from IP address 192.168.1.150 (where 192.168.1.150 is that address of your second computer that is allowed to share your resources).
- You can give the firewall a list of *trusted* computers – those that you know are not trying to hurt you. Only trusted computers will be able to initiate communications with you. You can still communicate with other computers, such as web servers on the Internet, but you must initiate the communication.

Software firewalls consume resources on your computer, but have the added advantage that they not only look at the datagram (with its to/from address and ports), but they can check which program is sending the message. If it sees a program initiating a communication that you had not explicitly allowed, the firewall can ask you for your permission before allowing it to go through. A hardware firewall cannot determine which program is being used, but since it is a separate piece of equipment, it does not slow your computer down at all.

Like all security-related precautions, if you have a firewall, whether hardware or software, you must keep the software and firmware up to date. Attackers are very innovative and it is essential that the tools that you are using to protect your system and data are current.

### Private Address Spaces and Network Address Translation (NAT)

As the Internet was originally designed, every computer or device on the Internet had its own address, so there was the ubiquitous ability of every computer to talk to every other computer. Today, there are cases where universal connectivity is no longer appropriate. There are two primary reasons:

- You *want* to isolate a set of computers so that they cannot directly talk to the rest of the Internet – and the Internet cannot talk directly to them. This is the case with computers within some organizations, both public and private.
- Because of the way that IP addresses are allocated within the Internet, your organization does not have enough IP addresses to assign unique addresses to every machine. This is often the case with developing countries where national Internets were built (or are being built) several years after comparable networks in developed countries.

There are certain IP addresses that are not usable over the Internet. These are called *Private Address Spaces* and can be used in the above two cases. Since these computers will not directly interact with the rest of the Internet, they do not need unique addresses. Although several organizations may be using this same set of addresses, neither of them can *see* the other and there is no problem. In the first case in the bullet point above, even though you do not want to allow most contacts between the internal machines and the Internet, there will be *some* interactions that are desirable and necessary. In the second case, there is no prohibition on such access.

There are two mechanisms that allow a computer with a private address to communicate over the Internet.

#### Proxy servers

A proxy server is a specific type of firewall. The proxy server has an address in the private address space, but also has a second connection and address connected to the Internet. If a user wants to (and is allowed to) communicate with a machine in the Internet, it sends the message to the proxy server, and requests that this message be forwarded to the target machine in the Internet. The proxy server keeps track of this request, and when the answer comes back, it returns the answer to the originating machine.

Proxy servers can also be used if you have a normal IP address. They are used to control what type of traffic goes out onto the Internet, or to simplify a user's interaction with the network. A web proxy server will keep copies of pages requested, and if a second user requests the same page, it simply provides the copy – limiting the number of requests sent to the Internet and therefore reducing external bandwidth requirements. Keeping recently requested pages is called *caching*.

#### Network Address Translation

Network Address Translation (NAT) is normally implemented by having a special box sit between the local network and the Internet. Like the proxy server, it is connected to both the local network where private IP addresses are used, and to the Internet. When a message from the local network bound for the Internet is received by the NAT box, the NAT box sends the message out to the Internet using *its* IP address, and says it is coming from an port number that is unused. When the reply comes in, it is returned to the

originating computer on the local network. A NAT box is similar to a proxy server, but it works for all kinds of traffic, not only a specific kind (such as web traffic) and it does not do any caching.

Both proxy servers and NAT boxes are effectively firewalls and implicitly protect the machines within the local private address spaces from many of the types of attacks that machines with normal IP address are subject to.

### Remote access/management/administration tools

Remote access, remote management and remote administration tools allow you to control your computer remotely, either via a dial-up telephone line or via the Internet. When you are connected to your computer in this way, it is equivalent to sitting at the keyboard.

**Rule 24: If you use remote access facilities to remotely control any computers, make sure that they have robust security (at the very least, excellent usernames and passwords) to ensure that attackers do not use these same tools.**

Remote access tools have many important uses. Among them are:

- They allow you to use your office computer while not at the office. This allows you to use data, applications programs, and network services that are accessible at work.
- They allow you turn over control of your machine to a specialist to diagnose or fix a problem without the specialist having to come to your location.
- They allow multiple people to use an application program that is only installed on one machine.
- They allow systems support personnel to manage multiple servers easily.

Remote access tools also allow an attacker to do all of the same things. In fact, there is often little functional difference between a remote access tool that is sold for the above type of applications (such as pcAnywhere), and the backdoor Trojan (such as NetBus or Back Orifice).

