**Computer Systems Technology**

==========

**NIST Special Publication**

# Computer Viruses and Related Threats: A Management Guide

**John P. Wack**

**Lisa J. Carnahan**

# Table of Contents

Executive Summary . . . . . v

1. Introduction . . . . . 1-1
   1.1 Audience and Scope . . . . . 1-1
   1.2 How to Use This Guide . . . . . 1-2

2. A Brief Overview on Viruses and Related Threats . . . . . 2-1
   2.1 Trojan Horses . . . . . 2-1
   2.2 Computer Viruses . . . . . 2-2
   2.3 Network Worms . . . . . 2-4
   2.4 Other Related Software Threats . . . . . 2-6
   2.5 The Threat of Unauthorized Use . . . . . 2-6

3. Virus Prevention in General . . . . . 3-1
   3.1 User Education . . . . . 3-2
   3.2 Software Management . . . . . 3-3
   3.3 Technical Controls . . . . . 3-5
   3.4 General Monitoring . . . . . 3-6
   3.5 Contingency Planning . . . . . 3-7

4. Virus Prevention for Multi-User Computers and Associated Networks . . . . . 4-1
   4.1 General Policies . . . . . 4-1
   4.2 Software Management . . . . . 4-2
   4.3 Technical Controls . . . . . 4-3
   4.4 Monitoring . . . . . 4-5
   4.5 Contingency Planning . . . . . 4-7
   4.6 Associated Network Concerns . . . . . 4-8

5. Virus Prevention for Personal Computers and Associated Networks . . . . . 5-1
   5.1 General Policies . . . . . 5-2
   5.2 Software Management . . . . . 5-2
   5.3 Technical Controls . . . . . 5-3
   5.4 Monitoring . . . . . 5-5
   5.5 Contingency Planning . . . . . 5-6
   5.6 Associated Network Concerns . . . . . 5-7

References . . . . . A-1

Suggested Reading . . . . . B-1

# Executive Summary

Computer viruses and related threats represent an increasingly serious security problem in computing systems and networks. This document presents guidelines for preventing, deterring, containing, and recovering from attacks of viruses and related threats. This section acquaints senior management with the nature of the problem and outlines some of the steps that can be taken to reduce an organization's vulnerability.

## What Are Computer Viruses and Related Threats?

Computer viruses are the most widely recognized example of a class of programs written to cause some form of intentional damage to computer systems or networks. A computer virus performs two basic functions: it copies itself to other programs, thereby *infecting* them, and it executes the instructions the author has included in it. Depending on the author's motives, a program infected with a virus may cause damage immediately upon its execution, or it may wait until a certain event has occurred, such as a particular date and time. The damage can vary widely, and can be so extensive as to require the complete rebuilding of all system software and data. Because viruses can spread rapidly to other programs and systems, the damage can multiply geometrically.

Related threats include other forms of destructive programs such as Trojan horses and network worms. Collectively, they are sometimes referred to as *malicious software*. These programs are often written to masquerade as useful programs, so that users are induced into copying them and sharing them with friends and work colleagues. The malicious software phenomena is fundamentally a people problem, as it is authored and initially spread by individuals who use systems in an unauthorized manner. Thus, the threat of unauthorized use, by unauthorized *and* authorized users, must be addressed as a part of virus prevention.

## What Are the Vulnerabilities They Exploit?

Unauthorized users and malicious software may gain access to systems through inadequate system security mechanisms, through security holes in applications or systems, and through weaknesses in computer management, such as the failure to properly use existing security mechanisms. Malicious software can be copied intentionally onto systems, or be spread when users unwittingly copy and share infected software obtained from public software repositories, such as software bulletin boards and shareware. Because malicious software often hides its destructive nature by performing or

claiming to perform some useful function, users generally don't suspect that they are copying and spreading the problem.

## Why Are Incidents of Viruses and Related Threats On the Rise?

Viruses and related threats, while not a recent phenomena, have had relatively little attention focused on them in the past. They occurred less frequently and caused relatively little damage. For these reasons, they were frequently treated lightly in computer design and by management, even though their potential for harm was known to be great.

Computer users have become increasingly proficient and sophisticated. Software applications are increasingly complex, making their bugs and security loopholes more difficult to initially detect and correct by the manufacturer. In conjunction with these two factors, some brands of software are now widely used, thus their bugs and security loopholes are often known to users. With the widespread use of personal computers that lack effective security mechanisms, it is relatively easy for knowledgeable users to author malicious software and then dupe unsuspecting users into copying it.

## Steps Toward Reducing Risk

Organizations can take steps to reduce their risk to viruses and related threats. Some of the more important steps are outlined below.

- Include the damage potential of viruses, unauthorized use, and related threats in risk analysis and contingency planning. Develop a plan to deal with potential incidents.
- Make computer security education a prerequisite to any computer use. Teach users how to protect their systems and detect evidence of tampering or unusual activity.
- Ensure that technically oriented security and management staff are in place to deal with security incidents.
- Use the security mechanisms that exist in your current software. Ensure that they are used correctly. Add to them as necessary.
- Purchase and use software tools to aid in auditing computing activity and detecting the presence of tampering and damage.

# 1. Introduction

This document provides guidance for technical managers for the reduction of risk to their computer systems and networks from attack by computer viruses, unauthorized users, and related threats. The guidance discusses the combined use of policies, procedures, and controls to address security vulnerabilities that can leave systems open to attack. The aim of this document is not to provide solutions to the wide range of specific problems or vulnerabilities, rather it is to help technical managers administer their systems and networks such that manifestations of viruses and related threats can be initially prevented, detected, and contained.

## 1.1 Audience and Scope

This document is intended primarily for the managers of multi-user systems, personal computers, and associated networks, and managers of end-user groups. Additionally, the document is useful for the users of such systems. The document presents an overview of computer viruses and related threats, how they typically work, the methods by which they can attack, and the harm they can potentially cause. It then presents guidance in the following areas:

- *Multi-User Systems and Associated Networks* - with guidance directed at managers of medium to small systems (as opposed to mainframes that already provide generally effective security controls or are by their nature more secure) and associated wide area and large local area networks, as well as managers of end-users of such systems

- *Personal Computer Systems and Networks* - guidance is directed at those responsible for the management of personal computers and personal computer networks, as well as the managers of personal computer end-users

Within these general categories, individual computing environments will vary widely, from size of computer to user population to type of software and computing requirements. To accommodate these differences, the guidance presented here is general in nature. It attempts to address computer security problems and vulnerabilities that are likely to be found in most computing environments. This document does not address problems directly related to specific brands of software or hardware. A reading list at the end of the document contains references and pointers to other literature that address specific systems and software.

Recommended control measures are grouped according to categories that include general policies and procedures, education, software management, technical controls, monitoring, and contingency planning. The guidance emphasizes the need for a strong security program as a means for protection from manifestations of viruses and related threats, and as a means for providing detection, containment, and recovery. Such a security program requires personal involvement on the part of management to ensure that the proper policies, procedures, and technical controls exist, and that users are educated so that they can follow safe computing practices and understand the proper actions to take if they detect the presence of viruses or related threats. The guidelines recommend that network managers, multi-user system managers, end-users, and end-user managers work with each other and approach virus protection from an organizationally consistent basis.

## 1.2 How to Use This Guide

This document is divided into five chapters and two appendices. Chapter 2 describes in general how viruses and related software operate, the vulnerabilities they exploit, and how they can be introduced into systems and networks. Chapter 3 discusses general protection strategies and control measures that apply to technical and end-user management in general; this is done so that the same guidance need not be repeated for each of the succeeding chapters that deal with specific environments. Chapters 4 and 5 present guidance specific to multi-user and personal computer environments, respectively. The guidance in these chapters is directed at the respective technical managers and managers of associated networks, as well as the managers of end-user groups that use such systems and networks. It is recommended that all readers, regardless of their management perspective, examine Chapters 3, 4, and 5 to gain a fuller appreciation of the whole environment with regard to threats, vulnerabilities, and controls.

