NORMAN®

The Norman Book on Computer Viruses

Norman ASA
Mailing address: P.O. Box 43, N-1324 Lysaker, Norway   Physical address: Strandveien 37, Lysaker
Tel. +47 10 97 00 Fax. +47 67 58 99 40 E-mail: norman@norman.no

Norman Data Defense Systems Inc
9302 Lee Highway Suite 950a, Fairfax, VA 22031, USA
Tel. +1703 267 6109 Fax. +1703 934 6367 E-mail: norman@norman.com

Norman Data Defense Systems GmbH
Kieler Str. 15, D-42697 Solingen, Germany
Tel. +49 212/26718 0 Fax. +49 212/26718 15 E-mail: norman@norman.de

Norman/SHARK BV
Mailing address: P.O. Box 159, NL-2130 AD Hoofddorp, The Netherlands
Tel. +31 23 563 3960 Fax. +31 23 561 3165 E-mail: sales@shark.nl

Norman Data Defense Systems AG
Postfach, CH-4015 Basel, Switzerland
Tel. +41 61 487 25 00 Fax. +41 61 487 25 01 E-mail: norman@norman.ch

Norman Data Defense Systems Pty. Ltd.
6 Sarton Road, Clayton, Victoria, 3168 Australia
Tel. +61 3 9562-7655 Fax. +61 3 9562-9663 E-mail: norman@norman.com.au

Disclaimer

Norman ASA is not liable for any other form of loss or damage arising from use of the documentation or from errors or deficiencies therein, including but not limited to loss of earnings.

In particular, and without the limitations imposed by the licensing agreement with regard to any special use or purpose, Norman ASA will in no event be liable for loss of profits or other commercial damage including but not limited to incidental or consequential damages.

The information in this document as well as the functionality of the software is subject to change without notice. No part of this documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the explicit written permission of Norman ASA.

The Norman logo is a registered trademark of Norman ASA.

Names of products mentioned in this documentation are either trademarks or registered trademarks of their respective owners. They are mentioned for identification purposes only.

Norman documentation is

Copyright © 1999 Norman ASA.

All rights reserved.

March 1999

Table of Contents

Copyright © 1999 Norman

Introduction

It’s hard to believe that the first IBM personal computer (PC) was introduced in August, 1981. In the beginning they were used by a small group of people. Today, however, we can’t imagine life without them, both at work and at home. Look around your office when the electricity goes out, and you’ll see people standing around talking because they feel they can’t get any work done without their computers.

We have become dependent on these machines and the information stored within. As the importance of a “thing” rises, it becomes equally as important, if not more, to secure it. (How many of you have alarm systems in your cars?)

A large portion of modern computing life is to secure the information that we are creating and processing. There are many aspects of information security, ranging from physical access to ensuring that the information has not been changed in any way.

One of the most high-profile threats to information integrity is the computer virus. Surprisingly, PC viruses have been around for two-thirds of the IBM PC’s lifetime, appearing in 1986. With global computing on the rise, computer viruses have had more visibility in the past two years. In fact, the entertainment industry has helped by illustrating the effects of viruses in movies such as “Independence Day”, “The Net” and “Sneakers”.

Please note that computer viruses are also found on Macintoshes, but in this book, we will focus on PC viruses. The topics we will cover are:

- what a virus is
- the evolution of the virus problem
- viruses on different operating systems
- solutions to the virus problem
- how Norman Virus Control products help

What is a Virus?

The terms “computer virus” and “virus” are used very loosely in everyday conversation and have become synonymous with “trouble”.

Viruses, worms, Trojan horses, and logic bombs are all unwanted, uninvited, potentially dangerous software, but there are important distinctions among them. The differences lie in whether the category requires a host program and whether it makes copies of itself. All four may cause damage, but this is not integral to the definitions.The following overview defines each category:

Virus

Viruses require a host, and their goal is to infect other files so that the virus can “live” longer. Some viruses perform destructive actions although this is not necessarily the case. Many viruses attempt to hide from being discovered.

Note: Viruses are simply software programs.

Replicates?

Yes. All viruses make copies of themselves, infecting system boot sectors, master boot sectors, programs, or data files as the opportunity arises.

Worm

A host is not required, because worms are typically a mainframe problem and do not need to hide from most users.

Replicates?

Yes. A worm makes copies of itself as it finds the opportunity.

Trojan horse

Does not require a host. While the term “Trojan horse” sometimes refers to the program containing destructive code, the term is more often used to refer to the entire .COM or .EXE.

Replicates?

No. Most Trojan horses activate when they are run and often destroy the structure of the current drive (FATs, directory, etc.), obliterating themselves in the process.

Bug, Logic bomb, Time bomb

Requires a host. Programmers cannot write a bug without also writing other code — although it's fair to say that most programmers do not intentionally write bugs. Logic bombs and time bombs are intentionally inserted in otherwise “good” code.

Replicates?

No. This code generally has better things to do than making copies of itself. Logic bombs and time bombs wish to remain hidden, with only their effects being visible. Bugs do just about everything except make more bugs.

Virus Types Overview

When speaking about viruses, we normally speak about four different types:

File virus

File viruses infect executables (program files). These are able to infect over networks.

