86-10-30 The Future of Computer Viruses

Richard Ford

Payoff

The new generation of object-oriented operating systems is expected to fuel the development of increasingly destructive viruses. This article discusses the exposures to viruses that these new environments present and the types of viruses for which today's IS manager should prepare.

Introduction

Computer viruses are a fact of life for the Information Systems (IS) manager. Since 1986, when the first virus was written for the IBM PC, the number of known viruses and variants has skyrocketed; the current total is over 8,000.This chapter describes the virus-related threats that users will face in future.

The First Recorded Viruses

Although computer viruses are often associated with the burgeoning computing environment of the late 1980s and 1990s, self-replicating code has been around for far longer. In the mainframe environment, codes existed that could fork processes and, replicating wildly, undermine performance. These were the forerunners of today's computer viruses.

The Brain virus of 1986 is often identified erroneously as the first virus ever discovered. However, although Brain was the first PC virus, its discovery was preceded by that of a virus written for the Apple II in 1981 called Elk Cloner. Elk Cloner's payloads included inverting the screen and displaying a text message. In 1983, Dr. Fred Cohen developed self-replicating code in a series of VAX and UNIX experiments and, subsequently, Brain was discovered in 1986. Brain, like Elk Cloner, was a boot sector virus that spread from machine to machine through the boot sector of floppy disks. The Brain virus received a great deal of media attention, and the combination of growing public awareness and the 1988 publication of *Computer Viruses: A High Tech Disease* by R. Burger—the first published reference for virus source codes—fueled a sharp increase in virus development. Burger's book detailed the source code for the Vienna virus, which has many contemporary variants.

The Brain virus introduced the concept of stealth to the computing public. A stealth virus hides the changes that it makes to an infected system. For example, if a full-stealth virus infects the boot sector of a diskette, a reading of that boot sector on an infected system will return the original contents of the boot sector, not the virus code. Similarly, changes to the amount of free memory or to the length of infected files can be disguised. After the stealth virus, the next technical innovations in viruses were encryption and polymorphism.

The first encrypted virus, Cascade, was discovered by virus researchers in 1989. Decrypting an encrypted virus is simple. If a variable key has been used to encrypt the virus, a hex string can be used to identify the short decryption routine. Not long after the decryption of Cascade, the virus-writing fraternity realized that writing code that could produce many different decryption routines would render it impossible or, at best, impractical, to detect these viruses with a simple hex string. Thus, polymorphic viruses were born. Polymorphic viruses, plus add-on polymorphic modules that could be linked to ordinary, non-polymorphic virus code, were developed. The most famous of these was the Mutation Engine (MtE) written by the Dark Avenger.

Previous screen

Polymorphic viruses prompted the next level of virus scanner development, as vendors raced to catch the rapidly accelerating target that viruses had become. As recently as mid-1995, several products were still unable to detect MtE reliably. Over the last few years, virus designers have developed different methods of infecting files and have infected a wider range of targets.

Viruses Targeted to Windows Platforms

One of the most significant challenges that present and future virus designers face is the new generation of operating systems. Virus writers have not yet focused on Windows 95 and Windows NT. Although many of the existing DOS viruses can infect executables or boot sectors in these new operating systems, only one virus, Boza, has been discovered that explicitly targets these platforms. The Boza virus is unremarkable, however, and does not pose a serious threat to Windows 95 users.

The most significant threat to Windows 95 or Windows NT platforms currently are traditional boot sector viruses. If a Windows NT or Windows 95 system is booted from an infected diskette, the virus will infect the boot sector of the computer. Because neither of these operating systems accesses the fixed disk in the same manner as DOS, the virus will usually be unable to infect other diskettes and should spread no further. However, the virus may render the machine unbootable. Moreover, the virus can, in many cases, successfully execute its trigger routine. Thus, if the virus is configured to format the fixed disk at boot time on a particular date, that trigger routine will be executed. Windows operating systems' accommodation of ordinary MS-DOS executables also exposes the platforms to the existing crop of file-infecting DOS viruses.

It is expected that more virus writers will target Windows platforms soon, because many of the systems' inherent characteristics make them ideal hosts for malicious code. In a multitasking environment, it is much easier for programs to become resident. This fact may result in a variety of new strategies to subvert the systems. Furthermore, there will be more ways for a virus writer to hide a malicious code's presence in memory. Windows' functionality for sharing resources seamlessly around a company also places the system as risk, in that an application that a user accesses could be on the local fixed disk, on the fixed disk of a computer across the room, or on a machine linked across the Internet. This invisible networking could provide the perfect infection vector for a virus, allowing self-replicating code to spread across an organization rapidly.