Appendix A contains document references, while Appendix B contains a reading list with references to general and specific information on various types of viruses, systems, and protective measures. Readers can use these documents to obtain information specific to their individual systems and software.

# 2. A Brief Overview on Viruses and Related Threats

The term *computer virus* is often used in a general sense to indicate any software that can cause harm to systems or networks. However, computer viruses are just one example of many different but related forms of software that can act with great speed and power to cause extensive damage - other important examples are Trojan horses and network worms. In this document, the term *malicious software* refers to such software.

## 2.1 Trojan Horses

A Trojan horse[1] program is a useful or apparently useful program or command procedure containing hidden code that, when invoked, performs some unwanted function. An author of a Trojan horse program might first create or gain access to the source code of a useful program that is attractive to other users, and then add code so that the program performs some harmful function in addition to its useful function. A simple example of a Trojan horse program might be a calculator program that performs functions similar to that of a pocket calculator. When a user invokes the program, it appears to be performing calculations and nothing more, however it may also be quietly deleting the user's files, or performing any number of harmful actions. An example of an even simpler Trojan horse program is one that performs only a harmful function, such as a program that does nothing but delete files. However, it may appear to be a useful program by having a name such as CALCULATOR or something similar to promote acceptability.

Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly. For example, a user of a multi-user system who wishes to gain access to other users' files could create a Trojan horse program to circumvent the users' file security mechanisms. The Trojan horse program, when run, changes the invoking user's file permissions so that the files are readable by any user. The author could then induce users to run this program by placing it in a common directory and naming it such that users will think the program is a useful utility. After a user runs the program, the author can then access the information in the user's files, which in this example could be important work or personal information. Affected users may not notice the changes for long periods of time unless they are very observant.

---

[1] named after the use of a hollow wooden horse filled with enemy soldiers used to gain entry into the city of Troy in ancient Greece.

An example of a Trojan horse program that would be very difficult to detect would be a compiler on a multi-user system that has been modified to insert additional code into certain programs as they are compiled, such as a login program. The code creates a *trap door* in the login program which permits the Trojan horse's author to log onto the system using a special password. Whenever the login program is recompiled, the compiler will always insert the trap door code into the program, thus the Trojan horse code can never be discovered by reading the login program's source code. For more information on this example, see [THOMPSON84].

Trojan horse programs are introduced into systems in two ways: they are initially planted, and unsuspecting users copy and run them. They are planted in software repositories that many people can access, such as on personal computer network servers, publicly-accessible directories in a multi-user environment, and software bulletin boards. Users are then essentially duped into copying Trojan horse programs to their own systems or directories. If a Trojan horse program performs a useful function and causes no immediate or obvious damage, a user may continue to spread it by sharing the program with other friends and co-workers. The compiler that copies hidden code to a login program might be an example of a deliberately planted Trojan horse that could be planted by an authorized user of a system, such as a user assigned to maintain compilers and software tools.

## 2.2 Computer Viruses

Computer viruses, like Trojan horses, are programs that contain hidden code which performs some usually unwanted function. Whereas the hidden code in a Trojan horse program has been deliberately placed by the program's author, the hidden code in a computer virus program has been added by another program, that program itself being a computer virus or Trojan horse. Thus, computer viruses are programs that copy their hidden code to other programs, thereby *infecting* them. Once infected, a program may continue to infect even more programs. In due time, a computer could be completely overrun as the viruses spread in a geometric manner.

An example illustrating how a computer virus works might be an operating system program for a personal computer, in which an infected version of the operating system exists on a diskette that contains an attractive game. For the game to operate, the diskette must be used to boot the computer, regardless of whether the computer contains a hard disk with its own copy of the (uninfected) operating system program. When the computer is booted using the diskette, the infected program is loaded into memory and begins to run. It immediately searches for other copies of the operating system program, and finds one on the hard disk. It then copies its hidden code to the program on the hard disk. This happens so quickly that the user may not notice the

slight delay before his game is run. Later, when the computer is booted using the hard disk, the newly infected version of the operating system will be loaded into memory. It will in turn look for copies to infect. However, it may also perform any number of very destructive actions, such as deleting or scrambling all the files on the disk.

A computer virus exhibits three characteristics: a *replication* mechanism, an *activation* mechanism, and an *objective*. The replication mechanism performs the following functions:

- searches for other programs to infect
- when it finds a program, possibly determines whether the program has been previously infected by checking a flag
- inserts the hidden instructions somewhere in the program
- modifies the execution sequence of the program's instructions such that the hidden code will be executed whenever the program is invoked
- possibly creates a flag to indicate that the program has been infected

The flag may be necessary because without it, programs could be repeatedly infected and grow noticeably large. The replication mechanism could also perform other functions to help disguise that the file has been infected, such as resetting the program file's modification date to its previous value, and storing the hidden code within the program so that the program's size remains the same.

The *activation* mechanism checks for the occurrence of some event. When the event occurs, the computer virus executes its *objective*, which is generally some unwanted, harmful action. If the activation mechanism checks for a specific date or time before executing its objective, it is said to contain a *time bomb*. If it checks for a certain action, such as if an infected program has been executed a preset number of times, it is said to contain a *logic bomb*. There may be any number of variations, or there may be no activation mechanism other than the initial execution of the infected program.

As mentioned, the objective is usually some unwanted, possibly destructive event. Previous examples of computer viruses have varied widely in their objectives, with some causing irritating but harmless displays to appear, whereas others have erased or modified files or caused system hardware to behave differently. Generally, the objective consists of whatever actions the author has designed into the virus.

As with Trojan horse programs, computer viruses can be introduced into systems deliberately and by unsuspecting users. For example, a Trojan horse program whose purpose is to infect other programs could be planted on a software bulletin board that permits users to upload and download programs. When a user downloads the program and then executes it, the program proceeds to infect other programs in the user's system. If the computer virus hides itself well, the user may continue to spread it by copying the infected program to other disks, by backing it up, and by sharing it with other users. Other examples of how computer viruses are introduced include situations where authorized users of systems deliberately plant viruses, often with a time bomb mechanism. The virus may then activate itself at some later point in time, perhaps when the user is not logged onto the system or perhaps after the user has left the organization. For more information on computer viruses, see [DENNING88]

## 2.3 Network Worms

Network worm programs use network connections to spread from system to system, thus network worms attack systems that are linked via communications lines. Once active within a system, a network worm can behave as a computer virus, or it could implant Trojan horse programs or perform any number of disruptive or destructive actions. In a sense, network worms are like computer viruses with the ability to infect other systems as well as other programs. Some people use the term virus to include both cases.

To replicate themselves, network worms use some sort of network vehicle, depending on the type of network and systems. Examples of network vehicles include (a) a network mail facility, in which a worm can mail a copy of itself to other systems, or (b), a remote execution capability, in which a worm can execute a copy of itself on another system, or (c) a remote login capability, whereby a worm can log into a remote system as a user and then use commands to copy itself from one system to the other. The new copy of the network worm is then run on the remote system, where it may continue to spread to more systems in a like manner. Depending on the size of a network, a network worm can spread to many systems in a relatively short amount of time, thus the damage it can cause to one system is multiplied by the number of systems to which it can spread.

A network worm exhibits the same characteristics as a computer virus: a *replication* mechanism, possibly an *activation* mechanism, and an *objective*. The replication mechanism generally performs the following functions:

- searches for other systems to infect by examining host tables or similar repositories of remote system addresses

- establishes a connection with a remote system, possibly by logging in as a user or using a mail facility or remote execution capability

- copies itself to the remote system and causes the copy to be run

The network worm may also attempt to determine whether a system has previously been infected before copying itself to the system. In a multi-tasking computer, it may also disguise its presence by naming itself as a system process or using some other name that may not be noticed by a system operator.