Macro virus

Macro viruses infect data files. These are able to infect over networks.

Boot virus

Boot viruses infect boot sectors of hard drives and floppy disks. These are **not** able to infect over networks.

Multipartite virus

Multipartite viruses infect both executable files and boot sectors. These are able to infect over networks.

Because a macro virus infects files, it technically is a file virus. However, unlike traditional file viruses, it targets data files instead. Macro viruses are becoming increasingly common, and therefore they deserve to be treated as a separate category.

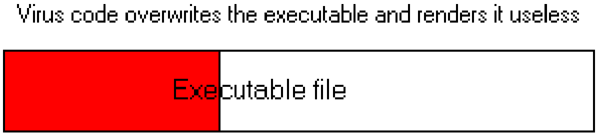
You have probably also heard other terms such as “polymorphic”, “stealth”, and “encrypted”. These are not types of viruses, per se, but rather are methods that viruses use to disguise themselves from anti-virus products.

The next sections describe file, macro, and boot viruses more thoroughly. Multipartite viruses are not common, and therefore will not be covered in this book.

File Virus

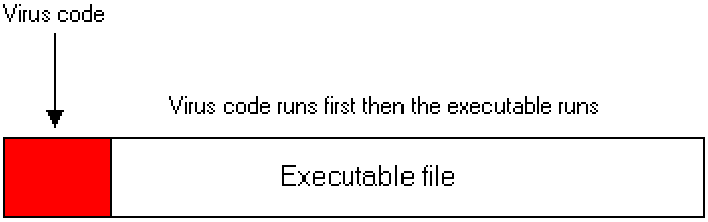
A file virus attaches itself to a program file (the host) and uses different techniques in order to infect other program files.

There are three basic techniques for infecting an executable file: overwrite, prepend, and append.



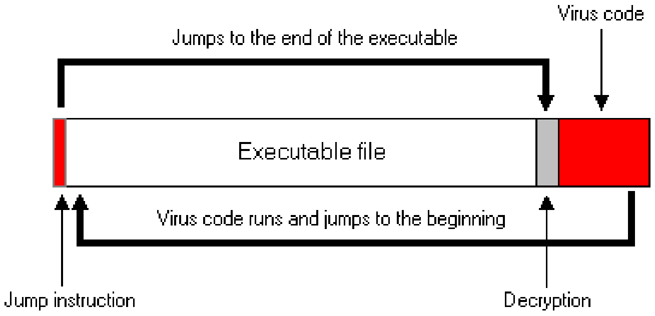
An overwriting virus places itself at the beginning of the program, directly over the original program code, so the program is now damaged. When you try to run this program, nothing happens except for the virus infecting another file.

Such viruses are easily apprehended and destroyed by users and user support staff, so they actually spread very poorly in the wild. You have almost no chance of ever getting an overwriting virus in your machine.



The pure prepending virus may simply place all of its code at the top of your original program. When you run a program infected by a prepending file virus, the virus code runs first, and then your original program runs.

Copyright © 1999 Norman


Virus code
Jumps to the end of the executable
Executable file
Virus code runs and jumps to the beginning
Jump instruction
Decryption

An appending virus places a “jump” at the beginning of the program file, moves the original beginning of the file to the end of the file, and places itself between what was originally the end of the file and what was originally at the beginning of the file. When you try to run this program, the “jump” calls the virus, and the virus runs. The virus then moves the original beginning of the file back to its normal position and then lets your program run.

That was a brief overview of how a virus attaches itself to a program file. It uses different techniques in order to infect. Most file viruses go memory-resident so that they can monitor all actions and infect other program files. Other file viruses infect by “direct action”, which means that they infect a program file when that program file is accessed.

Several other methods exist, but for the most part, file viruses like to go memory resident. If a file virus is memory resident, then it can be extremely easy to infect another program file by waiting until that program file is run and then attaching to that file. That file will then become infected (i.e., become a “carrier”), and it will go on to infect other program files.

Boot Virus

Boot viruses infect System Boot Sectors (SBS) and Master Boot Sectors (MBS).

The MBS is located on all physical hard drives. It contains, among other data, information about the partition table (information about how a physical disk is divided into logical disks), and a short program that can interpret the partition information to find out where the SBS is located. The MBS is operating system. The SBS contains, among other data, a program whose purpose is to find and run an operating system.

Because these system areas are read during the booting process on all IBM-compatibles, boot viruses are operating system-independent and are therefore able to propagate more effectively than file viruses.

Refer to "Viruses on Different Operating Systems" on page 19 for more information.

The Booting Process

To understand boot viruses, it is necessary to understand the booting process.

The BIOS (Basic Input/Output System), which controls the booting process, is initiated as soon as the power is switched on.

The next process that runs is called POST (Power On Self Test). It ensures that the computer is in working order. One POST function that all users will recognize is the display that counts the amount of RAM (Random Access Memory) in the machine.

POST's final act is to kick off the booting process. The first task is to determine whether or not there is a diskette in the floppy drive. If there is, the System Boot Sector on the floppy is read, and the machine attempts to boot from it.