### Malware detectors

It would be nice to assume that if you practice keep all of your software up-to-date, check incoming files for viruses and worms, use secure usernames and passwords, and protect yourself with a robust firewall, then you will be completely safe. To phrase this as a question, if you practice safe computing, will you be safe?

The answer is “probably”. There is always the chance that some sort of problem will hit you before a solution is generally available. It is also possible that occasionally you may do something that is less than 100% safe.

Malware detectors are programs that check your computer to see if there is anything there that looks suspicious, regardless of how it got there. Their functions overlap with virus checkers in some cases, as they will both detect the presence of some types of malware on your disk. Depending on the specific tool, they will check to verify that key system programs have not been surreptitiously changed.

Malware detectors will also look at browser plug-ins and add-ons and try to detect those that are potentially malicious or will violate your privacy. Some malware detectors also include tools to remove an offending program.

### Logs

Logs are an under-utilized and under-appreciated tool in ensuring that your computer is secure. A log is a file on disk into which programs can write messages. Typically a message is written into a log when something interesting happens or if some error occurs.

**Rule 25: System functions and applications logs should be judiciously enabled.**

Examples of “interesting” things include:

- the computer is powered on;
- someone logged onto the computer;
- someone tried to log onto the computer, but had a wrong password;
- an e-mail was received;

- an e-mail send was attempted, but the connection failed;
- there were many errors on a disk, or on a network connection;
- the firewall detected an illegal communication and blocked it;
- the virus checker automatically downloaded a new set of virus signatures;
- a virus scan of all files on your system was run and a virus was detected.

Depending on the program/system, log files can just grow until they are erased, or there may be a new log file created every so often, with the old log files being kept for later review (typically they will have a date in the filename)

In general, there is a separate log file for each application or system function. Sometimes you read a log with any text editor, and sometimes the application or system provides specialized tools to read and format logs.

Logs are very useful and should generally be enabled. However, you need to take care to ensure that you do not enable logging for functions that happen too often, or your system will spend all of its time writing logs and your disk will become clogged with log files.

If you understand what the detailed log entries are saying, you should review them periodically to see if anything unusual is happening. Otherwise, logs should be kept so that in the case of some sort of unusual happening, they may give some hint as to exactly what happened.

## CHAPTER 8. PLATFORM SPECIFIC ISSUES

### Microsoft Windows-based PCs?

#### Strengths and vulnerabilities

The Windows operating system for the Intel x86 (or equivalent) processor is by far the most popular computer system ever built. The capabilities of the operating system and related applications, from an end-user's perspective, are remarkable. There is a vast amount of commercial, shareware, and free software available for it. Although experts are hard to find (as with most systems), there are many people who have reasonable levels of knowledge about these systems. There are many competitors on the hardware side, resulting in much variety and relatively low prices.

From a security point of view, Windows is not quite as attractive. The core operating system was not originally written with either network connectivity or security in mind. The more recent versions (Windows 2000, Windows XP, and later) have addressed many of the original concerns, but security is still lacking and the current changes are of little help to users who are still running older systems. Until recently, Microsoft did not have a strong focus on security, although that is changing, particular with the media attention on bugs and other exploitable flaws in Microsoft operating systems.

The built-in functionality of their systems and applications has often been enhanced at the expense of security. In many cases, to make things *easy* for the novice user, systems are delivered with many sub-systems and capabilities enabled, which makes them available for exploitation. Due to the prevalence of these exposures and the number of installed computers, the Windows-based PC has become a major target of malicious programmers who have churned out viruses, worms, and Trojans by the tens of thousands. The Windows GUI (graphical user interface) is sufficiently user-friendly that the system is now used by millions of people with little technical knowledge or interest. This type of user base, coupled with the vulnerabilities cited above, has made Windows-based systems prone to security problems.

#### How to protect yourself

Virtually all of the rules in this manual apply to Windows systems and security-conscious users should consider each of the recommendations seriously.

---

**Software currency** If you have adequate bandwidth, use Microsoft's Windows's Update site to keep your operating system up-to-date. If reasonable bandwidth is not available, consider using Windows Update for critical security patches (they use far less bandwidth than the larger Service Packs). If Windows Update is not practical, updates can be downloaded from Microsoft's Download Center: (<http://www.microsoft.com/downloads>).

Perhaps your ISP or some other service provider could download them and distribute them locally on CD. Although it takes significant resources, a Windows Update-like service called Software Update Services can be run on a local site for Windows 2000 systems:

(<http://www.microsoft.com/windows2000/windowsupdate/sus/>).