Researchers estimate that these platforms have not been subjected to attack because youthful virus writers—the majority of virus writers are young enough never to have had a full-time job—do not yet have the resources to write viruses for high-end operating systems. The average home machine is still running Windows under MS-DOS. Additionally, programming tools are not readily available to general users. DOS contained a primitive assembler/disassembler(DEBUG) as a standard offering; however, the large software development kits required for Windows programming are comparatively expensive and difficult to use. Those programmers who do have the necessary skills and resources to write viable Windows NT and Windows 95 viruses tend to be experienced programmers who are not, statistically speaking, in the virus-designing business. For the time being, then, the platforms have been virtually virus-free. This will likely change when this new generation of operating system becomes the standard.

The Macro Virus

In the last quarter of 1995, the first macro virus was discovered in the wild. However, there was no virus in the wild that was capable of spreading by embedding itself in data files until the discovery of the WinWord.Concept virus.

Previous screen

The idea behind macro viruses is simple. Many application programs allow for the inclusion of certain macros within data files. As application software has developed, macro languages have become increasingly powerful, allowing programmers to edit, delete, and copy files, and even on occasion to invoke the DOS shell. One application package allows for at least 150 macro commands in each template. These commands allow the programs to perform functions without user action.

Although several different packages are susceptible to macro viruses, Microsoft Word has been the most affected to date. There are also known macro viruses for Excel and AmiPro.

The WinWord.Concept Virus

When a user selects a Microsoft Word document and loads it, either from within Word or by double-clicking on the document icon, Word examines the document type. A standard Word document does not contain any embedded macros and is loaded into the program, ready for editing. However, if the loaded file is a document template file, Word automatically loads any macros that are contained within it. Should the document template file contain a macro named AutoOpen, the contents of this macro are automatically executed. The WinWord.Concept virus exploits this ability by adding macros to existing documents. These macros, once installed, add additional functionality to menu-bar operations and allow the virus to infect other documents on the system. Notably, AutoOpen is not the only autoexecuting macro in the Word environment; there are several others that Concept's creator could have taken advantage of. The Concept virus does not contain a trigger routine, but the words “That's enough to prove my point” are contained within the code.

Concept is now a widespread virus. The reason for this is twofold. First, although users rarely share executables or boot from diskettes, their jobs usually involve the exchange of data files. In many cases, users have been taught that viruses are only a risk when they are swapping binaries or diskettes, and thus do not suspect a macro virus. Moreover, many companies' policy and procedure manuals do not address the risk posed by macro viruses. The second reason for Concept's rapid spread is that it was accidentally distributed on CD-ROM and over the Internet.

The Future of Macro Viruses

In a straw poll taken at a recent firewalls conference, a speaker asked the audience members to raise their hands if they had heard of the dangers posed by embedded macros. Less than half of the attendees responded. This is not surprising. Time after time, systems administrators report that they have never heard about macro viruses. More disturbing is that many of them admit that, although they have heard of the viruses, they have not taken any protective measures and have no idea whether they may have an infected environment.

Macro viruses are intimately involved with data files. Although the Concept virus does not intentionally damage files on the system, the potential exists for macro viruses to cause data loss or corruption. Since the discovery of Concept, other viruses and Trojan horses designed for Word have been discovered, including one that modifies documents by adding the string “And finally I would like to say: STOP ALL FRENCH NUCLEAR TESTING IN THE PACIFIC.”

The future of macro viruses will depend on how application vendors develop software products. Some applications are more resilient to the threat of macro viruses than others; however, resilience can often undermine functionality. For example, although creating macros that are portable across a suite of applications is a wise business decision for a Microsoft, it will allow a macro virus written for Microsoft Word to function and spread

under Microsoft Access. Thus, added functionality increases the population of computers that can be infected.

Antivirus Software

For years, security practitioners have predicted the death of the virus scanner. However, the scanner is still the most popular weapon in the fight against viruses. Predictions of the scanner's demise were based on the sheer number of new viruses that scanner vendors must process every month to keep their products current. Several of these viruses are polymorphic and require a great deal of analysis by the developer to ensure reliable detection. Thus, the work required to maintain a scanner continually increases.