If the diskette is not bootable (see below for more details), then you will see the following message on the screen:

```
Non system-disk or disk error.
Replace and strike any key when ready.
```

Normally, however, no diskette is present, and the Master Boot Sector on the hard drive is read. Then the System Boot Sector of the hard drive is read, and it will start the operating system.

This process remains the same on machines running DOS, Windows, Windows 95/98, Windows NT, and OS/2. The differences appear when the operating systems themselves are loaded.

A Bootable Diskette

When a floppy disk is formatted, a System Boot Sector is created. The diskette can have two functions: contain program files and data files and/or be a bootable diskette.

A bootable diskette is one that can be used to bypass the booting process on the hard drive. Instead, the boot process runs from the diskette.

To create a bootable diskette, you must either format the diskette with the “system” option (/S) or you must use DOS’s SYS command on the diskette.

A formatted diskette always has a System Boot Sector regardless of whether or not the diskette is bootable. The SBS happens to be the place that a boot virus calls home, so any formatted diskette that you have can potentially be infected with a boot virus.

How a Boot Virus Infects

If a diskette is left in drive A: of a machine, and CMOS is set up to first boot from drive A: and then drive C:, then the SBS of the diskette will be read. If the SBS contains a boot virus, the boot virus will become active, go memory resident, infect the system areas of the hard drive, and attempt to infect other write-enabled diskettes that are accessed.

People tend to leave floppies in the drive when they turn their machines off for the day and then forget about them when they turn their machine on in the morning. Consequently, boot viruses are the majority of the most common viruses seen today.

Macro Virus

Since the introduction of the first macro virus in August 1995, this virus type has been the fastest growing category. The first time we discussed this phenomenon in this publication, in January 1997, the number of known macro viruses was 100. As of March 1999, Norman has identified some 4,000 macro viruses, and the number is growing at a disturbing rate. Corporations and single users need to protect themselves by frequent updates of their virus control tools, which in turn involves the anti-virus industry to constantly update and distribute definition files. A definition file holds the virus signatures (fingerprints of known viruses) and are used by the scanning engine to detect and remove computer viruses.

Any virus scanner is only as effective as its most recent update, so obtaining frequent virus signature updates is critical to maintaining a secure computing environment.

Definition file updates are available from Norman’s Web site on a regular basis. We recommend that you pay us a visit at:

http://www.norman.no/update.htm

The differences between macro viruses and traditional file viruses lie in the host (data files) and the method of replication (use of macro programming languages inherent to applications). **These differences add up to a new, formidable data security threat.** Throw in the increased use of OLE (Object Linking and Embedding) as well as the explosive use of networks, e-mail, and the Internet as exchange media, and the outlook is grim.

How It Works

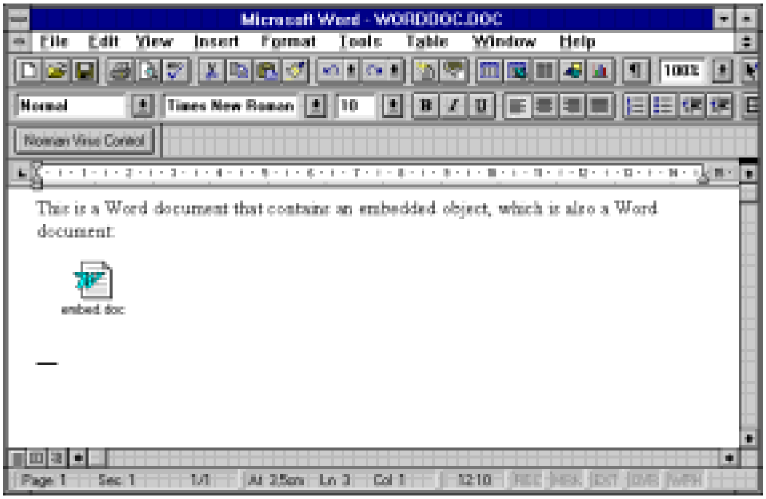
Traditional file viruses do not attempt to infect data files, for data files are not an ideal ground for replication. That is, one does not “run” a data file — one “reads” and “edits” a data file. However, in the past few years, organizations have been building upon open systems, in which data is shared more readily. This in turn means that there is little security. Macro viruses take advantage of the fact that many applications now contain **macro programming languages**. These languages allow users (and virus authors) more flexibility and power within the application than ever before. Often macro viruses are not detected early enough because many users are not familiar with the nuances of macros. As a result, macro viruses have an infection rate much higher than traditional file and boot viruses.

To date, the most targeted macro programming language is WordBasic, the language within Microsoft Word.

Why It’s Such a Risk

Since data files are shared more frequently than executable (program) files, the security threat posed by macro viruses is very real. The open systems in many of Microsoft's applications utilize OLE in order to combine different data types. You can **embed** an object such as a bitmap within a Word document. Embedding an object means that any edits to the object will not be reflected in any other copies of the object. You can also **link** an object such as a Excel spreadsheet to a Word document. Linking an object means that you may edit the object in either its source application or from within the application to which it is linked, and all copies of the object will be updated.