The activation mechanism might use a time bomb or logic bomb or any number of variations to activate itself. Its objective, like all malicious software, is whatever the author has designed into it. Some network worms have been designed for a useful purpose, such as to perform general "house-cleaning" on networked systems, or to use extra machine cycles on each networked system to perform large amounts of computations not practical on one system. A network worm with a harmful objective could perform a wide range of destructive functions, such as deleting files on each affected computer, or by implanting Trojan horse programs or computer viruses.

Two examples of actual network worms are presented here. The first involved a Trojan horse program that displayed a Christmas tree and a message of good cheer (this happened during the Christmas season). When a user executed this program, it examined network information files which listed the other personal computers that could receive mail from this user. The program then mailed itself to those systems. Users who received this message were invited to run the Christmas tree program themselves, which they did. The network worm thus continued to spread to other systems until the network was nearly saturated with traffic. The network worm did not cause any destructive action other than disrupting communications and causing a loss in productivity [BUNZEL88].

The second example concerns the incident whereby a network worm used the collection of networks known as the Internet to spread itself to several thousands of computers located throughout the United States. This worm spread itself automatically, employing somewhat sophisticated techniques for bypassing the systems' security mechanisms. The worm's replication mechanism accessed the systems by using one of three methods:

- it employed *password cracking*, in which it attempted to log into systems using usernames for passwords, as well as using words from an on-line dictionary

- it exploited a *trap door* mechanism in mail programs which permitted it to send commands to a remote system's command interpreter

- it exploited a *bug* in a network information program which permitted it to access a remote system's command interpreter

By using a combination of these methods, the network worm was able to copy itself to different brands of computers which used similar versions of a widely-used operating system. Many system managers were unable to detect its presence in their systems, thus it spread very quickly, affecting several thousands of computers within two days. Recovery efforts were hampered because many sites disconnected from the network to prevent further infections, thus preventing those sites from receiving network mail that explained how to correct the problems.

It was unclear what the network worm's objective was, as it did not destroy information, steal passwords, or plant viruses or Trojan horses. The potential for destruction was very high, as the worm could have contained code to effect many forms of damage, such as to destroy all files on each system. For more information, see [DENNING89] and [SPAFFORD88].

## 2.4 Other Related Software Threats

The number of variations of Trojan horses, computer viruses, and network worms is apparently endless. Some have names, such as a *rabbit*, whose objective is to spread wildly within or among other systems and disrupt network traffic, or a *bacterium*, whose objective is to replicate within a system and eat up processor time until computer throughput is halted [DENNING88]. It is likely that many new forms will be created, employing more sophisticated techniques for spreading and causing damage.

## 2.5 The Threat of Unauthorized Use

In that computer viruses and related forms of malicious software are intriguing issues in themselves, it is important not to overlook that they are created by people, and are fundamentally a people problem. In essence, examples of malicious software are tools that people use to extend and enhance their ability to create mischief and various other forms of damage. Such software can do things that the interactive user often cannot directly effect, such as working with great speed, or maintaining anonymity, or doing things that require programmatic system calls. But in general, malicious software exploits the same vulnerabilities as can knowledgeable users. Thus, any steps taken to reduce the likelihood of attack by malicious software should address the likelihood of unauthorized use by computer users.

# 3. Virus Prevention in General

To provide general protection from attacks by computer viruses, unauthorized users, and related threats, users and managers need to eliminate or reduce vulnerabilities. A general summary of the vulnerabilities that computer viruses and related threats are most likely to exploit is as follows:

- lack of user awareness - users copy and share infected software, fail to detect signs of virus activity, do not understand proper security techniques
- absence of or inadequate security controls - personal computers generally lack software and hardware security mechanisms that help to prevent and detect unauthorized use, existing controls on multi-user systems can sometimes be surmounted by knowledgeable users
- ineffective use of existing security controls - using easily guessed passwords, failing to use access controls, granting users more access to resources than necessary
- bugs and loopholes in system software - enabling knowledgeable users to break into systems or exceed their authorized privileges
- unauthorized use - unauthorized users can break in to systems, authorized users can exceed levels of privilege and misuse systems
- susceptibility of networks to misuse - networks can provide anonymous access to systems, many are in general only as secure as the systems which use them

As can be seen from this summary, virus prevention requires that many diverse vulnerabilities be addressed. Some of the vulnerabilities can be improved upon significantly, such as security controls that can be added or improved, while others are somewhat inherent in computing, such as the risk that users will not use security controls or follow policies, or the risk of unauthorized use of computers and networks. Thus, it may not be possible to completely protect systems from all virus-like attacks. However, to attain a realistic degree of protection, all areas of vulnerability must be addressed; improving upon some areas at the expense of others will still leave significant holes in security.

To adequately address all areas of vulnerability, the active involvement of individual users, the management structure, and the organization in a *virus prevention program* is essential. Such a program, whether formal or informal, depends on the mutual cooperation of the three groups to identify vulnerabilities, to take steps to correct them, and to monitor the results.

A virus prevention program must be initially based upon effective system computer administration that restricts access to authorized users, ensures that hardware and software are regularly monitored and maintained, makes backups regularly, and maintains contingency procedures for potential problems. Sites that do not maintain a basic computer administration program need to put one into place, regardless of their size or the types of computers used. Many system vendors supply system administration manuals that describe the aspects of a basic program, and one can consult documents such as [FIPS73], or [NBS120].

Once a basic administration program is in place, management and users need to incorporate virus prevention measures that will help to *deter* attacks by viruses and related threats, *detect* when they occur, *contain* the attacks to limit damage, and *recover* in a reasonable amount of time without loss of data. To accomplish these aims, attention needs to be focused on the following areas:

- *educating users* about malicious software in general, the risks that it poses, how to use control measures, policies, and procedures to protect themselves and the organization
- *software management* policies and procedures that address public-domain software, and the use and maintenance of software in general
- *use of technical controls* that help to prevent and deter attacks by malicious software and unauthorized users
- *monitoring of user and software activity* to detect signs of attacks, to detect policy violations, and to monitor the overall effectiveness of policies, procedures, and controls
- *contingency policies and procedures* for containing and recovering from attacks

General guidance in each of these areas is explained in the following sections.

## 3.1 User Education

Education is one of the primary methods by which systems and organizations can achieve greater protection from incidents of malicious software and unauthorized use. In situations where technical controls do not provide complete protection (i.e., most computers), it is ultimately people and their willingness to adhere to security policies that will determine whether systems and organizations are protected. By educating users about the general nature of computer viruses and related threats, an organization can improve its ability to deter, detect, contain and recover from potential incidents.

Users should be educated about the following:

- how malicious software operates, methods by which it is planted and spread, the vulnerabilities exploited by malicious software and unauthorized users
- general security policies and procedures and how to use them
- the policies to follow regarding the backup, storage, and use of software, especially public-domain software and shareware
- how to use the technical controls they have at their disposal to protect themselves
- how to monitor their systems and software to detect signs of abnormal activity, what to do or whom to contact for more information
- contingency procedures for containing and recovering from potential incidents

User education, while perhaps expensive in terms of time and resources required, is ultimately a cost-effective measure for protecting against incidents of malicious software and unauthorized use. Users who are better acquainted with the destructive potential of malicious software and the methods by which it can attack systems may in turn be prompted to take measures to protect themselves. The purpose of security policies and procedures will be more clear, thus users may be more willing to actively use them. By educating users how to detect abnormal system activity and the resultant steps to follow for containing and recovering from potential incidents, organizations will save money and time if and when actual incidents occur.