---

**Accounts** For Windows NT, 2000, and XP which support multiple users, you should ensure that there are no unnecessary user accounts set up. In addition, make sure that all users choose robust passwords, as described earlier in Part 2 of this Handbook. Users should only be given the privileges that they require. For example, even if a machine is administered by its' primary user, the user's basic operational account should not have administrator privileges.

---

**File Sharing** If you do not use file sharing or print serving, make sure that the capability is completely disabled. The procedure can be found in Windows Help or within the Microsoft support site; search for "disable file sharing XX" where XX is the version of your system, such as XP or 2000. If you do allow file sharing, make sure you give out no more privileges than necessary.

<b>File System</b>	The FAT and FAT32 file systems historically used by Windows cannot be properly secured, particularly if you are using file sharing. The NTFS file system should be used whenever possible, if there is any network file access. Note that NTFS can not be used in some cases where you have a dual-boot machine or need to access the hard disk from another operating system.
<b>Systems Services</b>	Some systems come with all services enabled in order to allow sophisticated computer-to-computer communications. If you are not in a corporate network, disable the services that you do not need.
<b>Firewalls</b>	Install a software or hardware firewall. Free software versions are available. Keep the firewall up-to-date. Make sure that the firewall is configured to warn you if unusual activities are taking place.
<b>Anti-virus software</b>	Install anti-virus software. If you cannot find freeware that is kept current, you should invest in commercial software. Some virus software companies offer dynamically downloaded free virus checking. Keep the virus signatures up-to-date; some vendors offer daily updates, others provide weekly updates, or longer term. The more current your virus definitions are, the better your system is protected.
<b>Malware detectors</b>	There are programs which will scan your system for all sorts of potentially malicious software. Pest Patrol ( <a href="http://www.pestpatrol.com">http://www.pestpatrol.com</a> ), Lavasoft ( <a href="http://www.lavasoftusa.com/software/adawareplus/">http://www.lavasoftusa.com/software/adawareplus/</a> ) and SpybotSD ( <a href="http://www.safer-networking.org">http://www.safer-networking.org</a> ) all have free programs that detect various malware.

**Security Review** If you are a non-technical user with no support organization available to help you, take a look at Microsoft's recommendations for home users:  
<http://www.microsoft.com/security/home>  
or <http://www.microsoft.com/protect/>.

If you are an IT professional, go to:  
<http://www.microsoft.com/technet/security>.  
If you have a newer system, consider running the Microsoft Baseline Security Analyzer (MBSA) that covers Windows 2000 and XP systems.

## Macintosh

### Strengths and vulnerabilities

Historically, the Apple Macintosh computer and operating system has been far less prone to security problems than the Windows PC. Moreover, since there are far fewer Mac users than there are PC users, malicious attackers have not been as interested in targeting them. Perhaps the largest vulnerability is that, for these reasons, Mac users often *think* they are safe and do not bother to take precautions. MacOS systems prior to MacOS X used a proprietary operating system. MacOS X is based on the FreeBSD Unix system, and should be considered a specialized Unix system with regard to security (see next section on Unix). For MacOS X, there are many system services bundled within the core system, but they are all shipped disabled.

### How to protect yourself

**Software currency** Make sure that your system is full patched. Go to: <http://www.apple.com> and click on support. As with Windows systems, there is a good chance that an unpatched system will be infiltrated within hours or days, particularly if it is permanently attached to a network.

---

<b>Accounts</b>	Make sure that all accounts that you do not need are disabled or deleted. In particular, make sure there are no <i>Guest</i> accounts without a password. Limit administrative privileges to accounts that actively need them and do not use an administrative-capable account for your routine work.
<b>File Sharing</b>	Disable file sharing if you are not using it. If you are using file sharing, make sure the privileges are granted at minimum level required.
<b>Services</b>	Do not enable services that you do not need. If you enable them temporarily, but will not use them often, disable them when you are through.
<b>New applications</b>	If you install new network-oriented applications, particularly those originally designed for Unix, be aware that they may be vulnerable in ways that were uncommon in systems built prior to MacOS X.
<b>Firewalls</b>	Install a software or hardware firewall. Keep it up-to-date. Make sure that the firewall is set to warn you if unusual activities take place.
<b>Anti-virus software</b>	Install anti-virus software. If you cannot find freeware that is kept current, you should invest in commercial software. Keep the virus signatures up-to-date. The more current your virus definitions are, the better your system is protected.

---

## Unix, Linux, and Related Systems

### Strengths and vulnerabilities

Unix systems have historically been used as servers (both for system services and for multi-user computing) and as workstations in computer science and physical science environments. Over the last decade, they have made some modest inroads against Windows and Macintosh systems as single-user workstations in other environments.

