10
4-13-92 OSD

MARTIN MARIETTA

ORNL/CSD/TM-281

# An Introduction to Computer Viruses

David R. Brown



MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED S..TES
DEPARTMENT OF ENERGY

ORNL/CSD/TM--281

DE92 010727

# An Introduction to Computer Viruses

David R. Brown
Computing and Telecommunications Division
Oak Ridge National Laboratory
P. O. Box 2008
Oak Ridge, TN 37831

Date Published - March 1992

Prepared by
Martin Marietta Energy Systems, Inc.
managing the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge K-25 Site Paducah Gaseous Diffusion Plant
for the
U.S. Department of Energy
under contract DE-AC05-84OR21400

## Contents

## List of Figures

## Abstract

This report on computer viruses is based upon a thesis written for the Master of Science degree in Computer Science from the University of Tennessee in December 1989 by David R. Brown. This thesis is entitled *An Analysis of Computer Virus Construction, Proliferation, and Control* and is available through the University of Tennessee Library.

This paper contains an overview of the computer virus arena that can help the reader to evaluate the threat that computer viruses pose. The extent of this threat can only be determined by evaluating many different factors. These factors include the relative ease with which a computer virus can be written, the motivation involved in writing a computer virus, the damage and overhead incurred by infected systems, and the legal implications of computer viruses, among others.

Based upon the research, the development of a computer virus seems to require more persistance than technical expertise. This is a frightening proclamation to the computing community. The education of computing professionals to the dangers that viruses pose to the welfare of the computing industry as a whole is stressed as a means of inhibiting the current proliferation of computer virus programs.

Recommendations are made to assist computer users in preventing infection by computer viruses. These recommendations support solid general computer security practices as a means of combating computer viruses.

## Introduction

Computer viruses and other destructive computer programs have recently made national headlines. On November 2, 1988, a self-replicating computer program infected approximately 6000 computers across the United States and served to focus national attention to the problem.[1] Computer viruses have also been encountered on most major microcomputer hardware platforms. Countless hours have been lost in isolating and eradicating these viruses. Some of these viruses corrupt data or disrupt the normal operation of the infected computer system.

To understand the operation of a computer virus requires a fairly detailed knowledge of the inner workings of computers. The general public, and even some within the computing profession, do not currently possess this understanding. Journalists outside of the computing industry have a tendency to emphasize the sensational when reporting on a computer virus infection. For these reasons, common misconceptions of the nature of computer viruses are allowed to flourish.[2]

On the other hand, professionals in the computer industry usually possess a much greater understanding of computers. Computers are a part of their day-to-day life. Like the old saying, "Familiarity breeds contempt," these users may be more reluctant to accept the fact that computer viruses pose a real threat to security. This feeling has caused something of a backlash of apathy from computing professionals.[3]

Computer viruses can pose a real threat to security. This point is driven home by the following news item. A computer virus discovered by Jack E. Juni, M.D., and Richard Ponto had installed itself into a self-contained unit used in the nuclear medicine department at three hospitals. Although there were no reported deaths, some patients were needlessly exposed to additional radiation caused by the corruption of patient records by the virus.[4]

Admittedly this is an extreme case, but it points out rather dramatically that computer viruses are not a subject to be taken lightly. The damage they can cause is real. The subject is an important one in our increasingly information-based society where we are dependent upon the integrity of the information we receive. Financial decisions, and sometimes even life-and-death decisions, are based on this information.

According to William H. Murray of Ernst & Whitney, "If you can conceive it, and if it could be done by any other program, then it can be done by a virus." [5] This is a frightening proclamation for the computing industry.

Steve Gibson, president of Gibson Research Corporation and a contributing editor to *Info World*, is quoted as saying,

> Viruses are created by people with nothing better to do. They have trigger dates far into the future. The authors want to see what happens on that date; they want to read about the effect in the paper. Viruses are almost impossible to get out of your system and they can spread insidiously.
>
> In the not-too-distant future, I am sure there's going to be a major-scale corporate information system disaster that will be caused by a system-wide viral attack. The question is where it's going to hit, what can be done to prevent it, and how the computer industry will be changed as a consequence. . . .[6]

## Kinds of Destructive Software

Intentionally destructive software can come in several different guises, including computer viruses, and can be grouped into the following broad categories.

### Trojan Horses

These programs, like their historical namesake, have a purpose beyond their exterior appearance. While a Trojan horse program is executing, it may also be deleting files from a hard disk, or it may be performing any number of other disastrous actions. They are the most simplistic of the destructive software group, as their "hidden agenda" is usually readily apparent. Unfortunately, without prior precautions, the discovery is usually made too late.

### Logic Bombs

These programs are intended to perform an unexpected (by the user) function at a logically determined (by the program) time. Many trigger mechanisms exist for this type of destructive software. These mechanisms include:

1. A specified time period since installation has elapsed.
2. A certain predetermined operation has been selected by the user.
3. Some other predetermined combination of events that have taken place.

Because their destructive nature is not as readily apparent, logic bombs are slightly more diabolical than Trojan horses in that they may lull the user into accepting that the program is safe for use in a networked environment.

### Worm or Bacterium

These programs replicate and spread, but they do not attach themselves to other programs. Unlike viruses, they do not require a host computer system to survive and replicate. Worms usually spread within a single computer or through a network of computers. They are not spread through the sharing of programs. The most well-known example is the November 1988 Internet Worm, which infected and disabled several thousand government and university UNIX computers in a single day.[7]

The Internet Worm is worthy of a short diversion from the main topic of the paper. This particular worm took advantage of flaws in standard software installed on many UNIX systems, as well as mechanisms used to simplify sharing of resources in local area networks. Once established on a computer, the worm first gathered information about other machines connected in the network by reading public configuration files and executing system utility programs. Through this information, the worm could establish the status of networked nodes. It then attempted to load itself onto the networked systems by exploiting flaws in system software and by guessing user passwords. By exploiting the tendency of users to use common words as passwords, the worm achieved a 50% success rate at some sites.[8]

It should be noted that the worm did not infect other software. It simply slithered its way through the defenses of the computer systems and took up residence on the victimized

machine. Services were disrupted on these machines and system CPU cycles were wasted processing the worm code.

### Computer Virus

These programs are somewhat like a cross between a Trojan horse and a worm, with one deadly exception; not only do they mask their true purpose, they multiply and infect other software, making them carriers as well. A virus may also take on some of the properties of a "logic bomb." The virus may lay dormant for some time until a particular sequence of events awaken it. Given these properties, it is safe to say that computer viruses have an extremely high potential for destruction.