## 3.2 Software Management

As shown by examples in Chapter 2, one of the prime methods by which malicious software is initially copied onto systems is by unsuspecting users. When users download programs from sources such as software bulletin boards, or public directories on systems or network servers, or in general use and share software that has not been obtained from a reputable source, users are in danger of spreading malicious software. To prevent users from potentially spreading malicious software, managers need to

- ensure that users understand the nature of malicious software, how it is generally spread, and the technical controls to use to protect themselves
- develop policies for the downloading and use of public-domain and shareware software
- create some mechanism for validating such software prior to allowing users to copy and use it

- minimize the exchange of executable software within an organization as much as possible
- do not create software repositories on LAN servers or in multi-user system directories unless technical controls exist to prevent users from freely uploading or downloading the software

The role of education is important, as users who do not understand the risks yet who are asked to follow necessarily restrictive policies may share and copy software anyway. Where technical controls cannot prevent placing new software onto a system, users are then primarily responsible for the success or failure of whatever policies are developed.

A policy that prohibits any copying or use of public-domain software may be overly restrictive, as some public domain programs have proved to be useful. A less restrictive policy would allow some copying, however a user might first require permission from the appropriate manager. A special system should be used from which to perform the copy and then to test the software. This type of system, called an *isolated system*, should be configured so that there is no risk of spreading a potentially malicious program to other areas of an organization. The system should not be used by other users, should not connect to networks, and should not contain any valuable data. An isolated system should also be used to test internally developed software and updates to vendor software.

Other policies for managing vendor software should be developed. These policies should control how and where software is purchased, and should govern where the software is installed and how it is to be used. The following policies and procedures are suggested:

- purchase vendor software only from reputable sources
- maintain the software properly and update it as necessary
- don't use pirated software, as it may have been modified
- keep records of where software is installed readily available for contingency purposes
- ensure that vendors can be contacted quickly if problems occur
- store the original disks or tapes from the vendor in a secure location

## 3.3 Technical Controls

Technical controls are the mechanisms used to protect the security and integrity of systems and associated data. The use of technical controls can help to prevent occurrences of viruses and related threats by deterring them or making it more difficult for them to gain access to systems and data. Examples of technical controls include user authentication mechanisms such as passwords, mechanisms which provide selective levels of access to files and directories (read-only, no access, access to certain users, etc.), and write-protection mechanisms on tapes and diskettes.

The different types of technical controls and the degree to which they can provide protection and deterrence varies from system to system, thus the use of specific types of controls is discussed in Chapters 4 and 5. However, the following general points are important to note:

- technical controls should be used as available to restrict system access to authorized users only
- in the multi-user environment, technical controls should be used to limit users' privileges to the minimum practical level; they should work automatically and need not be initiated by users
- users and system managers must be educated as to how and when to use technical controls
- where technical controls are weak or non-existent (i.e., personal computers), they should be supplemented with alternative physical controls or add-on control mechanisms

Managers need to determine which technical controls are available on their systems, and then the degree to which they should be used and whether additional add-on controls are necessary. One way to answer these questions is to first categorize the different classes of data being processed by a system or systems, and then to rank the categories according to criteria such as sensitivity to the organization and vulnerability of the system to attack. The rankings should then help determine the degree to which the controls should be applied and whether additional controls are necessary. Ideally, those systems with the most effective controls should be used to process the most sensitive data, and vice-versa. As an example, a personal computer which processes sensitive employee information should require add-on user authentication mechanisms, whereas a personal computer used for general word processing may not need additional controls.

It is important to note that technical controls do not generally provide complete protection against viruses and related threats. They may be cracked by determined users who are knowledgeable of hidden bugs and weaknesses, and they may be surmounted through the use of Trojan horse programs, as shown by examples in Chapter 2. An inherent weakness in technical controls is that,

while deterring users and software from objects to which they do not have access, they may be totally ineffective against attacks which target objects that are accessible. For example, technical controls may not prevent an authorized user from destroying files to which the user has authorized access. Most importantly, when technical controls are not used properly, they may increase a system's degree of vulnerability. It is generally agreed that fully effective technical controls will not be widely available for some time. Because of the immediate nature of the computer virus threat, technical controls must be supplemented by less technically-oriented control measures such as described in this chapter.

## 3.4 General Monitoring

An important aspect of computer viruses and related threats is that they potentially can cause extensive damage within a very small amount of time, such as minutes or seconds. Through proper monitoring of software, system activity, and in some cases user activity, managers can increase their chances that they will detect early signs of malicious software and unauthorized activity. Once the presence is noted or suspected, managers can then use contingency procedures to contain the activity and recover from whatever damage has been caused. An additional benefit of general monitoring is that over time, it can aid in determining the necessary level or degree of security by indicating whether security policies, procedures, and controls are working as planned.

Monitoring is a combination of continual system and system management activity. Its effectiveness depends on cooperation between management and users. The following items are necessary for effective monitoring:

- user education - users must know, specific to their computing environment, what constitutes normal and abnormal system activity and whom to contact for further information - this is especially important for users of personal computers, which generally lack automated methods for monitoring

- automated system monitoring tools - generally on multi-user systems, to automate logging or accounting of user and software accesses to accounts, files, and other system objects - can sometimes be tuned to record only certain types of accesses such as "illegal" accesses

- anti-viral software - generally on personal computers, these tools alert users of certain types of system access that are indicative of "typical" malicious software

- system-sweep programs - programs to automatically check files for changes in size, date, or content

- network monitoring tools - as with system monitoring tools, to record network accesses or attempts to access

The statistics gained from monitoring activities should be used as input for periodic reviews of security programs. The reviews should evaluate the effectiveness of general system management, and associated security policies, procedures, and controls. The statistics will indicate the need for changes and will help to fine tune the program so that security is distributed to where it is most necessary. The reviews should also incorporate users' suggestions, and to ensure that the program is not overly restrictive, their criticisms.

## 3.5 Contingency Planning

The purpose of contingency planning with regard to computer viruses and related threats is to be able to contain and recover completely from actual attacks. In many ways, effective system management that includes user education, use of technical controls, software management, and monitoring activities, is a form of contingency planning, generally because a well-run, organized system or facility is better able to withstand the disruption that could result from a computer virus attack. In addition to effective system management activities, managers need to consider other contingency procedures that specifically take into account the nature of computer viruses and related threats.

Possibly the most important contingency planning activity involves the use of backups. The ability to recover from a virus attack depends upon maintaining regular, frequent backups of all system data. Each backup should be checked to ensure that the backup media has not been corrupted. Backup media could easily be corrupted because of defects, because the backup procedure was incorrect, or perhaps because the backup software itself has been attacked and modified to corrupt backups as they are made.

Contingency procedures for restoring from backups after a virus attack are equally important. Backups may contain copies of malicious software that have been hiding in the system. Restoring the malicious software to a system that has been attacked could cause a recurrence of the problem. To avoid this possibility, software should be restored only from its original media: the tapes or diskettes from the vendor. In some cases, this may involve reconfiguring the software, therefore managers must maintain copies of configuration information for system and application software. Because data is not directly executable, it can be restored from routine backups. However, data that has been damaged may need to be restored manually or from older backups. Command files such as batch procedures and files executed when systems boot or when user log on should be inspected to ensure that they have not been damaged or modified. Thus, managers will need to

retain successive versions of backups, and search through them when restoring damaged data and command files.

Other contingency procedures for containing virus attacks need to be developed. The following are suggested; they are discussed in more detail in Chapters 4 and 5:

- ensure that accurate records are kept of each system's configuration, including the system's location, the software it runs, the system's network and modem connections, and the name of the system's manager or responsible individual
- create a group of skilled users to deal with virus incidents and ensure that users can quickly contact this group if they suspect signs of viral activity
- maintain a security distribution list at each site with appropriate telephone numbers of managers to contact when problems occur
- isolate critical systems from networks and other sources of infection
- place outside network connections on systems with the best protections, use central gateways to facilitate rapid disconnects

# 4. Virus Prevention for Multi-User Computers and Associated Networks

Virus prevention in the multi-user computer environment is aided by the centralized system and user management, and the relative richness of technical controls. Unlike personal computers, many multi-user systems possess basic controls for user authentication, for levels of access to files and directories, and for protected regions of memory. By themselves, these controls are not adequate, but combined with other policies and procedures that specifically target viruses and related threats, multi-user systems can greatly reduce their vulnerabilities to exploitation and attack.