Copyright © 1999 Norman



This is a Word document that contains an embedded object, which is also a Word document:

MS Word

Microsoft Word has the ability to embed and link objects. In addition, Word documents have the ability to be embedded and linked in other applications. The risk, therefore, results in the ability to run a Word macro virus from another application. For example, Microsoft's MSMail messages can contain attachments such as Word documents. If the correct association exists, then the MSMail user can simply double-click on the Word document, Word will start, and the specified document will be loaded. This is only one example of OLE in action. There are other uses of OLE in conjunction with Word documents, and it is the frequency of such use that drives the scope of the Word macro security threat.

Some macro viruses contain destructive code and some even create and execute traditional file and boot viruses. While traditional file and boot viruses affect the operation of a machine, macro viruses affect the quality and reliability of **information** contained within data files.

MS Excel

It did not take long after the first Word Macro virus before the first Excel virus appeared: *XM/Laroux.A*. This was an event that was expected, as the techniques necessary to create such a virus are the same as for Word macro viruses.

The difference between viruses for Word and Excel is that viruses for Word are written in WordBasic, whereas viruses for Excel are written in VBA3 (Visual Basic for Applications version 3). The format is different, and the macros are not stored inside the spreadsheet (viruses for Word are stored in the Word document), but in separate streams. This technique complicates detection, identification, and removal.

Macro viruses for Excel pose a bigger threat than Word viruses, because of the possible practical implications. Imagine that a macro virus for Excel multiplies a certain cell by a factor 10 and that this particular cell specifies your salary. Definitely not the end of the world, but what if this cell was *divided* by 10?

These are minor inconveniences compared to similar changes to calculation formulas for estimating the strength of concrete for a skyscraper. Spreadsheets are often large, and anomalies are not easily recognized.

Office 97

The introduction of Office97 also included changed formats for almost all programs in the suite, but at least the changes were consistent. Both Excel and Word are using VBA5, based on VBA3 with many extensions.

VBA5 is not compatible with WordBasic, which should indicate that macro viruses written for previous versions of Word would not affect Word 8.0 in Office97.

However, Microsoft initiated a WordBasic to VBA5, and a VBA3 to VBA5 conversion to upgrade existing macros to the new formats.

Consequently, macro *viruses* for previous versions of Word and Excel can also be ‘upgraded’. Not every virus will work after the conversion, but we know that quite a few do.

Predictions for the Future

Norman expects that macro viruses still represent a serious threat to data security, even though there is reason to believe that the growth rate will flatten. We also expect to see viruses taking advantage of common macro languages, enabling the macro virus to become application-independent. (Remember that currently, Microsoft Word is the application most afflicted with macro virus infections because most macro viruses are written in WordBasic.) In addition, we expect macro viruses to utilize polymorphic and stealth techniques. Norman will continue to keep abreast of the macro virus problem, along with traditional file and boot viruses, by looking towards scanning for and removing macro viruses on the application level as well as on the binary level.

How Many Viruses Are There...

Anti-virus vendors are asked this question all the time. The answer is difficult for several reasons:

1. There is no central organization that counts the number of viruses.
2. New viruses appear every day. Some experts say that the growth of new viruses is exponential and others say that it’s quadratic. If we were able to count them all, then the count would be valid only for a short period of time such as a day.
3. Often we find that many variants are made based upon one virus, and often there is disagreement among the virus research community on the definition of “variant”.

4. There is no standard naming convention for viruses, and as a result it is possible to have several different names for the same virus.

   This brings up the question of how viruses get their names. Sometimes the virus author puts text into the virus that indicates a name for the virus or for him/herself (e.g., The xxx virus is here; Greetings from yyy). But most of the time, names are given by people who discover them. Different methods are used, such as: estimates of place of origin or place of detection (e.g., the Lehigh virus), number of bytes that the virus adds to files, what the virus does, and so on.

With those caveats in mind, Norman Virus Control products detect over 36,000 virus variants as of this writing (March, 1999).

...And Does It Matter?

To the ordinary user, it’s immaterial if the number of viruses is this or that, as long as your anti-virus software keeps your machine virus free. The number of known viruses does not really reflect what is going on in the virus makers’ world. Statistics are sometimes useful as a visible manifestation of the evolution of computer viruses, for example.

The computer virus problem is best evaluated by analyzing the nature of the individual virus and looking at what they do, rather than keeping track of how many there are. This may have been the message from the virus maker(s) who sent some 14,000 brand new viruses to anti-virus vendors all over the world (fall of 1998). All of these viruses had been generated automatically, and they were therefore not technically sublime. In fact, most of these viruses could be detected by heuristic methods. Nevertheless the *number* of virus signatures in our virus definition files almost doubled overnight, while the overall virus threat was unchanged.

In the Wild Viruses

Although virus researchers know of thousands of viruses, you need not worry about all of them. Of those thousands, most of them exist only in research labs, and the remaining handful are actually seen in homes and organizations around the world.

As a result, virus researchers group viruses into two categories: “in the wild” and “in the zoo”, sometimes referred to as “ITW” and “ITZ” respectively.

Viruses that are “in the wild” have been seen outside the research labs. In the wild viruses comprise about 10% of the viruses that we know about, and it is these viruses that you and your organization should concern yourselves with.