Because scanning engines that are currently in use cannot keep up with the pace of virus development, most new products will be virus nonspecific in design. The latest crop of new products provide for the automatic detection and removal of viruses. In the last year, there has been a growing interest in studying existing viruses to create a definitive must-detect list for antivirus products. Among the most important new developments are products that remove boot sector infections at boot time. Because boot sector viruses are the most common, this software could conceivably reduce the amount of time required to detect and repair damaged boot sectors. Although this software has certain limitations (i.e., it is vulnerable to those viruses that target it as well as to those that happen to meet its trigger conditions at boot time), it can provide a quick and easy method for catching boot viruses early.

Fully Automated Response for in the Wild Viruses

Some practitioners believe that new developments must be taken a step further. In a paper presented at the 1995 Virus Bulletin Conference, Mike Lambert proposed a concept called Fully Automated Response for in the Wild viruses (FAR-ITW). Lambert's philosophy is that, where possible, virus incidents should be cleaned up automatically, with no user disruption. Furthermore, Lambert believes that antivirus software should automatically and centrally report incidents, and gather samples of infected files for later analysis. Although several products can do part of this job, no product as yet is capable of fulfilling the role. FAR-ITW would allow vendors to concentrate repair and identification on those viruses that are known to be spreading. Additionally, the user does not necessarily need to be informed of a form-infected disk, although logging and reporting of the incident is vital in terms of evaluating the threat.

The level of automated response required by a company will depend on the type of business that the company is in and on the company's established policies and procedures. The principal advantage of automated removal is that it will not disrupt the workplace. Concerns about false positives must be addressed by identifying viruses before removal and by automatically removing only those viruses that are not likely to have caused data damage and that can be removed with certainty. Of course, the memory and performance impact of such a product would have to be thoroughly assessed before companies would opt to buy it. Although FAR-ITW is not yet available, several products seem to be addressing the automation goal.

Viruses Transmitted Over the Internet

Corporate IS personnel exploring the new vistas offered by the Internet are finding themselves inundated with the amount of information that is presently available on Internet virus activity and security. The intentional distribution of viruses over the Internet is increasing as quickly as the new commercial opportunities it provides. Virus exchange has always been a standard practice of the computer virus subculture. Although some reports

indicate that exchange is not a significant factor in increasing the number of viruses in the wild, the development of the Internet and the increased accessibility within corporate environments necessitate a reevaluation of the threat posed to users from this source.

Any service that is used to obtain binary files can be a source of virus infection. Several different Internet services, and the risks for users who access them, are discussed in the following sections.

File Transfer Protocol

Before the advent of the World Wide Web, the most common way for users to obtain information from remote systems was through the File transfer protocol (FTP). There are a large number of File Transfer Protocol sites around the world, many of which allow users to access anonymously certain files that are stored on them. FTP access is a useful facility for functions as diverse as locating the lyrics to an old pop song to obtaining the latest NetWare drivers for OS/2. However, users must be aware that the files they obtain may contain viruses.

Because of the volatile nature of the Internet, users should be taught to pay particular attention to where they obtain files. Although the well-known and popular archive sites have a good track record for containing clean files, lapses occur. The Computer Emergency Response Team (CERT) Coordination Center has received confirmation that some copies of the source code for wuarchive FTP daemon have been modified by an intruder and contain a Trojan horse. Any site running this archive should take immediate action by installing version 2.3 or disabling the File Transfer Protocol daemon. In this case, the software was “trojanised” to allow an intruder to access systems that were running the software. It would have been just as easy for a virus to have been planted.

Netnews

Usenet, or Netnews, is a widely circulated collection of newsgroups similar to a bulletin board system (BBS). Netnews membership is comprised of the entire online community. Netnews newsgroups range from the bizarre (e.g., alt.swedishchef.bork.bork.bork) to the useful (e.g., comp.virus, alt.security) and encompass a vast array of hobbies and pastimes. Although Netnews is a text-only environment, several newsgroups contain binaries that can be executed. These binaries are converted to text format with a UNIX utility known as uuencode and are posted to groups. A reader can then download the text, decode it, and recreate the original binary.

Aside from the usual problems of importing binaries onto a trusted machine, binaries obtained from a newsgroup are of highly questionable origin. There is little to prevent one user from masquerading as another. Further, several of the newsgroups contain files that have little or no work-related value. For example, the alt.binaries hierarchy contains a newsgroup called alt.binaries.pictures.erotica, and it is from this group that many users have contracted their first Internet-transmitted virus.