However, some relatively powerful multi-user machines are now so compact as to be able to be located in an office or on a desk-top. These machines are still fully able to support a small user population, to connect to major networks, and to perform complex real-time operations. But due to their size and increased ease of operation, they are more vulnerable to unauthorized access. Also, multi-user machines are sometimes managed by untrained personnel who do not have adequate time to devote to proper system management and who may not possess a technical background or understanding of the system's operation. Thus, it is especially important for organizations who use or are considering machines of this nature to pay particular attention to the risks of attack by unauthorized users, viruses, and related software.

The following sections offer guidance and recommendations for improving the management and reducing the risk of attack for multi-user computers and associated networks.

## 4.1 General Policies

Two general policies are suggested here. They are intended for uniform adoption throughout an organization, i.e., they will not be entirely effective if they are not uniformly followed. These policies are as follows:

- An organization must assign a dedicated system manager to operate each multi-user computer. The manager should be trained, if necessary, to operate the system in a practical and secure manner. This individual should be assigned the management duties as part of his job description; the management duties should not be assigned "on top" of the individual's other duties, but rather adequate time should be taken from other duties. System management is a demanding and time-consuming operation that can unexpectedly require complete dedication. As systems are increasingly inter-connected via networks, a poorly managed system that can be used as a pathway for unauthorized

access to other systems will present a significant vulnerability to an organization. Thus, the job of system manager should be assigned carefully, and adequate time be given so that the job can be performed completely.

- Management needs to impress upon users the need for their involvement and cooperation in computer security. A method for doing this is to create an organizational security policy. This policy should be a superset of all other computer-related policy, and should serve to clearly define what is expected of the user. It should detail how systems are to be used and what sorts of computing are permitted and not permitted. Users should read this policy and agree to it as a prerequisite to computer use. It would also be helpful to use this policy to create other policies specific to each multi-user system.

## 4.2 Software Management

Effective software management can help to make a system less vulnerable to attack and can make containment and recovery more successful. Carefully controlled access to software will prevent or discourage unauthorized access. If accurate records and backups are maintained, software restoral can be accomplished with a minimum of lost time and data. A policy of testing all new software, especially public-domain software, will help prevent accidental infection of a system by viruses and related software. Thus, the following policies and procedures are recommended:

- Use only licensed copies of vendor software, or software that can be verified to be free of harmful code or other destructive aspects. Maintain complete information about the software, such as the vendor address and telephone number, the license number and version, and update information. Store the software in a secure, tamper-proof location.

- Maintain configuration reports of all installed software, including the operating system. This information will be necessary if the software must be re-installed later.

- Prevent user access to system software and data. Ensure that such software is fully protected, and that appropriate monitoring is done to detect attempts at unauthorized access.

- Prohibit users from installing software. Users should first contact the system manager regarding new software. The software should then be tested on an *isolated* system to determine whether the software may contain destructive elements. The isolated system should be set up so that, to a practical degree, it replicates the target system, but does not connect to networks or process sensitive data. A highly-skilled user knowledgeable about viruses and related threats should perform the testing and ensure that the software does not change or delete other software or data. Do not allow users to directly add any software to the system, whether from public software repositories, or other systems, or their home systems.

- Teach users to protect their data from unauthorized access. Ensure that they know how to use access controls or file protection mechanisms to prevent others from reading or modifying their files. As possible, set default file protections such that when a user creates a file, the file can be accessed only by that user, and no others. Each user should not permit others to use his or her account.

- Do not set-up directories to serve as software repositories unless technical controls are used to prevent users from writing to the directory. Make sure that users contact the system manager regarding software they wish to place in a software repository. It would be helpful to track where the software is installed by setting up a process whereby users must first register their names before they can copy software from the directory.

- If developing software, control the update process so that the software is not modified without authorization. Use a software management and control application to control access to the software and to automate the logging of modifications.

- Accept system and application bug fixes or patches only from highly reliable sources, such as the software vendor. Do not accept patches from anonymous sources, such as received via a network. Test the new software on an isolated system to ensure that the software does not make an existing problem worse.

## 4.3 Technical Controls

Many multi-user computers contain basic built-in technical controls. These include user authentication via passwords, levels of user privilege, and file access controls. By using these basic controls effectively, managers can significantly reduce the risk of attack by preventing or deterring viruses and related threats from accessing a system.

Perhaps the most important technical control is user authentication, with the most widely form of user authentication being a username associated with a password. Every user account should use a password that is deliberately chosen so that simple attempts at password cracking cannot occur. An effective password should not consist of a person's name or a recognizable word, but rather should consist of alphanumeric characters and/or strings of words that cannot easily be guessed. The passwords should be changed at regular intervals, such as every three to six months. Some systems include or can be modified to include a password history, to prevent users from reusing old passwords. For more information on effective password practices, see [FIPS73].

The username/password mechanism can sometimes be modified to reduce opportunities for password cracking. One method is to increase the running time of the password encryption to several

seconds. Another method is to cause the user login program to accept from three to five incorrect password attempts in a row before disabling the user account for several minutes. Both methods significantly increase the amount of time a password cracker would spend when making repeated attempts at guessing a password. A method for ensuring that passwords are difficult to crack involves the use of a program that could systematically guess passwords, and then send warning messages to the system manager and corresponding users if successful. The program could attempt passwords that are permutations of each user's name, as well as using words from an on-line dictionary.

Besides user authentication, access control mechanisms are perhaps the next most important technical control. Access control mechanisms permit a system manager to selectively permit or bar user access to system resources regardless of the user's level of privilege. For example, a user at a low-level of system privilege can be granted access to a resource at a higher level of privilege without raising the user's privilege through the use of an access control that specifically grants that user access. Usually, the access control can determine the type of access, e.g., read or write. Some access controls can send alarm messages to audit logs or the system manager when unsuccessful attempts are made to access resources protected by an access control.

Systems which do not use access controls usually contain another more basic form that grants access based on user categories. Usually, there are four: *owner*, where only the user who "owns" or creates the resource can access it; *group*, where anyone in the same group as the owner can access the resource; *world*, where all users can access the resource, and *system*, which supersedes all other user privileges. Usually, a file or directory can be set up to allow any combination of the four. Unlike access controls, this scheme doesn't permit access to resources on a specific user basis, thus if a user at a low level of privilege requires access to a system level resource, the user must be granted system privilege. However, if used carefully, this scheme can adequately protect users' files from being accessed without authorization. The most effective mode is to create a unique group for each user. Some systems may permit a default file permission mask to be set so that every file created would be accessible only by the file's owner.

Other technical control guidelines are as follows:

- Do not use the same password on several systems. Additionally, sets of computers that are mutually trusting in the sense that login to one constitutes login to all should be carefully controlled.

- Disable or remove old or unnecessary user accounts. Whenever users leave an organization or no longer use a system, change all passwords that the users had knowledge of.

- Practice a "least privilege" policy, whereby users are restricted to accessing resources on a need-to-know basis only. User privileges should be as restricting as possible without adversely affecting the performance of their work. To determine what level of access is required, err first by setting privileges to their most restrictive, and upgrade them as necessary. If the system uses access controls, attempt to maintain a user's system privileges at a low level while using the access controls to specifically grant access to the required resources.

- Users are generally able to determine other users' access to their files and directories, thus instruct users to carefully maintain their files and directories such that they are not accessible, or at a minimum, not writable, by other users. As possible, set default file protections such that files and directories created by each user are accessible by only that user.

- When using modems, do not provide more access to the system than is necessary. For example, if only dial-out service is required, set up the modem or telephone line so that dial-in service is not possible. If dial-in service is necessary, use modems that require an additional passwords or modems that use a call-back mechanism. These modems may work such that a caller must first identify himself to the system. If the identification has been pre-recorded with the system and therefore valid, the system then calls back at a pre-recorded telephone number.