If you are interested in more details, contact your nearest dealer or Norman directly.

The Evolution of the Virus Problem

In the beginning, computers were not connected together very well, and computer viruses spread extremely slowly. Files were transmitted via BBSs (bulletin board systems) or on diskette. As a result, the transmission of infected files and boot sectors was geographically limited.

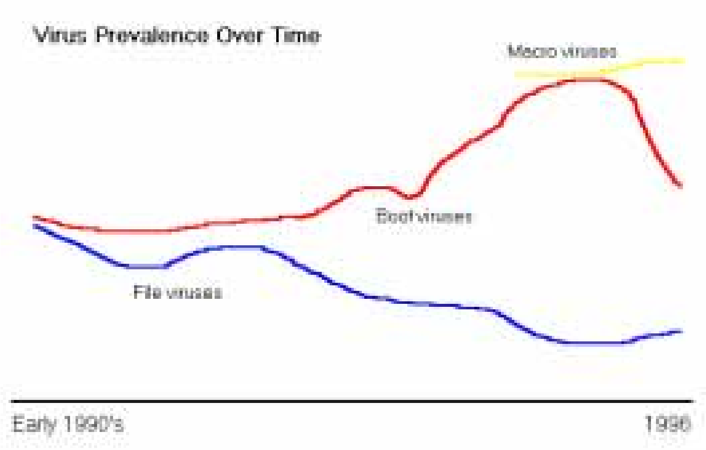
However, as soon as connectivity increased, mostly by the use of computers in the workplace, the boundaries of computer viruses widened. First there was the local area network (LAN), then there was the wide area network (WAN), and now there is the Internet. The extensive use of e-mail has also contributed to the meteoric rise in the number of macro virus incidents.

We are now living in a society in which global technology has taken the forefront, and global commerce is driven by communication pathways. Computers are an integral part of this technology and so the information they contain (as well as the malicious code they unwittingly contain) also becomes global.

Consequently, it is much easier to get a virus today than it was two years ago. However, the **types** of viruses that are common today are different than those that were common two years ago.

Steve White, Jeff Kephart, and David Chess of the IBM Thomas J. Watson Research Center have been following the evolution of viruses, and (among other things) they have concluded that the prevalence of certain types of viruses have been in part determined by changes in operating systems.[1]

Briefly, the trend has been as follows:



In the following sections, we will discuss how viruses operate in different operating systems.

[1] Steve R White, Jeffrey O Kephart and David M Chess, 'The Changing Ecology of Computer Viruses' *Proceedings of the Fifth International Virus Bulletin Conference*, Brighton, UK, 1996.

Copyright © 1999 Norman

Viruses on Different Operating Systems

When viruses first appeared, the only operating system of note was MS-DOS. It took some years for Windows to stabilize and become popular, so viruses flourished in MS-DOS. In fact, almost all file viruses (excluding macro viruses) are MS-DOS based.

Although OS/2 came on the scene shortly after viruses did, OS/2 is not as mainstream an operating system as DOS. Therefore, virus writers were — and still are — less likely to be running OS/2 themselves, and even if OS/2 viruses were written frequently, they would not be as widespread as MS-DOS viruses. As a result, there are only two known OS/2 viruses today.

Both Windows 95/98 and Windows NT are becoming increasingly common, and both are backwards compatible with MS-DOS, which means that they are backwards compatible with MS-DOS viruses. However, the architecture of the new operating systems pose interesting challenges to viruses.

Let's take a look at viruses on MS-DOS, Windows, OS/2, Windows 95/98, and Windows NT.

MS-DOS

Since the macro viruses that we have seen to date infect data files generated from and read by Windows applications, macro viruses are not a problem on MS-DOS-only machines.

Traditional file viruses and boot viruses prosper in MS-DOS machines because MS-DOS has no inherent security features. Viruses, therefore, have free rein to infect memory, and program files as described in “File Virus” on page 5.

Windows

When Windows was introduced, users had to change the way they interacted with their computers. The images on the screen were more colorful, navigating around in a program was more intuitive, and the prospect of being able to switch tasks without exiting an application was very enticing.

Since DOS runs “underneath” Windows, file viruses are able to infect machines that run Windows, but their lifespans are cut short. In general, file viruses are able to infect Windows executables, but the executables then do not generally work properly. Impatient users would either replace the executables, or if they were frustrated enough, reinstall Windows. This was enough to cause the demise of the traditional file virus. In addition, the structure of the executables in Windows is more complicated than Windows 95/98 executables, and memory has better protection. Viruses under Windows therefore never became the nuisance as they have proved to be under Windows 95/98.

Macro viruses and boot viruses, however, have not suffered the same fate. To date, macro viruses have been written to target Windows applications, and therefore the presence of Windows is required. Combining the wide acceptance of Windows with the fact that macro viruses infect data files rather than program files (see “Macro Virus” on page 10) has led to one macro virus, Macro.Word.Concept, becoming one of the ten most common viruses.

The actual booting process on a Windows machine is no different than on a DOS-only machine. Therefore, boot viruses have not been hindered by Windows, and they continue to propagate by infecting hard drives, going memory resident, and then infecting floppy disks.