The virus that was spread in the newsgroup is the KAOS4 virus. An infected copy of a program named “Sexotica” was posted to the group. Several users downloaded it; some took the precaution of scanning it. Unfortunately, the KAOS4 virus was not familiar to the antivirus community and therefore was passed by scanners as clean.

The KAOS4 virus is poorly written, and it was not long before infected users began to notice malfunctions on their systems. The virus was isolated shortly thereafter, and antivirus software updated to detect it. One problem with tracking down the source of the infection within companies was that many users were reluctant to tell the staff how they had become infected—or that they were infected—because of the questionable source. Consequently, many systems were not cleaned up as effectively as they might have been.

Previous screen

This incident highlights the need to educate users about the dangers posed by text-encoded binary files that are available within the Usenet environment. Companies should restrict newsgroup access to those that are required for business functions. Virus code is not confined to accidental postings: there is a newsgroup dedicated solely to posting virus code.

The World Wide Web

In terms of viruses and the Internet, the most disturbing trend is the exposure of the World Wide Web (WWW) to infestation. Although viruses have been available through anonymous FTP for a long time, sites have been relatively hard to locate. With the advent of the WWW, it is possible to search out sites that provide large collections of viruses simply by using a Web search engine. Most of these virus collections are clearly labeled as such, so it is not likely that a user will unknowingly download an infected file. However, working with viruses is tricky, and it is easy for a user's experiment to get out of control.

Some users may feel that this availability of viruses will allow them to test antivirus software easily and thus perform private evaluations. This is not the case, however, as the maintenance of a virus collection can be extremely difficult. Using this method to test antivirus software is unwise for a variety of reasons. First, the majority of the virus threat comes from those viruses already known to be in the wild, rather than from so-called “zoo” viruses. Although there are over 8,000 different viruses written for the IBM PC, less than 200 have actually been encountered in the wild. Thus, the only way to test effectively the efficacy of a virus scanner is to create a collection of those viruses known to be in the wild. A collection obtained from a WWW site is unlikely to contain all of these viruses.

Another argument against testing antiviral software against virus collections is that the WWW collections contain many files that are not viruses, including text files renamed to COM extensions, damaged executables, and clean files. The only way to be sure that the files are real viruses is to replicate them onto clean files, known as goat files, and then to ensure that these files are also capable of replication. This is a very large job; if it is not carried out, the test results are meaningless at best and misleading at worst. Furthermore, many collections contain droppers or first generation samples of viruses. These are not suitable for testing scanners, because they are not the same as all further replications of that virus. Finally, the most common viruses are boot sector viruses; these are usually not found in such collections.

The correct way to test polymorphic viruses is to produce a large number of different samples. For a meaningful test of a product's polymorphic virus detection capability, thousands of samples must be replicated out, and samples that are missed must be confirmed as genuine replicants.

Although the ability of users to obtain viruses intentionally from the WWW is cause for concern, a far bigger problem is posed by the unintentional retrieval of viruses from the Web. Many corporate users are unaware of the risks to which they are exposed when browsing the Web. For example, one user searched for information on the Word macro virus, Concept. Finding a site, the user downloaded a document file that purported to contain more details. This document was infected with another Word macro virus. Consequently, when the browser passed control to the helper application, the virus code was executed and the local system became infected. The virus spread rapidly throughout the company.

In most cases, users are unaware of the risks to which they expose an organization's systems. There are several ways to address this problem. Two of the most effective strategies are comprised of technologically based solutions and denial-of-service solutions.

Previous screen

Technologically Based Solutions.

The simplest way to provide virus protection for binaries received over the Internet is to install some form of continuous virus protection on the host machine. Terminate and Stay Resident programs (TSRs)or, for Windows 95, VxDs, can provide effective protection against these viruses in the wild. However, many memory-resident applications do not have the same detection capabilities as their nonresident counterparts. Furthermore, users should ensure that TSRs are capable of detecting macro-based viruses. A TSRs should detect macro viruses even if the host application is launched by another program, such as a WWW browser. A company's antivirus software vendor will typically be able to supply this information. Publications such as *SECURE Computing* and *Virus Bulletin*are also reliable resources.

The threat posed by malicious electronic mail attachments can be limited at the mail gateway. Several vendors have developed products that attempt to scan incoming mail messages for binary file attachments before the messages are distributed to the eventual recipients. However, this is not a fool-proof solution, because encrytped information that is transmitted across the Internet cannot be scanned by the gateway. Although it would be possible to supply the gateway with a copy of all of the encryption keys, this is an unacceptable solution.