- If file encryption mechanisms are available, make them accessible to users. Users may wish to use encryption as a further means of protecting the confidentiality of their files, especially if the system is accessible via networks or modems.

- Include software so that users can temporarily "lock" their terminals from accepting keystrokes while they are away. Use software that automatically disables a user's account if no activity occurs after a certain interval, such as 10 - 15 minutes.

## 4.4 Monitoring

Many multi-user systems provide a mechanism for automatically recording some aspects of user and system activity. This monitoring mechanism, if used regularly, can help to detect evidence of viruses and related threats. Early detection is of great value, because malicious software potentially can cause significant damage within a matter of minutes. Once evidence of an attack has been verified, managers can use contingency procedures to contain and recover from any resultant damage.

Effective monitoring also requires user involvement, and therefore, user education. Users must have some guidelines for what constitutes normal and abnormal system activity. They need to be aware of such items as whether files have been changed in content, date, or by access permissions,

whether disk space has become suddenly full, and whether abnormal error messages occur. They need to know whom to contact to report signs of trouble and then the steps to take to contain any damage.

The following policies and procedures for effective monitoring are recommended:

- Use the system monitoring/auditing tools that are available. Follow the procedures recommended by the system vendor, or start out by enabling the full level or most detailed level of monitoring. Use tools as available to help read the logs, and determine what level of monitoring is adequate, and cut back on the level of detail as necessary. Be on the guard for excessive attempts to access accounts or other resources that are protected. Examine the log regularly, at least weekly if not more often.

- As a further aid to monitoring, use alarm mechanisms found in some access controls. These mechanisms send a message to the audit log whenever an attempt is made to access a resource protected by an access control.

- If no system monitoring is available, or if the present mechanism is unwieldy or not sufficient, investigate and purchase other monitoring tools as available. Some third-party software companies sell monitoring tools for major operating systems with capabilities that supersede those of the vendor's.

- Educate users so that they understand the normal operating aspects of the system. Ensure that they have quick access to an individual or group who can answer their questions and investigate potential virus incidents.

- Purchase or build system sweep programs to checksum files at night, and report differences from previous runs. Use a password checker to monitor whether passwords are being used effectively.

- Always report, log, and investigate security problems, even when the problems appear insignificant. Use the log as input into regular security reviews. Use the reviews as a means for evaluating the effectiveness of security policies and procedures.

- Enforce some form of sanctions against users who *consistently* violate or attempt to violate security policies and procedures. Use the audit logs as evidence, and bar the users from system use.

## 4.5 Contingency Planning

As stressed in Chapter 3, backups are the most important contingency planning activity. A system manager must plan for the eventuality of having to restore all software and data from backup tapes for any number of reasons, such as disk drive failure or upgrades. It has been shown that viruses and related threats could potentially and unexpectedly destroy all system information or render it useless, thus managers should pay particular attention to the effectiveness of their backup policies. Backup policies will vary from system to system, however they should be performed daily, with a minimum of several months backup history. Backup tapes should be verified to be accurate, and should be stored *off-site* in a secured location.

Viruses and related software threats could go undetected in a system for months to years, and thus could be backed up along with normal system data. If such a program would suddenly trigger and cause damage, it may require much searching through old backups to determine when the program first appeared or was infected. Therefore the safest policy is to restore programs, i.e., executable and command files, from their original vendor media only. Only system data that is non-executable should be restored from regular backups. Of course, in the case of command files or batch procedures that are developed or modified in the course of daily system activity, these may need to be inspected manually to ensure that they have not been modified or damaged.

Other recommended contingency planning activities are as follows:

- Create a security distribution list for hand-out to each user. The list should include the system manager's name and number, and other similar information for individuals who can answer users' questions about suspicious or unusual system activity. The list should indicate when to contact these individuals, and where to reach them in emergencies.

- Coordinate with other system managers, especially if their computers are connected to the same network. Ensure that all can be contacted quickly in the event of a network emergency by using some mechanism other than the network.

- Besides observing physical security for the system as well as its software and backup media, locate terminals in offices that can be locked or in other secure areas.

- If users are accessing the system via personal computers and terminal emulation software, keep a record of where the personal computers are located and their network or port address for monitoring purposes. Control carefully whether such users are uploading software to the system.

- Exercise caution when accepting system patches. Do not accept patches that arrive over a network unless there is a high degree of certainty as to their validity. It is best to accept patches only from the appropriate software vendor.

## 4.6 Associated Network Concerns

Multi-user computers are more often associated with relatively large networks than very localized local area networks or personal computer networks that may use dedicated network servers. The viewpoint taken here is that wide area network and large local area network security is essentially a collective function of the systems connected to the network, i.e., it is not practical for a controlling system to monitor *all* network traffic and differentiate between authorized and unauthorized use. A system manager should generally assume that network connections pose inherent risks of unauthorized access to the system in the forms of unauthorized users and malicious software. Thus, a system manager needs to protect the system from network-borne threats and likewise exercise responsibility by ensuring that his system is not a source of such threats, while at the same time making network connections available to users as necessary. The accomplishment of these aims will require the use of technical controls to restrict certain types of access, monitoring to detect violations, and a certain amount of trust that users will use the controls and follow the policies.

Some guidelines for using networks in a more secure manner are as follows:

- Assume that network connections elevate the risk of unauthorized access. Place network connections on system which provide adequate controls, such as strong user authentication and access control mechanisms. Avoid placing network connections on system which process sensitive data.

- If the system permits, require an additional password or form of authentication for accounts accessed from network ports. If possible, do not permit access to system manager accounts from network ports.

- If anonymous or guest accounts are used, place restrictions on the types of commands that can be executed from the account. Don't permit access to software tools, commands that can increase privileges, and so forth.

- As possible, monitor usage of the network. Check if network connections are made at odd hours, such as during the night, or if repeated attempts are made to log in to the system from a network port.

- When more than one computer is connected to the same network, arrange the connections so that one machine serves as a central gateway for the other machines. This will allow a rapid disconnect from the network in case of an attack.

- Ensure that users are fully educated in network usage. Make them aware of the additional risks involved in network access. Instruct them to be on the alert for any signs of tampering, and to contact an appropriate person if they detect any suspicious activity. Create a policy for responsible network usage that details what sort of computing activity will and will not be tolerated. Have users read the policy as a prerequisite to network use.

- Warn users to be suspicious of any messages that are received from unidentified or unknown sources.

- Don't advertise a system to network users by printing more information than necessary on a welcome banner. For example, don't include messages such as "Welcome to the Payroll Accounting System" that may cause the system to be more attractive to unauthorized users.

- Don't network to outside organizations without a mutual review of security practices

# 5. Virus Prevention for Personal Computers and Associated Networks

Virus prevention in the personal computer environment differs from that of the multi-user computer environment mainly in the following two respects: the relative lack of technical controls, and the resultant emphasis this places on less-technically oriented means of protection which necessitates more reliance on user involvement. Personal computers typically do not provide technical controls for such things as user authorization, access controls, or memory protection that differentiates between system memory and memory used by user applications. Because of the lack of controls and the resultant freedom with which users can share and modify software, personal computers are more prone to attack by viruses, unauthorized users, and related threats.

Virus prevention in the personal computer environment must rely on continual user awareness to adequately detect potential threats and then to contain and recover from the damage. Personal computer users are in essence personal computer managers, and must practice their management as a part of their general computing. Personal computers generally do not contain auditing features, thus a user needs to be aware at all times of the computer's performance, i.e., what it is doing, or what is normal or abnormal activity. Ultimately, personal computer users need to understand some of the technical aspects of their computers in order to protect, deter, contain, and recover. Not all personal computer users are technically oriented, thus this poses some problems and places even more emphasis on user education and involvement in virus prevention.