OS/2

As mentioned above, OS/2 is not as widely used as Windows and other Microsoft operating systems. Because of the way that OS/2 was designed, however, it is still susceptible to non-OS/2-specific viruses.

Unlike Windows, MS-DOS does not run “underneath” OS/2. OS/2 is a powerful 32-bit operating system that supports DOS applications, Windows applications, and native OS/2 applications. In order to run DOS applications, OS/2 furnishes VDMs (virtual DOS machines). As the name suggests, VDMs “look” like DOS to DOS programs. Therefore, an infected DOS program can infect other DOS program files within that VDM, but not DOS programs in other VDMs. The newly infected DOS program file can then continue infecting other program files which might be started in VDMs in the future. So the infection path continues.

If Windows applications which include macro programming languages are run on an OS/2 machine, then the OS/2 machine is equally as susceptible to macro viruses as a Windows machine.

Again, since the booting process is the same on IBM-compatible machines prior to the operating system being loaded, boot viruses can infect OS/2 machines. OS/2 handles diskettes differently than DOS and Windows so the likelihood that the boot virus will propagate after it has infected the hard drive is lower on an OS/2 machine than on a Windows or DOS-only machine. The risk involved is rather one of the boot virus’s action on the hard drive. If the boot virus was designed to have a payload, then we can expect it to be delivered, regardless of whether it was able to infect any floppies.

OS/2 supports two file systems: FAT (file allocation table) and HPFS (high performance file system), and you may use just one or both. HPFS is more advanced and stores information in different places, so you can expect serious effects on an HPFS system from a boot virus that expected to only see FAT.

Windows 95/98

Windows 95 was launched at a time when the Internet became public property. Today the world wide web is available for everybody, not just for the seasoned user. Even though the majority of PC users welcome the Internet, e-mail, and chatting programs (see also page 25), the other side of the picture is a huge playground for virus makers, sometimes referred to as the Internet terrorists. The widespread use of these facilities has contributed to manifold the propagation of viruses under Windows 95/98.

Unlike Windows and DOS, Windows 95/98 is marketed as having built-in security features. Unfortunately, such features are not robust enough to safeguard Windows 95/98 against viruses. In fact, the first virus written especially to target Windows 95 (the Boza virus) emerged late in 1995. Furthermore, Windows 95’s workgroup networking environment has no file-level protection and therefore can potentially lead to increases in virus spreading.

After the rather primitive Boza virus, the Windows 95/98 viruses have increased in numbers and complexity. Like in the DOS environment, the first viruses were amateurish. Then they have become more technically complex as the virus writers have gained experience. Some of the viruses under Windows 95/98 spread by active use of the network protocol. A temporary "climax" of complexity and destructive capacity was reached with the CIH virus in 1998 (see page 24).

Windows 95/98 shares many characteristics with OS/2 with respect to system architecture and interaction with viruses:

Like OS/2, Windows 95/98 is a 32-bit operating system that supports DOS applications, Windows applications, and native Windows 95/98 applications.

Similar to OS/2's VDMs, Windows 95/98 has VMs (virtual machines) — a System Virtual Machine with separate address spaces for Win32 applications and a shared address space for all Win16 applications; and separate virtual machines for individual DOS applications.

File viruses can easily spread on a Windows 95/98 machine because DOS program files' only limitation under Windows 95/98 is that they cannot write directly to the hard drive.

Each DOS VM takes on the characteristics of the system from the point at which the machine was started. Since Windows 95/98 first starts up by running the same programs as a DOS-only machine does, it is possible that an infected program running during the startup process could go on to infect other program files within that VM. In addition, if that infected program originated from the startup process, it would become active in all VMs that were started in the future. Although program files from one VM cannot infect program files in another VM, it is possible for an infected program file to be loaded into a separate VM in the future, thereby continuing the infection path.

The macro viruses that have been written to date target data files generated from and read by Win16 and Win32 applications that are frequently run on Windows 95/98. As a result, macro virus infections abound on Windows 95/98.

Since the Windows 95/98 boot process is the same as a DOS-only or Windows machine (up to a certain point), boot viruses are able to infect hard drives of Windows 95/98 machines. When Windows 95/98 loads, however, boot viruses are often disabled and not allowed to propagate. On the other hand, if the boot virus has a payload, it may deliver it without requiring the virus to replicate beforehand.

The CIH Virus

Until 26 April 1998 it was true that viruses could inflict serious damages on software, but not on hardware. On this particular day, the virus Win95.CIH.1003 struck for the first time. The victims had to replace the flash BIOS chip, and even (especially on laptops) the PC's motherboard. In the months to follow, the CIH virus has been reported in the wild from most parts of the world. It now exists in four different variants, always triggering on the 26th in a month.

The CIH virus infects executable files under Windows 95/98 in a very covert manner. For example, infected files do not increase their length. Normally, an unexpected change of file length for a binary file is a sure sign of virus activity.

The technical description of what actions the virus take when it triggers is beyond the scope of this book. Simply put, when the flash BIOS is reprogrammed by the CIH virus, the PC is lobotomized and forgets its internal language. When this happens, there is no cure but replace this piece of hardware. The virus can also overwrite the harddisk and render it useless.