## Thesis Research Setting

The original thesis research on which this paper is based was conducted using the Apple Macintosh personal computer hardware platform. This setting provided a means of isolated experimentation. It also is an area where there is currently much virus activity. These factors combined to make the Apple Macintosh an ideal setting for computer virus research.

Destructive programs in the mainframe arena wreak havoc for a short period of time until they can be effectively eliminated. This area is where the most potential for destruction exists, but the lack of isolation and the short life span of viruses here make research difficult.

By using a personal computer, however, it was a simple matter to gain the needed distance from other users, thereby allowing the research to be conducted in an isolated environment. This advantage is in reality the primary problem in completely eliminating a viral strain from the personal computer arena. Since personal computers are so widespread, it is virtually impossible to completely obliterate a viral strain. Changes in the operating system or the hardware architecture could make the virus obsolete, but most personal computer viruses enjoy a long lifespan. The research conducted is still applicable to the mainframe environment since the properties of computer viruses in general are much the same.

### Methodology

To understand the risk that computer viruses pose to the computing community, the relative ease with which computer viruses can be written had to first be established. For this reason, a computer virus was written during the course of this research. It was assumed that this virus would be typical of other computer viruses in complexity and in the amount of technical knowledge required.

To comprehend the difficulties that may arise when attempting to infect a computer operating system, an operating system was infected using the virus program created during the research. The amount of damage inflicted upon the system was limited by the code of the virus, operating system constraints, and security measures exercised by the user of the system.

By examining the coding considerations of the virus, several security measures were illuminated that could have limited the spread of the virus or protected the system from loss of data.

The computer virus code was then examined to determine the amount of overhead required to support the virus infection. Three factors were considered in the calculation of the overhead required by the virus. These factors were: execution speed, memory requirements, and disk storage.

Finally, the virus written for the research was explored from the standpoint of an observer who did not have access to the technical specifics of the execution of the virus. The results obtained from this exploration were then generalized to provide a picture of how the virus appears to operate and what security measures could be taken to control the virus.

## Summary of Findings

The virus written during the course of this research was developed in approximately three weeks. The author held an undergraduate degree in Computer Science (at the time that the program was written) but had very little experience in programming the computer system that was chosen for infection. Software developers with extensive programming experience in a specific environment should be able to develop similar viruses in much less time.

The viral code developed consists of only three small modules totalling 2486 bytes (approximately 2.43K) of executable code.

The virus written during the research was intended to be a fairly simple one. Many ways were discovered that could have made the virus much more difficult to isolate and eradicate. Most of these discoveries were left out of the discussion since their dissemination could aid would-be virus writers in generating more virulent computer viruses.

The difficulties encountered in containing a virus were also noted during the research. Even a virus written purely for the sake of curiosity can be a threat if it somehow escapes into the computing community. When testing the virus written for this research, it was sometimes difficult to keep track of which applications were currently infected with the virus. If only one infected application was executed on another machine, the virus could have proliferated unchecked on machines that did not contain adequate security controls to inhibit this dissemination.

The virus written for this research can easily infect a system and will rapidly propagate itself throughout the entire computer system. Only two requirements were found to allow this to occur. First, the computer system must have inadequate security controls for dealing with the virus. Second, an infected application must be run on the computer system, either from attached media (such as a diskette), or from a remote server in a local area network.

The virus created for this research was benign in that no damage was intended; however, the infected computer user must still contend with the overhead imposed by the virus and the time and effort that will eventually be required to eradicate the virus. It was also pointed out in the research that not all viruses are benign. Some viruses cause erratic system behavior or permanently destroy data. Some may also be written by inexperienced or careless programmers, who leave bugs in their code that do inadvertent damage or have inadequate controls for handling error conditions.

Based upon the research that has been conducted, the threat posed to the computing community by computer viruses is very serious indeed. Developing a computer virus seems to require more persistance than technical expertise.

## Origins of Viruses

The origin of computer viruses can be traced to a 1949 paper by John von Neumann on the "Theory and Organization of Complicated Automata." In this paper the concept of a self-reproducing program was first discussed.[9]

Self-replicating programs were developed in several computing research sites by 1960. By the late 1960s, powerful Trojan horse programs had begun disrupting operations at major data processing centers. These disruptions have continued to the present day. At 7:30 a.m. on April 11, 1980, all IBM 4341 computers already delivered to customers ceased operation. The disruption was caused by the presence of a logic bomb in the systems, placed there by a disgruntled IBM employee.[10]

In 1984, Fred S. Cohen carried out controlled viral infection experiments at the University of Southern California. He found that viruses could infect an entire network of computers in a matter of minutes and concluded that it is impossible to write a computer program that can detect every conceivable virus.[11]

"Core War," a software program, written by A. K. Dewdney in 1984 simulates a computing environment where rival programs battle for control of a computer system. Although the game is not a virus, and the action takes place within the confines of the game software, it graphically illustrates how a computer system can become a virtual battleground pitting virus against virus-protection software.[12]

Virus-L is a virus discussion list available through BITNET that provides a forum for sharing information about computer viruses. It was founded by Kenneth van Wyk of Lehigh University in Bethlehem, Pennsylvania, following an outbreak of a virus on the university's IBM-PC population in November 1987. Virus-L currently has 1135 subscribers, representing 26 countries and 631 computing installations worldwide.[13] The forum provides information on current virus sightings, virus prevention, current research, and questions and answers on a variety of concerns relating to computer viruses.

The British Computer Virus Research Centre (BCVRC) has been founded to collect and catalogue viruses, analyze these viruses, disseminate pertinent results of this analysis to researchers, and produce support programs intended to reduce the risks of computer virus infection.[14] The BCVRC is an active participant in the BITNET Virus-L discussion list.

Finally, The Sophco Company of Boulder, Colorado, has established a Center for Computer Disease Control.[15]

## General Behavior of Computer Viruses

Computer viruses behave very much like their biological counterparts. The American Heritage Dictionary of the English Language defines a virus as:

> 1. Any of various submicroscopic pathogens consisting essentially of a core of a single nucleic acid surrounded by a protein coat, having the ability to replicate only inside a living cell. 2. Any specific pathogen. 3. Something that poisons one's soul or mind.[16]

The important part of this definition is that a virus is capable of replicating itself. Also note that replication is possible only in the presence of a "living" host, which differentiates a virus from a worm. Both of these points carry over well to computer viruses. They also are self-replicating and need a host system to survive.