With the recent popularity of Linux, this phenomenon has spread, partly because the system is so attractive and partly because Linux is viewed as a (free) replacement for Windows. This latter trend is probably stronger in the developing world than it is in developed countries, due to the higher relative cost of software compared to salaries in developing countries. Traditionally, Unix's strengths have been its flexibility coupled with the impressive base of user and corporate-developed software that has grown over the years.

Unfortunately, Unix's flexibility and power has not been accompanied by a user-friendly front-end (from a novice user's point of view). As a result, when these systems have been used as workstations for those who do not wish to become Unix experts, strong systems support staff were needed. To some extent, this is being addressed, with MacOS X being the best example. However, the foundation of the system is still complex, and there are many opportunities for a naive user to leave doors open for security breaches. Although Unix systems have been relatively virus free, they have the distinction of hosting some of the earliest worms and Trojans; these are still major potential problems.

### How to protect yourself

The following comments augment information supplied in the rest of this Handbook. Virtually all of the items in the preceding seven chapters apply to Unix, Linux and related systems, and must be addressed if your computer is to be moderately secure. This section focuses primarily on single-user workstations. Those responsible for servers should read Part 5 of this Handbook.

---

<b>Multiple Unix Variants</b>	Because there have been a variety of versions of Unix-like operating systems, many pre-installed security mechanisms are vendor-specific. It's particularly important to read all of the manuals for your vendor's version of Unix. Several good books, web sites, and mailing lists devoted to Unix security are listed in Annexes 2-5.
-------------------------------	--

---

**Software currency**

It is imperative that software be kept current, and that all security patches be applied quickly. Details on where to get updates and how to apply them vary from system to system.

---

**User Privileges**

The user *root* (uid 0) is the superuser and usually has the ability to modify every aspect of the system. Accordingly, protecting the root account and processes that run with root privileges is a critical aspect of Unix security. Avoid using the root account for routine activities, and disable logins by root. When you must use root, use the superuser command (*su*, or a variation like *sudo*) to change from your normal user account to root.

If you have multiple users on your system, consider using access control lists of other mechanisms to limit the file access that these users have.

When possible, run network services as a non-root user.

Never unpack or compile new software as *root*. It's often possible to compile software in a *chroot* environment to protect yourself against some kinds of Trojan horses.

---

**Remote disk mounts**

If you use some mechanism to allow remote access to your disks (whether to other Unix systems or to PCs) use robust passwords and, when possible, limit access to the files that the applications demand.

---

**System Services**

Many Unix systems are shipped with a large variety of system services including FTP servers, web servers, and mail servers. In many cases, these systems are active and operating by default. All network-based services that you are not using should be disabled. Some people feel that since the service is there, it should be used, even though they do not have the technical expertise to manage it securely. This is a *big* mistake and such services should not be run on user workstations without good reason and adequate support.

Many network services are started by the *inetd* (or *xinetd*) daemon. Examine the configuration file(s) used by this daemon and disable any services that you do not need. Other network services are started at system boot by files in the */etc/init.d* or */etc/rc\*.d* directories on in the files */etc/rc* and */etc/rc.local*. Disable any services that you do not use. Pay particular attention to services that may provide outsiders with information about your system or its users, such as *fingerd*.

If you run anonymous FTP services, use an up-to-date version of the FTP daemon. Don't provide your real */etc/passwd* file in the FTP area. Make sure that */etc/ftpusers*, the list of users who cannot connect by FTP, includes at least *root*, *uucp*, *bin*, and any other account that does not belong to a human being. Be wary of directory permissions and ownership in the FTP area; configure "incoming" directories to prevent downloads and "outgoing" directories to prevent uploads. Scan your FTP logs regularly.

- 
- Firewall** Every Unix system should run its own host-based packet-filtering firewall. Consult vendor documentation to determine if your system has a firewall and how to use it. Typical firewall configuration tools include *ipfw*, *ipchains*, and *iptables*. These firewalls should be configured to block all packets by default, and to allow only packets destined for services that you intend to provide.
- 
- Default Accounts** Many Unix systems come with several default accounts that are used to separate process or file ownership privileges, such as *daemon*, *bin*, *uucp*, etc. Make sure that the encrypted password entry for all of these accounts begins with a "\*" character so that no possible password can be used to access the account. Only the root account should have a valid password. No one can log into the other accounts (although root can still assume their privileges with the *su* command if necessary).
- 
- Malware detectors** There are a number of tools which help a Unix administrator ensure that there is no malicious software on their system. One of the oldest is Tripwire, which verifies that the critical system utilities (and other files) have not been surreptitiously altered.