Because of the dependance on user involvement, policies for the personal computer environment are more difficult to implement than in the multi-user computer environment. However, emphasizing these policies as part of a user education program will help to ingrain them in users' behavior. Users should be shown via examples what can happen if they don't follow the policies. An example where users share infected software and then spread the software throughout an organization would serve to effectively illustrate the point, thus making the purpose of the policy more clear and more likely to be followed. Another effective method for increasing user cooperation is to create a list of effective personal computer management practices specific to each personal computing environment. Creating such a list would save users the problem of determining how best to enact the policies, and would serve as a convenient checklist that users could reference as necessary.

It will likely be years before personal computers incorporate strong technical controls in their architectures. In the meantime, managers and users must be actively involved in protecting their

computers from viruses and related threats. The following sections provide guidance to help achieve that aim.

## 5.1 General Policies

Two general policies are suggested here. The first requires that management make firm, unambiguous decisions as to how users should operate personal computers, and state that policy in writing. This policy will be a general re-statement of all other policies affecting personal computer use. It is important that users read this policy and agree to its conditions as a prerequisite to personal computer use. The purposes of the policy are to (1) ensure that users are aware of all policies, and (2) impress upon users the need for their active involvement in computer security.

The second policy is that every personal computer should have an "owner" or "system manager" who is responsible for the maintenance and security of the computer, and for following all policies and procedures associated with the use of the computer. It would be preferable that the primary user of the computer fill this role. It would not be too extreme to make this responsibility a part of the user's job description. This policy will require that resources be spent on educating users so that they can adequately follow all policies and procedures.

## 5.2 Software Management

Due to the wide variety of software available for many types of personal computers, it is especially important that software be carefully controlled. The following policies are suggested:

- Use only licensed copies of vendor software for personal computers. Ensure that the license numbers are logged, that warranty information is completed, and that updates or update notices will be mailed to the appropriate users. Ensure that software versions are uniform on all personal computers. Purchase software from known, reputable sources - do not purchase software that is priced suspiciously low and do not use pirated software, even on a trial basis. As possible, buy software with built-in security features.

- Do not install software that is not clearly needed. For example, software tools such as compilers or debuggers should not be installed on machines where they are not needed.

- Store the original copies of vendor software in a secure location for use when restoring the software.

- Develop a clear policy for use of public-domain software and shareware. It is recommended that the policy prohibit indiscriminate downloading from software bulletin boards. A special *isolated* system should be configured to perform the downloading, as well as for testing downloaded and other software or shareware. The operation of the system should be managed by a technically skilled user who can use anti-virus software and other techniques to test new software before it is released for use by other users.

- Maintain an easily-updated database of installed software. For each type of software, the database should list the computers where the software is installed, the license numbers, software version number, the vendor contact information, and the responsible person for each computer listed. This database should be used to quickly identify users, machines, and software when problems or emergencies arise, such as when a particular type of software is discovered to contain a virus or other harmful aspects.

- Minimize software sharing within the organization. Do not permit software to be placed on computers unless the proper manager is notified and the software database is updated. If computer networks permit software to be mailed or otherwise transferred among machines, prohibit this as a policy. Instruct users not to run software that has been mailed to them.

- If using software repositories on LAN servers, set up the server directory such that users can copy from the directory, but not add software to the directory. Assign a user to manage the repository; all updates to the repository should be cleared through this individual. The software should be tested on an isolated system as described earlier.

- If developing software, consider the use of software management and control programs that automate record keeping for software updates, and that provide a degree of protection against unauthorized modifications to the software under development.

- Prohibit users from using software or disks from their home systems. A home system that is used to access software bulletin boards or that uses shared copies of software could be infected with viruses or other malicious software.

## 5.3 Technical Controls

As stated earlier, personal computers suffer from a relative lack of technical controls. There are usually no mechanisms for user authentication and for preventing users or software from modifying system and application software. Generally, all software and hardware is accessible by the personal computer user, thus the potential for misuse is substantially greater than in the multi-user computer environment.

However, some technical controls can be added to personal computers, e.g., user authentication devices. The technical controls that do not exist can be simulated by other controls, such as a lock

on an office door to substitute for a user authentication device, or anti-virus software to take the place of system auditing software. Lastly, some of the personal computer's accessibility can be reduced, such as by the removal of floppy diskette drives or by the use of diskless computers that must download their software from a LAN server. The following items are suggested:

- Where technical controls exist, use them. If basic file access controls are available to make files read-only, make sure that operating system files and other executable files are marked as read-only. Use write-protect tabs on floppy diskettes and tapes. If LAN access requires a password, ensure that passwords are used carefully - follow the guidelines for password usage presented in Chapter 4 or see [FIPS73].

- Use new cost-effective forms of user identification such as magnetic access cards. Or, setup other software such as password mechanism that at a minimum deters unauthorized users.

- If using a LAN, consider downloading the personal computer's operating system and other applications from a read-only directory on the LAN server (instead of the personal computer's hard disk). If the LAN server is well protected, this arrangement would significantly reduce chances of the software becoming infected, and would simplify software management.

- Consider booting personal computers from write-protected floppy diskettes (instead of the computer's hard disk). Use a unique diskette per computer, and keep the diskette secured when not in use.

- Do not leave a personal computer running but unattended. Lock the computer with a hardware lock (if possible), or purchase vendor add-on software to "lock" the keyboard using a password mechanism. Alternatively, turn off the computer and lock the office door. Shut down and lock the computer at the end of the day.

- When using modems connected to personal computers, do not provide more access to the computer than necessary. If only dial-out service is required, configure the modem so that it won't answer calls. If dial-in service is necessary, consider purchasing modems that require a password or that use a call-back mechanism to force a caller to call from a telephone number that is known to the modem.

- Consider using "limited-use" systems, whereby the capabilities of a system are restricted to only what is absolutely required. For example, users who run only a certain application (such as word-processor) may not require the flexibility of a personal computer. At the minimum, do not install applications or network connections where they are not needed.

## 5.4 Monitoring

Personal computer operating systems typically do not provide any software or user monitoring/auditing features. Monitoring, then, is largely a user function whereby the user must be aware of what the computer is doing, such as when the computer is accessing the disk or the general speed of its response to commands, and then must decide whether the activity is normal or abnormal. Anti-viral software can be added to the operating system and run in such a way that the software flags or in some way alerts a user when suspicious activity occurs, such as when critical files or memory regions are written.

Effective monitoring depends on user education. Users must know what constitutes normal and abnormal activity on their personal computers. They need to have a reporting structure available so that they can alert an informed individual to determine whether there is indeed a problem. They need to know the steps to take to contain the damage, and how to recover. Thus, the following policies and procedures are recommended:

- Form a team of skilled technical people to investigate problems reported by users. This same group could be responsible for other aspects of virus prevention, such as testing new software and handling the containment and recovery from virus-related incidents. Ensure that users have quick access to this group, e.g., via a telephone number.

- Educate users so that they are familiar with how their computers function. Show them how to use such items as anti-viral software. Acquaint them with how their computers boot, what files are loaded, whether start-up batch files are executed, and so forth.

- Users need to watch for changes in patterns of system activity. They need to watch for program loads that suddenly take longer, whether disk accesses seem excessive for simple tasks, do unusual error messages occur, do access lights for disks turn on when no disk activity should occur, is less memory available than usual, do files disappear mysteriously, is there less disk space than normal?

- Users also need to examine whether important files have changed in size, date, or content. Such files would include the operating system, regularly-run applications, and other batch files. System sweep programs may be purchased or built to perform checksums on selected files, and then to report whether changes have occurred since the last time the program was run.

- Purchase virus prevention software as applicable. At a minimum, use anti-viral software to test new software before releasing it to other users. However, do not download or use pirated copies of anti-viral software.

- Always report, log, and investigate security problems, even when the problems appear insignificant. Then use the log as input into regular security reviews. Use the reviews as a means for evaluating the effectiveness of security policies and procedures.