### Phases of Virus Infection

Computer viruses generally pass through four unique phases during the course of their existence in a computer system.[17] These phases are as follows:

1. Dormancy phase (optional)
2. Propagation phase
3. Triggering phase (optional)
4. Damaging phase (optional)

During the dormancy phase, the user may be lulled into believing that the software containing the virus is safe. The operating system of the computer is infected, but no additional damage is inflicted during this phase, and the code does not propagate itself to other software. The Macintosh SCORES virus lies dormant for two calendar days before progressing to the next phase.[18]

The propagation phase is the only phase necessary for a program to be labeled a virus. During this phase, the virus attempts to attach itself to other applications or data files within the system. The main purpose is to have the attached viral code executed prior to the execution of the infected software. Once the code is attached to other software, it may be spread to other machines via the transfer of floppy disks or across a network.

An optional triggering phase determines when the true purpose of the virus is revealed to the user. Like logic bombs, the triggering mechanism is limited only by the imagination of the software author. Common mechanisms have been based on the date and time, the number of times the virus has replicated itself, or the number of executions since infection.

Finally, the virus performs its intended purpose. This purpose could be blatant, as is the case with the IBM-compatible "Israeli" or "Friday the Thirteenth" virus. Among other damaging features, when triggered, this virus will delete any applications that are executed.[19] However, the purpose is not always so obvious. Once again, the damage is limited only by the imagination. By changing bits of information in sensitive areas of memory (such as the operating system), the system is made to exhibit erratic behavior. By changing bits within a spreadsheet file, the spreadsheet may no longer be used or could contain erroneous information.

### Virus Classification

One way to classify viruses is by the extent of damage inflicted. In this manner, viruses may be considered either benign or malignant.[20] The Macintosh nVIR virus, for example, is programmed to speak the words "Don't Panic" when triggered and is considered benign.[21] On the other hand, the "Friday the Thirteenth" virus described above is definitely classified in the malignant category.

This distinction is somewhat misleading, however, in that even benign viruses have a potential for destruction. If nothing else, a virus consumes CPU time that could be put to more constructive use. When the virus is detected, time and effort must be expended to eradicate it from the infected system. A survey of 1000 *MacWorld* readers revealed that of the 42 that had experienced a viral infection, only 35% were able to eliminate the virus in less than two hours, and some were not able to eliminate the virus after more than 20 hours of work.[22]

Finally, it must be noted that the user of an infected application is at the mercy of the virus writer. If the writer did not take sufficient precautions for handling all error conditions that could arise while the virus is executing, valuable data can be lost or rendered unusable. In short, there is no such thing as a "safe" virus.

## Anatomy of a Typical Virus

As stated previously, a program must be able to propagate itself to be considered a virus. This is really the only requirement. To propagate, the virus must attach itself in some way to the operating environment of the computer. Once attached, the virus is then free to infect other applications. These applications may then be carried on floppy disks to other computers, or the applications may be run on other machines across a local area network (Figure 1). Either way, the applications then can infect other operating systems, and the cycle continues. There must be this two-way infection process: system -> application and system <- application.

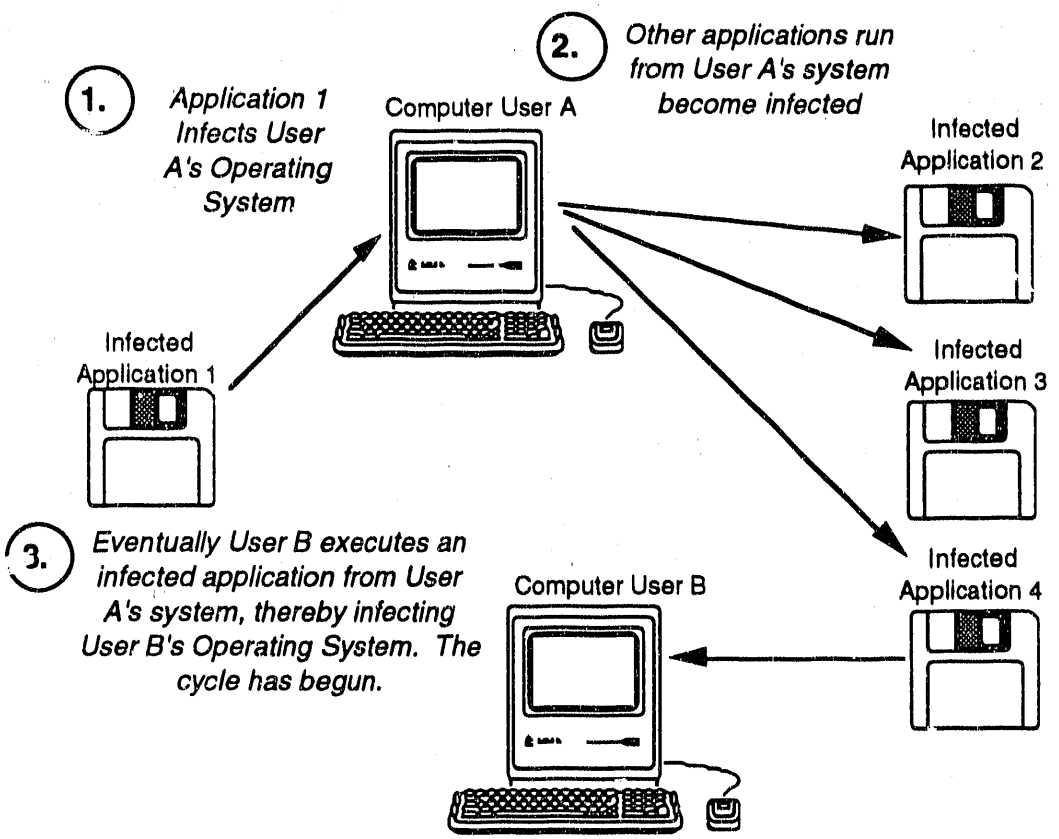


**Figure 1. Example of Virus Propagation.**

### System -> Application Infection

Because the act of virus propagation is circular in nature, any discussion of the process must begin by assuming that either the system or the application is already infected. For this discussion, it is first assumed that the operating system is already infected by a virus.

The primary purpose when infecting an application is to get viral code executed some time during the execution of the application. The viral code can be destructive or nondestructive, and it may incorporate a dormancy phase but at some point it will always attempt to propagate the virus (if the application is still infectious).

Assume that the operating system of a computer has become infected by a computer virus. The first step in propagating the virus is to decide when to target and infect applications. Some viruses attack as soon as a diskette is inserted into the disk drive; some attack while an uninfected application is executing; others may simply attack at random.

The following discussion shows how a "typical" computer virus can propagate through a computer system. Of course, this is not the only way that a virus can propagate, but it does serve as a good example.