## ADDENDUM 1. INTRODUCTION TO ENCODING AND ENCRYPTION

Encoding and Encryption are techniques that transform a string of characters into some other form for a specific reason. In the sense that they are used in computing, encoding is a transformation that alters the look of the object, so that the result meets some specific criteria. Encryption is a transformation designed to disguise or hide the original contents.

### Encoding

Encoding changes the format of an object to meet some criteria. It is a reversible process, so that the encoded format can later be decoded to recover the original object.

#### The Encoding Process

Let us say that you want to send a message consisting of a normal English language sentence:

SECURITY IS IMPORTANT.

However, there is a restriction that you may only send the decimal digits: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

To do this, we use a simple set of rules:

Instead of A, send the digits 01;  
 Instead of B, send the digits 02;  
 Instead of C, send the digits 03;  
 Instead of D, send the digits 04;  
 Instead of E, send the digits 05;  
 .....  
 Instead of X, send the digits 24;  
 Instead of Y, send the digits 25;  
 Instead of Z, send the digits 26;  
 Instead of the space character,  
 send the digits 27;  
 Instead of the period character,  
 send the digits 28.

We take the original sentence, and replace each character with its code:

19 replaces the S  
 05 replaces the E  
 03 replaces the C and so forth

We can now send the string:

19050321180920252709192709131615182001142028.

If we put some spaces in the previous line so it is more legible, it looks like this:

19 05 03 21 18 09 20 25 27 09 19 27 09 13 16 15 18  
 20 01 14 20 28.

When the message is received, the recipient does a reverse translation:

S replaces the 19

E replaces the 05

C replaces the 03 and so forth resulting in the original sentence.

#### Encoding Applications

The main application of encoding that we will consider is the transmission of e-mail attachments. E-mail was originally designed for sending English-language text. It was based on the ASCII character set which allows 128 unique characters. 128 is sufficient for representing the 26 letters of the English alphabet in upper and lower case, the 10 digits, a number of special characters (such as comma, period, brackets, etc.) and a variety of control characters (such as tab and end-of-line).

Unfortunately, many languages include more characters than English. Programs, word processing files, pictures, and many other types of files are composed of 8-bit bytes which allow 256 unique characters. None of these could be sent in e-mail.

To overcome this problem, the concept of attachments was developed, in which the file to be transmitted would first be encoded so that it would only contain the legal ASCII characters. This process is similar to how our sample sentence was encoded using only digits. As with our sample, the resultant encoded message is longer than the original, but it can be transmitted legally, and, when received, decoded into its original form.

### Unicode

Unicode is a method of encoding all characters used in all commonly used languages so that computers may uniformly handle them. Details are available through the Unicode Consortium (<http://www.unicode.org>), in brief:

“Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one. Before Unicode was invented, there were hundreds of different encoding systems for assigning these numbers. No single encoding could contain enough characters, for example, the European Union alone requires several different encodings to cover all its languages. Even for a single language like English, no single encoding was adequate for all the letters, punctuation, and technical symbols in common use.

These encoding systems also conflict with one another. That is, two encodings can use the same number for two different characters, or use different numbers for the same character. Any given computer (especially a server) needs to support many different encodings, yet whenever data is passed between different encodings or platforms, that data always runs the risk of corruption. Unicode is changing all that!

Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others.”

## Encryption

Encryption is similar to encoding in that the process transforms some original text or object into another form. In this case, the intent is to hide the original contents.

There are three types of encryption that we will be looking at:

- Symmetric Encryption
- Public-key Encryption
- One-way Hash Encryption

### Symmetric Encryption

In its simplest form, symmetric encryption is similar to encoding. The characters in the original object are transformed. A very simple-minded encryption algorithm (rules governing the process) is to take each alphabetic character and replace it with 1 character higher. So:

A is replaced by B  
 B is replaced by C  
 C is replaced by D  
 .....  
 X is replaced by Y  
 Y is replaced by Z  
 Z is replaced by A (at the end of the alphabet, it loops back to the beginning)

If we use this algorithm, our sample sentence becomes (ignoring the space and period in this simple case):

TFDVSJUZ JT JNQPSUBOU

The message is now disguised. The recipient will do the reverse translation, changing each letter by using the *previous* letter and will obtain the original sentence.

Instead of shifting each character 1 place, we could have shifted them some other number of characters. As long as the recipient knows the number of shifts, they can decrypt the message.