## 5.5 Contingency Planning

As described in Chapter 3, backups are the single most important contingency procedure. It is especially important to emphasize regular backups for personal computers, due to their greater susceptibility to misuse and due to the usual requirement of direct user involvement in the backup procedure, unlike that of multi-user computers. Because of the second factor, where users must directly copy files to one or more floppy diskettes, personal computer backups are sometimes ignored or not done completely. To help ensure that backups are done regularly, external backup mechanisms that use a high-density tape cartridge can be purchased and a user assigned to run the backup procedure on a regular basis. Additionally, some personal computer networks contain a personal computer backup feature, where a computer can directly access a network server's backup mechanism, sometimes in an off-line mode at a selected time. If neither of these mechanisms are available, then users must be supplied with an adequate number of diskettes to make complete backups and to maintain a reasonable amount of backup history, with a minimum of several weeks.

Users should maintain the original installation media for software applications and store it in a secure area, such as a locked cabinet, container, or desk. If a user needs to restore software, the user should use only the original media; the user should not use any other type of backup or a copy belonging to another user, as they could be infected or damaged by some form of malicious software.

The effectiveness of a backup policy can be judged by whether a user is able to recover with a minimum loss of data from a situation whereby the user would have to format the computer's disk and reload all software. Several incidents of malicious software have required that users go to this length to recover - see [MACAFEE89].

Other important contingency procedures are described below:

- Maintain a database of personal computer information. Each record should include items such as the computer's configuration, i.e., network connections, disks, modems, etc., the computer's location, how it is used, the software it runs, and the name of the computer's primary user/manager. Maintain this database to facilitate rapid communication and identification when security problems arise.

- Create a security distribution list for each user. The list should include names of people to contact who can help identify the cause of unusual computer activity, and other appropriate security personnel to contact when actual problems arise.

- Create a group of skilled users who can respond to users' inquiries regarding virus detection. This group should be able to determine when a computer has been attacked, and how best to contain and recover from the problem.

- Set up some means of distributing information rapidly to all affected users in the event of an emergency. This should not rely upon a computer network, as the network could actually be attacked, but could use other means such as telephone mail or a general announcement mechanism.

- Observe physical security for personal computers. Locate them in offices that can be locked. Do not store software and backups in unsecured cabinets.

## 5.6 Associated Network Concerns

Personal computer networks offer many advantages to users, however they must be managed carefully so that they do not increase vulnerability to viruses and related threats. Used incorrectly, they can become an additional pathway to unauthorized access to systems, and can be used to plant malicious software such as network worms. This section does not provide specific management guidance, as there are many different types of personal computer networks with widely varying degrees of similarity. However, some general suggestions for improving basic management are listed below:

- Assign a network administrator, and make the required duties part of the administrator's job description. Personal computer networks are becoming increasingly complex to administer, thus the administration should not be left to an individual who cannot dedicate time as necessary.

- Protect the network server(s) by locating them in secure areas. Make sure that physical access is restricted during off-hours. If possible, lock or remove a server's keyboard to prevent tampering.

- Do not provide for more than one administrator account, i.e., do not give other users administrator privileges. Similar to the problem of multiple system manager accounts on multi-user systems, this situation makes it more likely that a password will become known, and makes overall management more difficult to control. Users should coordinate their requests through a single network administrator.

- Do not permit users to connect personal computers to the network cable without permission. The administrator should keep an updated diagram of the network's topology, complete with corresponding network addresses and users.

- Use the network monitoring tools that are available. Track network usage and access to resources, and pinpoint unauthorized access attempts. Take appropriate action when violations consistently occur, such as requiring the user in question to attend a network user class or disabling the user's network account.

- Ensure that users know how to properly use the network. Show them how to use all security features. Ensure that users know how to use passwords and access controls effectively - see [FIPS73] for information on password usage. Show them the difference between normal and abnormal network activity or response. Encourage users to contact the administrator if they detect unusual activity. Log and investigate all problems.

- Do not give users more access to network resources than they require. If using shared directories, make them read-only if write permission is not required, or use a password. Encourage users to do the same with their shared directories.

- Do not set up directories for software repository unless (1) someone can first verify whether the software is not infected, and (2) users are not permitted to write to the directory without prior approval.

- Backup the network server(s) regularly. If possible or practical, backup personal computers using the network server backup mechanism.

- Disable the network mail facility from transferring executable files, if possible. This will prevent software from being indiscriminately shared, and may prevent network worm programs from accessing personal computers.

- For network guest or anonymous accounts, limit the types of commands that can be executed.

- Warn network users to be suspicious of any messages or programs that are received from unidentified sources - network users should have a critical and suspicious attitude towards anything received from an unknown source.

- Always remove old accounts or change passwords. Change important passwords immediately when users leave the organization or no longer require access to the network.

# References

BUNZEL88 Bunzel, Rick; Flu Season; Connect, Summer 1988.

DENNING88 Denning, Peter J.; Computer Viruses; American Scientist, Vol 76, May-June, 1988.

DENNING89 Denning, Peter J.; The Internet Worm; American Scientist, Vol 77, March-April, 1989.

FIPS73 Federal Information Processing Standards Publication 73, Guidelines for Security of Computer Applications; National Bureau of Standards, June, 1980.

FIPS112 Federal Information Processing Standards Publication 112, Password Usage; National Bureau of Standards, May, 1985.

MACAFEE89 McAfee, John; The Virus Cure; Datamation, Feb 15, 1989.

NBS120 NBS Special Publication 500-120; Security of Personal Computer Systems: A Management Guide; National Bureau of Standards, Jan 1985.

SPAFFORD88 Spafford, Eugene H.; The Internet Worm Program: An Analysis; Purdue Technical Report CSD-TR-823, Nov 28, 1988.

THOMPSON84 Thompson, Ken; Reflections on Trusting Trust (Deliberate Software Bugs); Communications of the ACM, Vol 27, Aug 1984.

# Suggested Reading

In addition to the references listed in Appendix A, the following documents are suggested reading for specific and general information on computer viruses and related forms, and other related security information.

Brenner, Aaron; LAN Security; LAN Magazine, Aug 1989.

Cohen, Fred; Computer Viruses, Theory and Experiments; 7th Security Conference, DOD/NBS Sept 1984.

Computer Viruses - Proceedings of an Invitational Symposium, Oct 10/11, 1988;
Deloitte, Haskins, and Sells; 1989

Dvorak, John; Virus Wars: A Serious Warning; PC Magazine; Feb 29, 1988.

Federal Information Processing Standards Publication 83, Guideline on User Authentication Techniques for Computer Network Access Control; National Bureau of Standards, Sept, 1980.

Federal Information Processing Standards Publication 87, Guidelines for ADP Contingency Planning; National Bureau of Standards, March, 1981.

Fiedler, David and Hunter, Bruce M.; Unix System Administration; Hayden Books, 1987

Fitzgerald, Jerry; Business Data Communications: Basic Concepts, Security, and Design; John Wiley and Sons, Inc., 1984

Gasser, Morrie; Building a Secure Computer System; Van Nostrand Reinhold, New York, 1988.

Grampp, F. T. and Morris, R. H.; UNIX Operating System Security; AT&T Bell Laboratories Technical Journal, Oct 1984.

Highland, Harold J.; From the Editor -- Computer Viruses; Computers & Security; Aug 1987.

Longley, Dennis and Shain, Michael; Data and Computer Security

NBS Special Publication 500-120; Security of Personal Computer Systems: A Management Guide; National Bureau of Standards, Jan 1985.

Parker, T.; Public domain software review: Trojans revisited, CROBOTS, and ATC; Computer Language; April 1987.

Schnaidt, Patricia; Fasten Your Safety Belt; LAN Magazine, Oct 1987.

Shoch, J. F. and Hupp, J. A.; The Worm Programs: Early Experience with a Distributed Computation; Comm of ACM, Mar 1982.

White, Stephen and Chess, David; Coping with Computer Viruses and Related Problems; IBM Research Report RC 14405 (#64367), Jan 1989.

Witten, I. H.; Computer (In)security: infiltrating open systems; Abacus (USA) Summer 1987.