Figure 2 illustrates how a normal operating system call can be implemented. The actual operational details of this call have been omitted from this paper.

Figure 3 shows how a virus might be able to infiltrate an operating system. Notice that viral code is now being executed within the execution cycle of the program.

### Application -> System Infection

The assumption made above was that an operating system had somehow been remapped to point to viral code. This section will describe how that could be accomplished.

When an infected application is executed, the viral code gains control. This code then attempts to infect the operating system of the computer on which it is executing. If it is determined that the operating system is already infected, no action is taken. If the system is not already infected, it can be infected by inserting code into the operating system such that this code will be executed at startup or another predetermined time.

The purpose of this inserted viral code is to remap one or more of the operating system calls (as seen in Figure 3). This completes the virus cycle of infection. The operating system infects applications, and applications in turn infect operating systems.

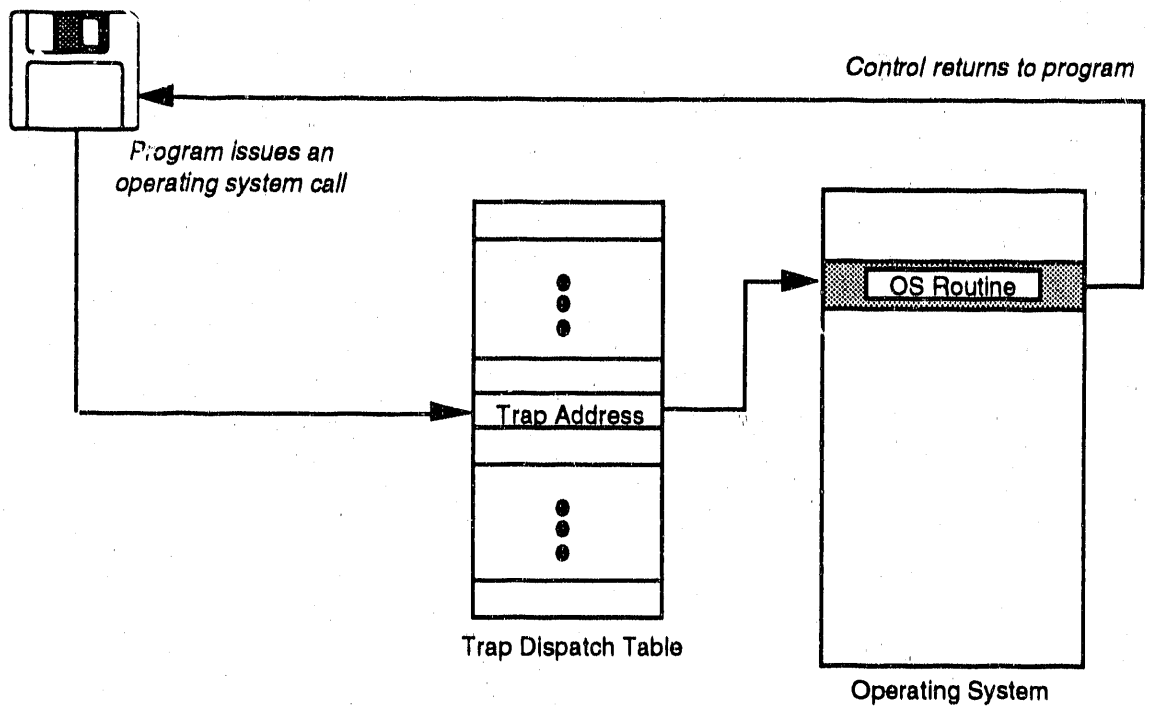


**Figure 2. Normal Operating System Call.**

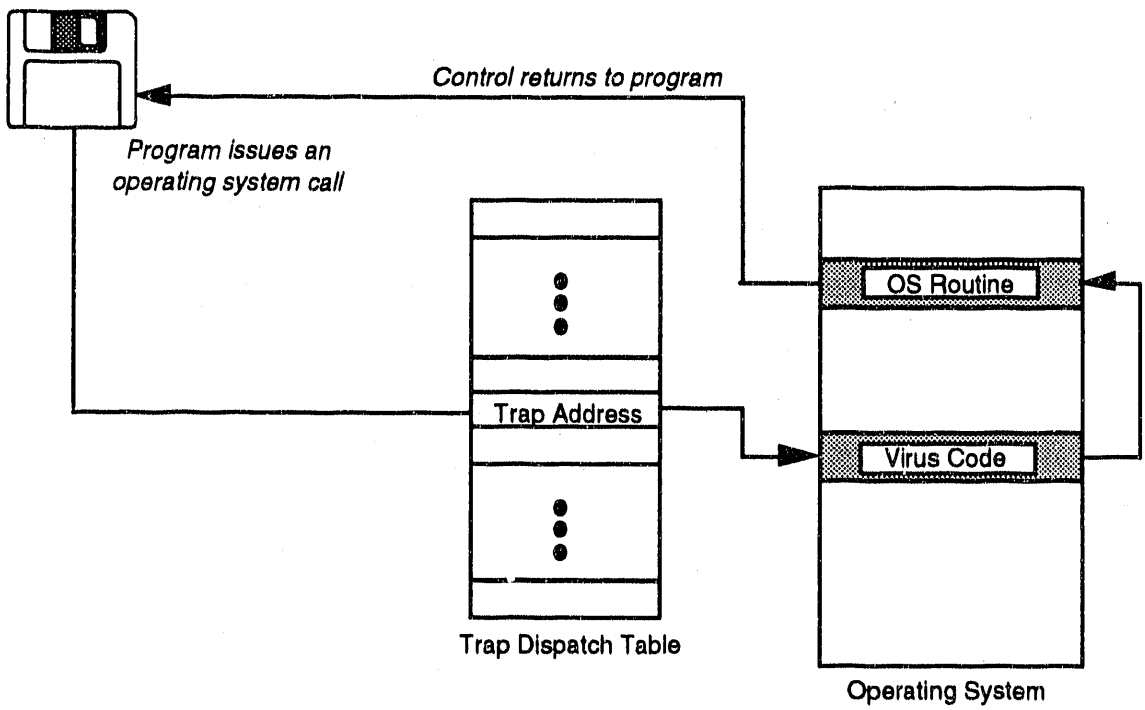


**Figure 3. Redirected Operating System Call.**

### Application -> System Infection

The assumption made above was that an operating system had somehow been remapped to point to viral code. This section will describe how that could be accomplished.

When an infected application is executed, the viral code gains control. This code then attempts to infect the operating system of the computer on which it is executing. If it is determined that the operating system is already infected, no action is taken. If the system is not already infected, it can be infected by inserting code into the operating system such that this code will be executed at startup or another predetermined time.