The number of shifts is called the encryption *key*. This same number is used to encrypt the message, and later decrypt it. Julius Caesar used this encryption method to keep messages he sent secret (he used a key of 3).

With this simple algorithm, if the message is intercepted *and* the interceptor understood the concept of encryption, he or she might be able to guess the contents by trying various shifts. If the algorithm was more complex than simply shifting each letter by the same amount, it would be more difficult to decipher. Until recently, many encryption algorithms were just such shifting algorithms.

Today, instead of shifting letters, we use mathematical formulas to encrypt messages. We still use a key and this key is part of the formula to perform the encryption. If you want to decrypt the message, you need the key. If you don't have the key, you could, of course, try various keys until the message made sense. If the key was restricted to the numbers from 1 to 10, this guessing would not take very long. If it were allowed values from 1 to 100, it would probably take longer. Today, keys typically are 128-bit binary numbers. That is equivalent to about 340,000,000,000,000,000,000,000,000,000,000 possible choices and guessing is not practical.

Symmetrical encryption is used when it makes sense for both the sender and recipient to use the same key (that is, they need to agree to it ahead of time). It is used for encrypting messages while they are being transmitted, over a wireless link, for example, and for encrypting information on disk so that others cannot read it. In the latter case, if you lose the key, the data is essentially lost!

### Public-key Encryption

Public key encryption is similar to symmetrical encryption with one major exception. Instead of one key, there are two. A different key is used to encrypt the message than is used to decrypt it. In a typical use, the first key is made public and anyone can learn it. If you want to send me a private message, you use my *public key* that I have given to everyone to encrypt it. To decrypt the message, my *private key* (which is different from the my public key) is needed, and I do not share that key with anyone else. If your message is intercepted, no one else can read it.

Note that in this simple case, I cannot be sure who sent me the message, because anyone might have my public key, but you can be reasonably sure that only I can read it.

Public/Private keys can also be used in reverse. In this case you encrypt the message with your private key, and *anyone* who has your public key can decrypt it.

### One-way Hash Encryption

You can think of a one-way Hash encryption as a type of public-key encryption for which no one has the private key. So things can be encrypted, but not decrypted. It is different in that the encrypted message is typically relatively short. A common one-way hash encryption algorithm is called MD5. The output of the MD5 algorithm is always 128 bits (16 bytes). If you create a hash code for two different things, the chances are virtually zero that the two hash codes will be the same.

There are two prime uses of such a code:

---

**Authentication** You can take a long document or a program, compute the MD5 code for it, and keep the code in a safe place. Later, you can go back and compute the code again. If the new code is different from the original one, you will know that the document or program has been changed. Even a tiny change in a large document or program will result in a markedly different MD5 code.

---

**Storing passwords** In many systems, when a user sets a password, it is encrypted using MD5 (or a similar algorithm) and that encrypted version is stored. When the user later attempts to sign on, what they enter is again encrypted, and compared to the one on disk. If they match, you know the password was correct. Note that it is not possible to decrypt the password if the user forgets it – a new one must be set. This method is used because it never allows your password to be seen in its original form.

Unfortunately, there is still one problem and this is the reason why one should not use passwords that are short, simple, or guessable words: if you obtain a list of encrypted passwords (from a system that you broke into), it is easy to encrypt all sorts of “easy” passwords to see if the encrypted versions match those in the password table.

## Digital Signatures

If I want to send you a message, and ensure that you know that I was the one who sent it, I can use a combination of the encryption techniques:

- I compose the message, and I use MD5 to create a hash code for the message.
- I encrypt the hash code using my private key.
- I send you the message, and the encrypted hash code.
- You receive the message.
- You decrypt the hash code using my public key, which will result in the original hash code.
- You take the text of the message that I sent, and calculate an MD5 hash code from that.
- If the two hash codes are identical, then you can be sure that the message has not been changed since I sent it (otherwise it would result in a different hash code) and that I was the one who sent it (otherwise my public key would not have allowed you to decrypt the original hash code).

The *Digital Certificates* used by web browsers for secure authentication rely on digital signature techniques such as this one.

## ADDENDUM 2.

### TCP/IP

TCP/IP (Internet Protocol) is the protocol (set of rules) governing all messages sent over the Internet. Although a typical user does not need to know anything about TCP/IP to use the Internet, one does need an overview to configure firewalls and to understand some of the other threats on the Internet. What follows is a very simplistic description of TCP/IP. If you are already familiar with the TCP/IP protocol, you probably do not need to read this chapter.