The CIH virus is a reminder that virus writers often have detailed information on undocumented procedures deep down inside the operating systems. When they use their expertise to write almost bug-free viruses as nasty as CIH, the need for proper virus protection is a must.

It's common knowledge that many CIH victims were infected after downloading files from Internet gaming sites. Therefore, we think it's appropriate to remind all surfers on the web about the everyday perils out there.

However, we do recommend that you visit the home page of your virus protection supplier in order to download program updates. The days when you could get away with updating you virus protection software on a quarterly basis are definitely over.

Viruses and IRC

Internet Relay Chat (IRC) is a system that allows users to have real-time conversations over the Internet. IRC programs have become customary as well as rather advanced. It is possible to exchange files and automate certain routines, for example. Certain IRC client programs accept file transfers that enable other chatters to send scripts that become a part of the IRC client itself.

There have been a number of incidents where such scripts have proved to contain malicious scripts that, in effect, have taken over the victim's machine. *Script.ini*, sometimes called the Ananas virus, is the most widespread code of this kind, performing such tasks as sending copies of itself to other users, issuing embarrassing statements on behalf of the user, or setting up your machine as a public file server.

It's debatable whether scripts like *script.ini*, *DMSetup.exe*, or *DMSetup2.exe*, for example, should be classified as worms or viruses. In any case, they represent a security risk, and regular viruses can be distributed via IRC just like any other files.

The best prevention is to disable the "auto-get" function, and never accept unknown files or programs.

Viruses and HTML

HyperText Markup Language (HTML), the authoring language used to create documents on the world wide web cannot in itself contain viruses as HTML is used solely "To publish information for global distribution" (WC3's HTML 4.0 specification). The HTML language has no ability to write to disks.

However, HTML's <SCRIPT> tag enables script statements and/or script files to be included in an HTML file. This is how HTML pages may be a potential "host" for malicious scripts, including viruses.

There have been reports about some simple viruses that use HTML pages with Microsoft's VBScript to spread, and Windows Scripting Host (WSH) to execute the VBScript files. Computers running Windows 98 and the beta version of Windows NT 5.0 are vulnerable as WSH is installed with the operating systems. Windows 95 and NT 4.0 may be updated to include MSH.

Only users of Internet Explorer are exposed to VBScript viruses from a malicious web site. A user is not vulnerable if he/she does not change the default security settings in the browser, and does not agree to run the file when asked. Make sure that the settings in Internet Explorer require user confirmation before potentially dangerous content can be run.

Users of Netscape Navigator and Communicator are not vulnerable to this threat as these browsers do not run VBScript.

So far similar threats have not been reported for the Java script language. As security for this environment is tighter, such threats are not likely to occur.

So far this type of viruses has not been reported "In the Wild". In addition, the viruses themselves don't have the same potential for propagating as most other viruses, because users do not normally exchange HTML or VBS files. The viruses potential for propagating is mainly from malicious web servers.

Thus Norman does not look upon this new threat as particularly dangerous. The developers of the NVC program family is nevertheless monitoring the situation closely.

Windows NT

As discussed in the sections on OS/2 and Windows 95/98, Windows NT supports DOS applications, Windows applications, and native Windows NT applications. Like Windows 95/98, Windows NT is backwards compatible with DOS and Windows. Despite the fact that NT's security features are more robust than Windows 95/98's, file viruses can still infect and propagate within Windows NT. DOS applications run in separate VDMs (virtual DOS machines), and file viruses can function within the VDM. Some DOS file viruses might not work in the intended fashion under NT, but there is nothing about NT's security that prevents file viruses from infecting.

As with Windows 95/98, Windows NT supports applications that contain macro programming languages, making NT as vulnerable to macro viruses as Windows-only machines.

Because Windows NT machines boot the same way that DOS machines do (up to the point at which NT takes over), boot viruses are able to infect NT hard drives. However, when these boot viruses attempt to go memory resident, they will be stopped by NT and therefore be unable to infect floppies. In effect, this stops the infection path, but the user must still deal with any side effects that the boot virus may have on the system — destructive payloads or manhandling NT's boot area in such a way that NT refuses to load.

Copyright © 1999 Norman

Solutions to the Virus Problem

Establish Routines

Unless organizations and single-users have established internal routines for data handling, the chance for running a virus-free computing environment is not likely to succeed. We have seen that when strategies and routines for data handling are initiated at management level, the organization is less exposed to virus infections. And when they occur, routines make it easier to root out the infected files before they spread.

Anti-virus Solutions

When people think of anti-virus solutions, they normally think of scanners. Scanners are the most readily available type of anti-virus solution, but they are not the only type.

It’s perhaps best to think of anti-virus solutions in terms of:

- what is required to detect the virus
  - generic methods
  - specific methods

and

- when the virus is detected
  - prior to the attempted infection
  - after the infection

A virus can be detected using either generic methods or specific methods. Generic methods look for virus-like behavior rather than specific viruses. As a result, even new viruses can be detected, and there is little need for frequent updates to the tool that is being used. Because generic methods look for behavior rather than specific viruses, the name of the virus is normally not given. Instead users are simply warned that a virus is likely to be present. Some shy away from this method because it can give false alarms — either by detecting a virus that is not there or not detecting a virus that is there.