The purpose of this inserted viral code is to remap one or more of the operating system calls (as seen in Figure 3). This completes the virus cycle of infection. The operating system infects applications, and applications in turn infect operating systems.

### Overhead Incurred Because of a Virus

Once viral code has infected an application or an operating system, a certain amount of overhead is imposed upon the software. This overhead is a result of attempts by the viral code to continue propagation. The overhead manifests itself in three areas: processing time, memory demands, and disk space utilization.

As the virus interrupts the normal processing of the operating system and executing applications, it takes CPU cycles away from these activities, thereby lowering the throughput of the system. In a multiuser environment, so many cycles may be stolen that, in some situations, system performance is drastically degraded.

The virus code will have to be in main memory to be executed, therefore this memory will not be available to legitimate applications.

Finally, the virus code will add size to the operating system and to all infected applications. These storage demands will continue to grow with each fresh infection of an application.

## Motivations Behind the Generation of Computer Viruses

Given that viruses can be so terribly destructive, why would anyone ever write one in the first place?

Many viruses likely spring from an innate curiosity of programmers towards how computers work. Viewed in this vein, the publicity that viruses have received in the past only serves to encourage programmers to experiment with viruses. Viruses written in this fashion may be intended as harmless pranks. Self-proclaimed virus writer Roger Gonzalez says that his viruses ". . . don't destroy, they annoy. I wrote the programs as a challenge to myself, and to get back at a friend who played a practical joke on me."[23] These pranks usually only provide enjoyment for the author, not for the victim.

On the darker side of the picture, viruses can provide an extremely effective mode of sabotage for disgruntled or otherwise disturbed employees. Given that the destruction of some types of data can result in the loss of millions of dollars to an organization, it can be argued that viruses and other destructive software are the most easily available method of crippling an organization. Couple this with the difficulties that result from attempting to prosecute the perpetrators of this "white collar" crime, and the makings for a highly explosive situation exist.[24]

One graphic example is provided by Jim Pierce, former director for Globe Security in Philadelphia. An employee of a large oil exploration company planted a logic bomb in a proprietary software system used for exploration. The bomb had to be defused by the entry of a code word every 30 days. When the employee was terminated for unrelated reasons, the bomb went off, destroying millions of dollars worth of data and suspending exploration and pumping operations until the data was reentered manually.[25]

Viruses can also be a weapon of industrial sabotage. According to a March 15, 1988, article in the *San Jose Mercury News*, the MACMAG virus was found to have infected commercially-distributed copies of Aldus Freehand, a graphics program on the Apple Macintosh. These disks were then sold to Aldus customers. In this case, it appears that the infection was accidental, but it points out the opportunity that exists to damage the reputation of a competitor.[26]

The quest for recognition or notoriety may also be a motivating factor behind the writing of computer viruses and other destructive code. This appears to have been the motivation behind the actions taken by Robert Morris, Jr., the alleged author of the Internet Worm. Morris has achieved notoriety throughout the computer industry and national exposure as well. He is currently the subject of a grand jury investigation by the United States Attorney for the Northern District of New York.[27]

## Legal Implications

Against the backdrop of proliferating viruses and increasing indignation from a computing community forced to defend itself from attack, several pieces of legislation have been introduced in hopes of providing relief.

In 1986, the U. S. Congress passed the "Computer Fraud and Abuse Act of 1986," which makes it a felony to gain unauthorized access to classified information and a misdemeanor to access financial records or credit histories in financial institutions or trespass into a federal government computer.[28]

The General Accounting Office (GAO) issued a report following the "Internet Worm" incident that pointed out the difficulties faced when attempting to prosecute the writer of a computer virus. The GAO cited the fact that because no federal statute exists that specifically makes the writing of a virus a crime, virus writers must be prosecuted under other statutes. The "Computer Fraud and Abuse Act of 1986" is the statute that most closely relates to the act of writing of a computer virus. However, some terms in the statute are not well defined when applied to a computer virus.[29]

For example, the act defines the phrase "exceeds authorized access" as "access to a computer with authorization and use of such access to obtain or alter information in the computer that the accessor is not entitled to obtain or alter." However, the terms "access" and "information" are not specifically defined.[30]

The GAO report also cited the fact that the technical nature of virus-type incidents may hinder prosecution. Even when facing a clear-cut case of a violation under the existing statute, the evidence may be technical in nature and difficult to communicate to a jury.[31]

Robert Morris, the alleged author of the "Internet Worm," has been indicted under the "Computer Fraud and Abuse Act of 1986." According to the U. S. Department of Justice, this is the first federal prosecution of a crime involving a computer worm or virus.[32]

On July 14, 1988, Representative Wally Herger (R-Cal) introduced the "Computer Virus Eradication Act of 1988," which introduced civil and criminal punishment for anyone who "knowingly inserts into a program for a computer, or a computer itself, information or commands, knowing or having reason to believe that such information or commands may cause loss, expense, or risk to health or welfare."[33] This bill is intended to act as a deterrent to those who may intend to unleash a virus and to fill holes in existing legislation as they relate to computer viruses.

The "Computer Virus Eradication Act of 1988" was originally cosponsored by Representative Robert Carr (D-Mich) after the Macintosh computer in his office became infected with the Scores virus.[34]

In a report issued to Congress on December 15, 1988, Robert Helfant and Glenn J. McLoughlin suggested that "Congress could direct that research and development of techniques and standards directed at controlling viruses be undertaken" as a possible nonprosecution strategy.[35]

## Currently Active Viruses

The exact number of viruses actually virulent at a particular point in time cannot be accurately determined. However, there are many viruses that have been isolated and documented. A discussion of these strains is prudent before embarking more deeply into the world of computer viruses.