#### Internet Addressing

Every device on the Internet has an IP address. In general, this address uniquely defines that device, just as your mailing address on an envelope uniquely defines your home. Addresses in the current version of TCP/IP (known as IPv4) are 32-bit binary numbers, so there are  $2^{32} = 4,294,967,296$  possible addresses. To make it easier to represent and remember, the 32-bit binary number is broken up into 4 8-bit sections. Because  $2^8 = 256$ , each 8-bit section can have a value from 0 to 255. These 4 numbers are normally shown one after each other, connected by periods. So the lowest Internet address is 0.0.0.0 and the highest one is 255.255.255.255. A typical IP address might be 24.200.195.15. Devices called *routers* on the Internet keep track of where each IP address is and how to get to it.

#### Domain Name Service

Because long strings of numbers are not easy to remember, many computers on the Internet are given alphabetic names (called a *hostname*). An example of such a name is `www.infodev.org`. When you enter this name into your web browser, for example, your computer sends a message to a special service called the *Domain Name Service* or DNS. The DNS knows how to translate alphabetic names into numeric ones - 192.86.99.121 in this case. DNS also allows a web server to be moved to a different location on the Internet. The owner informs the DNS of the new address, but users can still use the original hostname.

#### IP: Internet Protocol

When data is sent over the Internet, it is sent in blocks of characters called a *packet* or *datagram*. The IP in

TCP/IP stands for Internet Protocol and the Internet Protocol defines how the packet looks inside. The IP packet contains a number of pieces of information. Among them are:

- the size of the packet;
- the IP address of the sender;
- the IP address where the packet is being sent;
- the type of packet.

When a packet leaves your computer, it is sent to the nearest router which attempts to send it to the next router along the way to its destination. If, due to congestion or some other problem, the packet cannot get delivered, it is simply ignored. For this reason, IP is called an *unreliable* protocol. Although in theory IP is unreliable, in most cases, the Internet delivers all the packets that are sent.

There are a number of different types of packets that can be sent, but there are only two that we will look at here. They are TCP and UDP.

#### TCP: Transmission Control Protocol

TCP is the protocol that is used for most messages, including the web (HTTP), File Transfer Protocol (FTP) and e-mail. In addition to the data being sent, the TCP packet includes:

- a 16-bit sending port number;
- a 16-bit receiving port number;
- sequencing information;
- acknowledgement information.

Because a single computer typically has just one IP address, the port number is used to indicate what program within the computer is sending or receiving the message. This is what allows you to have several web browser windows open on your computer and to have the pages that you request go back to the correct window. For a program to receive a TCP message, it must be *listening* on the correct port. Typically, a specific port is used for each type of application. For instance, a web server usually listens on port 80. When you open a browser window, it typically picks a semi-random port number (by convention higher than 1023) as its port, and this is the port that it listens on. Because IP packets are limited in length, and the data transmitted by

an application program may be much larger, the data can be chopped up into smaller segments. Each segment is sent in its own TCP packet. For various reasons, some packets may arrive faster than others, which means that they may arrive out of order. The sequencing information allows the receiving program to re-assemble the segments in the correct order. Since IP is potentially unreliable, it is possible that one of the segments never arrives. In this case, the receiving program will notice that there is a gap in the sequence and it can request that the missing packet be resent.

When a program sends a TCP packet, it expects the receiving program to acknowledge it. If an acknowledgement does not arrive in a reasonable time, the packet can be re-transmitted. Because of the sequence numbers and the acknowledgements, TCP is a *reliable* protocol. When it is used, the user application can be sure that if there is an error in transmission or reception, the application will be informed.

#### **UDP: User Datagram Protocol**

UDP is a simple format to allow data to be transmitted. Each UDP packet includes some information in addition to the data. These include:

- a 16-bit sending port number, and
- a 16-bit receiving port number.

Just as with TCP, because port numbers are used, there can be several program sending or receiving UDP streams in parallel. Also like TCP, to receive a message, the program must be listening on the correct port. There are no provisions for sequencing or acknowledgement in UDP, so it (like IP) is an unreliable protocol. In theory, messages can be lost. It is used in cases where it either does not matter if an occasional message is lost, or if there is a simple way to recover from the lost message. Because there are no acknowledgements or sequencing, it uses far fewer resources.

## ADDENDUM 3. MINI-GLOSSARY OF TECHNICAL TERMS

### Definitions Related to Security

**Attachment** An attachment is a method by which text and images can be sent via e-mail. Any non-text file (which could be a program or a picture or a video) is converted (“encoded”) into a printable form and inserted into the text message. Specifically, anything stored in your computer is composed of zeros and ones. Encoding, in its simplest form, would send the zeros and ones as printable characters.