Examples of generic detection methods are:

- checksumming and integrity checking
- heuristics
- decoys
- behavior blocking

Specific methods, on the other hand, rely on having prior knowledge of the virus. In this case the tool is able to both detect that the virus is present as well as identify it. As a result, frequent updates to the tool are necessary. Most users like to know what they’re “up against” if a virus is found, and the best way to do that is to determine the exact nature of the beast. For this reason, many users prefer this method, but they do not ultimately appreciate how often the tool must be updated.

Examples of specific detection methods are:

- on-demand and scheduled scanning
- real-time scanning

We have established that macro viruses pose the biggest threat for most PC users. Norman has introduced the program Cat’s Claw (presently available for Windows 3.1x and Windows 95/98), which enables the user to certify macros. In short, legal and guaranteed virus-free macros are the only macros that will be accepted. When a Word or Excel file is being accessed, non-certified macros will be removed, or access to the file will be denied. If you use this feature, you are protected against unknown macros and potential infections.

An equally important consideration is when the virus is detected. All users would probably agree that the ideal situation would be to prevent viruses from continuing to infect, and the next most ideal would be to identify those areas that have already been infected.

Let's examine where the above-mentioned methods fall:

Method	**Detection discussion**
Check-summing and integrity checking	Both methods store information about (hopefully) uninfected files in a certain place. Checks against the current status of the files and the stored information are performed periodically. If any change is detected, then a warning is issued. This method provides after-the-fact detection.
Heuristics	This is a method of analyzing files and boot areas in a general sense to determine if the code appears virus-like. Heuristics perform after-the-fact detection.
Decoys	This is a method of lying in wait for viruses, allowing certain files to become infected if a virus is present. Decoys detect viruses as they are infecting and are helpful in raising the warning flag.
Behavior blocking	This is a method of analyzing the behavior of all computing actions to determine if the sum of the parts adds up to a virus-like action. If so, then the action is stopped before infection can occur. Behavior blocking performs before-the-fact detection.

Copyright © 1999 Norman

Method	Detection discussion
On-demand and scheduled scanning	This is a method of scanning for specific viruses at certain times. This is always an after-the-fact detection.
Real-time scanning	This method also uses scanning, but the detection process occurs while other computer processes occur, such as copying a file. As a result, users are notified of existing viruses before they can be triggered.

As you can see, there is no single solution that meets all your virus detection needs. We therefore suggest that you combine these methods to best suit your needs, and follow them up with removal routines.

The subject of removal is similarly complex. Some people have a narrow definition of virus removal — if you delete the file or format the hard drive, the virus is no longer there. Please note, however, that such drastic measures are not often necessary, and that it is much healthier to define removal as removing the virus code and leaving behind a usable file and/or boot area.

Copyright © 1999 Norman

Some of the detection methods listed above can also perform removal (as defined the "healthy" way):

Method	**Removal discussion**
Checksumming and integrity checking	Can remove viruses.
Heuristics	Sometimes can remove boot and macro viruses.
Decoys	Cannot remove viruses.
Behavior blocking	Can remove viruses from memory and boot viruses from floppies.
On-demand and scheduled scanning	Can remove viruses.
Real-time scanning	Can remove viruses.

Please read on for an overview of what Norman Virus Control products can do.

Norman Virus Control Products

The Product Range

Because Norman realizes that no single anti-virus detection method addresses all your needs, our anti-virus products provide a wide range of solutions that are available for DOS, Windows, Windows 95, Windows NT, OS/2, NetWare, and Groupware.

We know that the hardware and software combination in organizations is mostly heterogeneous, and therefore we have designed our anti-virus products to coexist on these platforms and to send virus alert messages over the network, both via SNMP traps and NetWare messaging.

NVC for DOS/Windows

- Smart behavior blocking
- Cat’s Claw
- Integrity checking of system areas
- Decoys
- On-demand and scheduled scanning
- Removal
- SNMP trap messaging
- NetWare messaging

NVC for Windows 95

- Smart behavior blocking
- Cat’s Claw

- Right-click scanner
- On-demand and scheduled scanning
- Removal
- SNMP trap messaging
- NetWare messaging

NVC for Windows NT

- Real-time, on-demand, and scheduled scanning
- NVC NT Service
- Right-click scanner
- Removal
- SNMP trap messaging
- NetWare messaging

NVC for OS/2

- Smart behavior blocking
- On-demand and scheduled scanning
- Removal
- SNMP trap messaging
- NetWare messaging

Cat’s Claw

- Automatic removal of macro and file viruses, as well as boot sector virus removal
- Real-time scanning
- Macro certification

NVC for NetWare

- On-demand, scheduled, and real-time scanning on NetWare file servers
- SNMP trap messaging
- NetWare messaging

NVC for Groupware

- On-demand and real-time scanning on Lotus Domino® servers.
- SNMP trap messaging
- Virus removal

For more information, please contact your nearest Norman dealer or Norman directly.

Copyright © 1999 Norman

Copyright © 1999 Norman

Index

Copyright © 1999 Norman

Copyright © 1999 Norman