### Scores

This virus is also known as the "NASA Virus." It is the among the most virulent of known viruses in the Macintosh environment and is estimated to account for approximately one-third of known viral infections.[36] Programmers at the National Aeronautics and Space Administration (NASA) headquarters in Washington, D.C., were among the first to spot Scores in the spring of 1988. The virus has also infected computer systems at the U. S. Environmental Protection Agency, the National Oceanic and Atmospheric Administration, and the U. S. Sentencing Commission. NASA reportedly contacted the FBI to investigate the outbreak.[37]

Scores was designed to target specific proprietary applications developed by Electronic Data Systems (EDS), causing erratic behavior and printing problems in these programs. EDS supplies computer and communications services to large corporations and government agencies. Scores appears to have been designed to behave as a benign virus except when it comes into contact with the two EDS programs. However, Scores contains several bugs that can cause system crashes and other unpredictable behavior. The two target applications were fortunately never released to the public.[38]

### nVir

The nVIR virus first appeared in Europe in 1987 and in the United States in early 1988.[39] The nVIR virus attacks the Apple Macintosh line of personal computers. nVIR was named after the resource files that it attaches to the applications and operating systems that it infects.

nVIR is considered a benign virus in that it exhibits no overt destructive characteristics. When a system is first infected, a counter is set at 1000. The counter is decremented by one each time the system is started up, and by two each time an infected application is executed. When this counter reaches zero, the dormancy phase ends. The system will now beep or speak the words "Don't Panic" (if a speech driver is installed in the system) with a probability of 1/16 on system startup. This anomaly will also occur with a probability of 15/128 when an infected application is run. A 1/256 chance exists that the system will beep or speak twice.[40]

### INIT 29

The INIT 29 virus is arguably the most virulent of Macintosh viruses. It first appeared in late 1988. INIT 29 is named after the resource that is adds to infected files. It differs from most Macintosh viruses in that the operating system is left untouched and that an application need not be executed to become infected. INIT 29 infects almost any file that it comes in contact with. Applications, documents, and system files can all become infected. The virus attacks files on any disk inserted into the infected system.[41]

INIT 29 is also considered benign although many users have reported printing problems and system errors caused by incompatibility with system startup applications. Its sole purpose appears to be to replicate.[42]

### ANTI

The ANTI virus first appeared in France in early 1989. The string "ANTI" appears within the code of the virus, which is how it got its name. It is less contagious than INIT 29 but more contagious than Scores and nVIR. Its threat has been lessened considerably since the introduction of Multifinder, Apple's pseudo-multitasking operating system. ANTI does not spread under Multifinder.[43]

ANTI is benign, content to simply spread itself to applications. Like INIT 29, the operating system is not infected.

### Brain Virus

The "Pakistani" or "Brain" virus first appeared on the campus of the University of Delaware in October 1987.[44] The volume labels of infected diskettes are changed to "@Brain," which is how the virus got its name. The "Pakistani" nomenclature is derived from the Pakistani address listed by the virus authors in the boot area of infected disks. The virus attacks IBM PC-compatible computers.[45]

The Brain Virus was written by two computer store owners in Lehore, Pakistan. Unbelievably, they included their name, address, and telephone numbers within the virus code.[46]

The Brain Virus spreads across boot sectors of DOS floppy diskettes. The virus only infects 5 1/4" floppy diskettes. The Brain Virus appears to be benign although some infected sites have reported problems. Its sole purpose seems to be propagation and changing the volume name of infected diskettes.[47]

### Israeli PC Virus

In December 1987, several viruses were discovered at the Hebrew University in Jerusalem.[48] Among them was the Israeli PC Virus that attacks IBM PC-compatible computers.

Each time an application with a file extension of ".EXE" is executed on a system infected by this virus, an additional 1808 bytes are added to the size of the application. Applications with a file extension of ".COM" are also extended, but only the first time they are executed. The virus lays dormant until any Friday the thirteenth after 1987, at which time any application executed on an infected machine will be erased.[49]

A January 31, 1988 article in the *New York Times* suggested that this virus "was apparently intended as a weapon of political protest," since the first Friday the thirteenth after 1987 fell on May 13, 1988. This date was the fortieth anniversary of Palestine's independence from British mandate. Yisrael Radai, a member of the Computation Center at the University of Jerusalem, discounts this fact as merely a coincidence. He feels that the virus has no political significance and was probably intended merely as a prank.[50]

### Lehigh Virus

The Lehigh Virus was first encountered at Lehigh University in Bethlehem, Pennsylvania, in November 1987.[51] This destructive virus attacks IBM PC compatible personal computers.

The virus spreads by altering the COMMAND.COM file, which is part of the MS-DOS operating system. A rather insidious component of the Lehigh Virus is that the length of the COMMAND.COM file is not changed by infection. In an uninfected COMMAND.COM file, 300 bytes are used for temporary stack space. The virus inserts code into this area, leaving the overall length unchanged. It does, however, alter the change date for the file, leaving a method for discovering infected disks.[52]

The Lehigh Virus is a malignant virus. It copies itself to four other disks before destroying the entire contents of the original diskette.

## Current Virus Detection/Elimination Strategies

Many virus detection/elimination products have surfaced since the surge of virus attacks. These products first appeared as public domain software because viral attacks were becoming a serious threat to the free exchange of computer programs. Recently, commercial products have begun to appear on the shelves of computer dealers.

In general, computer virus detection/elimination strategies fall into four broad categories. Each of these strategies have their own inherent limitations and applicability.

### Active "Inoculation"

Some viruses have documented characteristics concerning the exact host system resources that are affected. Just like their biological disease counterparts, some computer viruses can be fooled into thinking that the system has already been infected and will not attempt to reinfect the system.

### Detection of Viral Infection

Once a host system has been infected, it may begin to show characteristic symptoms. These symptoms can lead to a determination of the virus itself, or at least to the part of the host that has been infected.

### Detection of Viral Infection Attempt

Sometimes an attempt at infection can be thwarted before the system actually becomes infected. By supervising the activity of known points in the host system that are susceptible to virus attack, some viruses can be terminated before they have a chance to become a serious problem.

### Elimination of the Infecting Agent

If a host has been infected with a virus whose characteristic behavior is known, the virus can sometimes be eliminated without causing harm to the host.

# Recommendations

## General Recommendations

The following general recommendations provide guidelines for dealing with the problem of computer viruses.

1. Limit the use of public-domain software. This software is not subject to strict configuration controls and can easily contain a virus. The availability and wide distribution of programs in the public domain make them an attractive target for computer virus authors and also increase the likelihood that the programs may have picked up a virus from someone else. If using public-domain software, first test it in an isolated environment.
2. Take note of any strange occurrences. This doesn't mean to become paranoid, but simply to be an alert user. An inordinately long delay when loading a program or booting a computer system doesn't mean that a computer virus is present, but these delays could be symptoms of a computer virus. Files or disks that suddenly become corrupted should be noted. If unusual occurrences persist, examine the disk in question for known viruses. If none are found, it may still be prudent to reinstall program and system software on the disk to eliminate any virus that may have infected it.
3. If a virus is found on a disk, isolate it as soon as possible. Programs that have possibly been infected by the virus should not be executed until it has been determined that they are not infected. If the virus is known to the computing community, methods will likely be readily available to eradicate it. If the virus is of unknown origin, many "reverse-engineering" techniques are available for use in discovering its method of operation.
4. Where feasible, write-protect the system startup diskette. The hardware write-protection mechanism, such as provided on floppy disks, is preferred over software write protection schemes, which can be subverted by viral code.
5. Make use of software protection mechanisms on files where possible. If the system in use allows write-protection at the file level, lock all important system and application files and resources. Applications programs rarely modify their own executable code. Exceptions to this rule are some copy-protection schemes.
6. Use publicly available virus detection/elimination programs with caution. These programs can be invaluable in combating the spread of computer viruses, but they can also allow unobstructed infiltration by a computer virus. The use of these programs is highly recommended, but treat them the same as any other program in the public domain. Commercially available virus detection/elimination programs are also available and provide a much safer method of insuring the integrity of a system.
7. Keep up-to-date on currently active viruses. By knowing what to look for, you can spot a viral infection more quickly and take appropriate action.
8. Keep good backups. If data is corrupted by a computer virus, you may be able to recover it if sufficient backups are maintained.