**Backdoor** A way to bypass the normal login security and gain control of a computer without obtaining the owner’s consent. If a backdoor is installed on a network-attached computer, a person anywhere on the Internet may be able to gain control of your computer without your knowledge or approval.

**Backup** The process of copying computer files to some other location either on the computer, or on storage devices that may be separated from the computer. Backups allow you to recover data in the event that the originals are no longer available (for reasons ranging from accidental deletion to physical damage, theft or other loss).

**Buffer Overflow** A software bug that occurs when a program moves data into a space in memory, but there is not enough room. The program may discard characters to try to make space for the new data.<sup>34</sup>

Destroying these characters can cause all sorts of problems, and often can allow things to happen which affect the integrity or security of the program. Buffer overflows can be avoided (if you are programming) by checking

that there is sufficient spaced in memory before doing a move.

**Cookie** A file that is written to or read from your hard disk at the request of a remote web site. The web site requests that the file be written and reads it later. As a simple example, if you tell a web site what your username is, it can request that this information be written to your disk. When you go back to that web site, it reads the cookie and knows what your username is.

**Daemon** A small program that runs all of the time waiting for someone to ask it to do something – often such requests may be made remotely over the network.

**Denial-of-Service** A Denial-of-Service attack is when computers on the Internet are bombarded with (garbage) messages to such a great extent that they spend all of their time responding to these messages. Real user traffic can no longer get through.

**E-mail** The computer-based equivalent of postal mail – e(lectronic)-mail. Properly addressed e-mail can be sent and received by anyone connected to the Internet. From the perspective of the Internet, all e-mail is composed of printable text (ASCII) messages.

**Encryption** Encryption is a way to disguise information so that it cannot be read easily, except by the intended recipient. In the simplest case, there is a “key” in conjunction with a set of rules that is used to disguise that information. It can only be read after being decrypted, and to decrypt it, you would need to know the proper “key” and the appropriate rules.

<sup>34</sup> For example, the program might move 100 characters into an area that is only 80 characters long. Assume that the programmer is moving the data into an area starting at location 1001 in memory. The first 80 characters go just where they should – into locations 1001-1080, but the last 20 characters go into locations 1081-1100 – they overlap on top of whatever was there before (since the maximum move was supposed to be just 80 characters).

---

**Firewall** Firewalls can block transmissions between you and the outside world that are unexpected or disallowed. Firewalls have two forms: a firewall may be software program running on your computer or it may be a separate piece of hardware that watches what is being sent and received over a network.

---

**HTML** HTML is short for **HyperText Markup Language**. A mark-up language allows commands or instructions embedded in the text to be displayed and printed. It is essentially a set of instructions that tells a web browser or mail program how to display text and images. It can also give other instructions to the browser/mail program. An example of a mark-up language is:

This sentence is <<Start Bold>>very<<End Bold>> short.

When the sentence is displayed, the words within the << >> are taken as instructions on what to do. As a result, the sentence would be displayed as: This sentence is **very** short.

---

**Identity theft** Identity theft occurs when someone gathers enough information about you to convince others (such as banks, stores or governments) that they *are* you.

---

**Keyboard logger** A program that captures everything that is typed on a keyboard. The data can be written to disk or sent to someone else via the Internet. If a keyboard logger is installed on a computer, everything that is entered on the computer, including usernames and passwords, can be captured, just as if someone was looking over your shoulder while you typed!

---

**Open Source** Programs that are distributed in source format under conditions that allow free modification and distribution. Since the source code is available, people can see how it works and are able to change it. The authors of Open Source programs often encourage other programmers to participate in the further development of the programs. Open Source also includes software that is given away for free and many Open Source programs, both free and for sale, offer functionality that is similar to proprietary programs that may cost a substantial amount of money. Sometimes Open Source programs are incorporated into fee-based programs in special licensing arrangements. See [www.opensource.org](http://www.opensource.org) and [www.fsf.org](http://www.fsf.org) for additional information.

---

**Spam** Advertising or other e-mail sent to you without your requesting it.

---

**URL** Universal Resource Locator – a generalized address to locate something in the Internet. Examples are <http://www.infodev.org/> and <mailto:security-handbook@worldbank.org>

---

**Username/ password** A name and a secret password that identifies a user to a computer system or a web site.

---

**Virus** The term “virus” has a very specific meaning that will be defined and discussed in more detail later. For the present, it will be used to describe a family of programs (including viruses, worms and Trojans) that can unexpectedly show up in your computer, may spread to other computers, and can do significant harm. This harm includes, but is not limited to, destroying files and data.