### Recommendation to Other Researchers

The design and coding of a computer virus was a worthwhile research pursuit. This path is recommended for researchers exploring virus infiltration on other hardware platforms. Systems can be better protected when their weak points are studied in-depth.

### Recommendations to Installations Processing Classified or Vital Information

Computer viruses can compromise the integrity of the data contained on a computer system. This is especially important in installations where classified or vital information is processed. Vital information is defined here as information that is considered to be important to the continued operation of the installation. These recommendations also apply to information that is of a sensitive nature such as payroll and personnel records. The following are additional recommendations made to these installations.

1. Use RAM disks when connecting to classified systems or systems where sensitive information is accessible. By placing the system software and communications software in volatile RAM memory, any classified or sensitive information covertly stored to disk during the communications session will be erased when the system is restarted. This sanitization will ensure that no classified or sensitive data can be captured by a virus or Trojan horse program and later retrieved.
2. Installations processing classified or vital information should already have strict access control procedures in place. These measures must be followed to ensure that virus programs are not installed on machines that process this information. If unobstructed access to a computer system takes place, all other security measures are in vain.
3. The general recommendation made for the limited use of public-domain software must be more stringent for installations processing classified or vital information. Public-domain software still has a place in these organizations, but more care should be taken. The software should be examined carefully before it is placed on computers cleared to process classified or vital information.
4. Virus protection software should be mandatory in installations processing classified or vital information. One or more standard virus protection programs should be distributed to all users cleared to process the information.
5. Computer security personnel should be well educated in the area of computer viruses. Users concerned that their machine is infected should be able to contact a central organization that understands how viruses operate, as well as the symptoms of currently active viruses. This group should not overreact to a possible infection citing. Overreaction only serves to complicate the problem. A potential virus should not be handled as a major security problem. The computer security group should inform other users of a potential virus and then quietly investigate the citing further.

### Recommendation to Would-Be Virus Authors

Do not write computer viruses except in an effort to understand them and thereby combat their spread. While technically challenging, the writing of a computer virus does not imply any special technical achievement. The computing field contains many more technically challenging pursuits that benefit the computing community. Computer viruses pose a serious problem. Everyone who enjoys the computing field should assist in eliminating the spread of computer viruses.

### Recommendation to Other Computer Users

Finally, computer users should resist the temptation to get caught up in the current paranoia surrounding computer viruses. A few years ago, periodic system malfunctions were simply shrugged off simply as an annoyance. With the proliferation of computer viruses, each system malfunction is now often suspect as a manifestation of a computer virus. The chances of becoming infected by a computer virus are not great, if the user is accustomed to safe computing practices. By remaining alert to the symptoms of computer viruses and understanding how viruses operate, computer users should be able to relieve this computer virus anxiety.

## Conclusions

Developing a computer virus can be likened in many ways to building a bomb. All of the information necessary for its construction are readily available. The technical proficiency required is substantial, but not beyond the reach of any reasonably intelligent computer hacker. The only missing element is a propensity for destruction on the part of the computer virus author.

Computer viruses can easily infect unprotected computer systems and there is no limit to the amount of damage that can be inflicted once a system becomes infected. Since computer viruses are simply malicious computer programs, anything that can be done on a computer can be done within the context of a computer virus. The damage inflicted in this manner is limited only by the virus author's imagination.

Computer viruses must be dealt with in a manner similar to any other computer security problem. The paranoia that exists in the computing community relating to computer viruses only serves to compound the problem. Although computer viruses are more difficult to detect and deal with, their infiltration methods are similar to those of any other type of destructive software. Computer viruses are more dangerous only because they are capable of replicating. If a computer system is protected such that it cannot be initially infected by a computer virus, then this capacity does not pose a further problem.

The authors of computer viruses are members of the community of computer users. Computer viruses pose a threat to this community. This threat manifests itself first in the destruction of information and disruption of computer operation and ultimately by endangering the free exchange of ideas among members of the community. Would-be computer virus authors should consider this before placing everyone at risk.

### Opportunities for Further Research

Many other opportunities exist for continued research in the area of computer viruses.

The design of future microcomputer operating systems should encompass some form of protection against the infiltration of the system by a computer virus. Research can be directed toward designing such an operating system.

The legal sector is currently struggling to restructure laws that provide criminal prosecution of the authors of destructive code to cover the area of computer viruses. Research can be directed toward assisting in this effort.

A general scheme for protecting current microcomputer hardware platforms from infection by computer viruses can be investigated. This level of protection will most likely involve a layered approach including both hardware and software components. Software protection alone can always be overcome by subverting the integrity of the software system.

Finally, the study of the weaknesses of any computer system is a valuable area for research. The information gained during the research can immediately be put to use to strengthen the weak areas.

## References


1. Ornstein, Severo, "ACM Forum: Beyond Worms," Communications of the ACM, Vol. 32, No. 6, June 1989, p. 672.
2. Dewdney, A. K., "Computer Recreations," *Scientific American*, March 1989, p. 110.
3. Lepage, Rick and Ford, Rick, "Zooming out from viruses," *MacWeek*, April 4,1989, p. 10.
4. Highland, Dr. Harold Joseph, FICS, "From the Editor," *Computers & Security*, June 1989, p. 272.
5. Kiel, Stephen E. and Lee, Raymond K. , "The Infection of PC Compatible Computers," Georgia Institute of Technology, Summer Quarter 1988, p. 8.
6. Crino, Michael D. , and Leap, Terry L. , "What HR Managers Must Know About Employee Sabotage," *Personnel*, May 1989, pp. 35-36.
7. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, p. 2.
8. Spafford, Eugene H., "Crisis and Aftermath," *Communications of the ACM*, Vol. 32, No. 6, June 1989, pp. 678-680.
9. von Neumann, John, *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana, IL,1966, pp. 31-74.
10. Menkus, Belden, "The Computer Virus Situation is not Encouraging," *Computers & Security*, Vol. 8, No. 2, June 1989, p. 115.
11. Dewdney, A. K., "Computer Recreations," *Scientific American*, March 1989, pp. 111-112.
12. Dewdney, A. K., "Computer Recreations," *Scientific American*, March 1989, p. 111.
13. van Wyk, Kenneth, personal communication, August 4, 1989.
14. Hirst, Joe, "National Virus Research Centres," *Virus-L Digest*, Vol. 2, No. 162, p. 1.
15. Helfant, Robert and McLoughlin, Glenn, "CRS Report for Congress: Computer Viruses: Technical Overview and Policy Considerations," Congressional Research Service, The Library of Congress, 88-556 SPR, December 15, 1988, p. 12.
16. American Heritage Dictionary of the English Language, Copyright 1985, Houghton Mifflin Company, Boston, Massachusetts, p. 1351.
17. Greenberg, Ross M., "Know Thy Viral Enemy," *Byte*, June 1989, p. 276.
18. Stefanac, Suzanne , "Mad Macs," *MacWorld*, November 1988, p. 94.
19. Radai, Yisrael, "The Israeli PC Virus," *Computers & Security*, Vol. 8, No. 2, June 1989, p. 112.
20. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989. pp. 1-2.
21. Coale, Kristi, "Razor Blades in Apples," *MacUser*, September 1988, p. 310.
22. Davis, Gil, "Macworld News," *MacWorld*, October 1988, p. 99.
23. Gonzalez, Roger, *BITNET VIRUS-L discussion list*, April 25.
24. Gemignani, Michael, "Viruses and Criminal Law," *Communications of the ACM*, Vol. 32, No. 6, June 1989, pp. 670-671.
25. Crino, Michael D. , and Leap, Terry L. , "What HR Managers Must Know About Employee Sabotage," *Personnel*, May 1989, p. 35.
26. Brower, Emily, "There are bugs . . . then there are viruses," *MacWeek*, Feb. 16, 1988, p. 8.
27. Eisenberg, Ted, David Gries, Juris Hartmanis, Don Holcomb, M. Stuart Lynn, Thomas Santoro, "The Cornell Commission: On Morris and the

Worm," *Communications of the ACM*, Vol. 32, No. 6, June 1989, pp. 706-707.
28. Helfant, Robert and McLoughlin, Glenn, "CRS Report for Congress: Computer Viruses: Technical Overview and Policy Considerations," Congressional Research Service, The Library of Congress, 88-556 SPR, December 15, 1988, p. 11.
29. "Virus Highlights Need for improved Internet Management," *United States General Accounting Office*, GAO/IMTEC-89-57, June 1989, pp. 32-33.
30. "Virus Highlights Need for improved Internet Management," *United States General Accounting Office*, GAO/IMTEC-89-57, June 1989, pp. 33.
31. "Virus Highlights Need for improved Internet Management," *United States General Accounting Office*, GAO/IMTEC-89-57, June 1989, pp. 34.
32. Alexander, Michael, "Morris indicted in Internet virus affair," *Computerworld*, July, 31, 1989, p. 8.
33. Herger, Wally, "H.R. 55," 101st Congress, 1st Session, July 14, 1988.
34. Winter, Christine, "Legislators Alerted to Computer Virus Danger," *The Washington Post*, Friday, October 14, 1988.
35. Helfant, Robert and McLoughlin, Glenn, "CRS Report for Congress: Computer Viruses: Technical Overview and Policy Considerations," Congressional Research Service, The Library of Congress, 88-556 SPR, December 15, 1988, p. 12.
36. Barron, Janet, "Two Mac Viruses," *Byte*, June 1989, p. 278.
37. Stefanac, Suzanne , "Mad Macs," *MacWorld*, November 1988, p. 97.
38. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, pp. 12-13.
39. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, pp. 12-13.
40. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, pp. 12-13.
41. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, pp. 12-13.
42. Schmitt, Henry, "The Virus Encyclopedia," The NorthWest of Us, 1988, p. 23.
43. Norstad, John, "Disinfectant Version 1.0," *Northwestern University*, March 19, 1989, pp. 12-13.
44. Highland, Joseph, "Reports from the Victims," *Computers and Security*, Vol. 8, No. 2, April 1989, p. 101.
45. Webster, Anne, "University of Delaware and the Pakistani Computer Virus," *Computers and Security*, Vol. 8, No. 2, April 1989, p. 103.
46. Winter, Christine, "Legislators Alerted to Computer Virus Danger," *The Washington Post*, Friday, October, 14, 1988.
47. Chess, David, "Viruses in MS-DOC / PC-DOS," *Virus-L Digest*, April 22,1988, p. 28.
48. Highland, Joseph, "Reports from the Victims," *Computers and Security*, Vol. 8, No. 2, April 1989, p. 101.
49. Radai, Yisrael, "The Israeli PC Virus," *Computers & Security*, Vol. 8, No. 2, June 1989, p. 111.
50. Radai, Yisrael, "The Israeli PC Virus," *Computers & Security*, Vol. 8, No. 2, June 1989, p. 111.
51. Highland, Joseph, "Reports from the Victims," *Computers and Security*, Vol. 8, No. 2, April 1989, p. 101.

52. van Wyk, Kenneth, "The Lehigh Virus," *Computers and Security*, Vol. 8, No. 2, April 1989, pp. 107-108.

## INTERNAL DISTRIBUTION

1. J. L. Arrowood
2-6. David R. Brown
7. H. P. Carter
8. L. F. Denton
9. W. E. Ford, III
10. R. P. Leinius
11. P. J. Mason
12. W. J. McClain
13. E. J. Nall
14. R. D. Phillips
15. D. H. Pike
16. B. T. Rhyne
17-19. C. H. Shappert
20. R. L. Shipp
21. R. E. Textor
22. G. E. Whitesides
23-25. A. R. Wilson
26. Central Research Library
27. ORNL Y-12 Research Library Document Reference Section
28. Laboratory Records Department
29. Laboratory Records ORNL (RC)
30. ORNL Patent Office

## EXTERNAL DISTRIBUTION

31. Office of the Assistant Manager for Energy Research and Development, U.S. Department of Energy Field Office, Oak Ridge (DOE-OR), P.O. Box 2001, Oak Ridge, TN 37831
32-41. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831

# END

DATE
FILMED
4/30/92