Finally, there has been some discussion about the possibility of checking incoming files for viruses at a company's firewall. This is a relatively new idea, however, and technology has not as yet been developed to support it.

Denial-of-Service Solutions.

A number of denial-of-service solutions have proven to be cost-effective security strategies. The idea of denying users access to particular services for security reasons has become unfashionable in recent years. Security as facilitator, not moderator, is the philosophy that has allowed security to become an accepted part of the IS manager's function. However, companies interested in accessing the Internet should, ideally, maintain a sensible balance between moderating and facilitating access to this powerful tool. In particular, the requirements for Internet access within a company should be closely examined. IS managers should evaluate how many users will need access over and above that required for E-mail. Most likely, the majority of the staff will not have a legitimate business need for unrestricted access. Further, for those users who require WWW access, the IS manager should consider providing isolated terminals that can connect to the Web. This approach has a number of benefits. With the use of limited terminals, anyone who wishes to use the WWW may do so, and the company will not be divided into those who have access and those who do not. Moreover, there will be a reduced opportunity for employees to waste time surfing the Internet. This approach does not preclude the installation of a wider net access if it is deemed necessary at a later date; it does, however, provide the benefits of the Internet within a controlled environment.

Firewalls can also be configured to limit the type of services that users are permitted to use. This is especially helpful if many users in a company require Internet access. For example, few Usenet groups will be used for business purposes. A full newsfeed is costly in terms of resources (e.g., disk space and bandwidth), lost time and productivity. Thus, companies should carefully prune the Newsnet hierarchy and, as a general rule, should set the default settings to deny particular newsgroups.

The Modern Day Virus Exchange and Its Writers

The ready availability of virus code on the WWW, both as precompiled binaries and as source code, has opened up the once-shadowy world of virus writers to the light of day. Modern day virus exchange is a simple matter of point and click. In many ways, this has

Previous screen

had the effect of legitimizing the activities of virus-writing groups. The moderated comp.virus Usenet discussion group has been, to a certain extent, supplanted by the anarchic but lively alt.comp.virus group, which is populated by a curious mix of virus writers, virus researchers, antivirus product developers, and users. Whereas at one time virus writers shielded themselves from the public eye, they are now happy to be at the forefront of discussion. Further exacerbating the situation has been the publication of a CD-ROM filled with live viruses collected by Mark Ludwig, and the public position adopted by several experts who preach that viruses are useful and valid research.

Research has shown that the general profile of virus writers has been affected by the changing computing environment and high visibility of virus design and protection. Over the past two years, research conducted by Sarah Gordon has yielded some interesting results. Although, in 1994, virus writers generally fell within the ethical norms for their age groups, recent writers appear to be desensitized to the negative effects of the viruses that they write. When confronted with the facts, today's crop of writers seem not to care; in fact, they seem to thrive on the fame and attention that virus writers are receiving.

Object Orientation

As operating systems increase in complexity, users are moving away from programs that relate to particular data and towards an integrated approach in which the borderlines between programs and data files are no longer clearly defined. This trend will create problems for the antivirus industry.

Already, the number of different objects that can be infected has grown immeasurably. Viruses currently exist that infect source code, bin files, SYS files, BAT files, OBJ files, and DOC files. Zhengxi, the latest virus that cannot be completely dissected and for which disinfection routines are difficult, is an example of these potent polymorphic viruses. Zhengxi infects EXE and OBJ files and attaches infected COM droppers to ZIP, ARJ, and RAR archives.

Windows Object Linking and Embedding provision, which allows users to plug executable code into many different objects, creates an environment in which a virus scanner must search through all Windows objects, regardless of their extensions, to find all of the OLE executables within a file and scan them. This results in slower scan times. Currently, no virus has been reported to target OLE. One company, however, was repeatedly reinfected by an infected executable OLE'd into a data file. Although the virus scanner was capable of detecting all of the subsequent replications of the virus, it could not detect the infected file within the document. One possible way to avoid this scenario is to use real-time virus protection in the form of Terminate and Stay Resident or VxDs.

Conclusion

It is difficult to make predictions concerning the future of viruses. The last decade of virus-writing has demonstrated that virus writers are extremely resourceful. The list of objects that may contain virus code continues to increase, and the development of increasingly object-based operating systems and environments will fuel this trend. The only safe predictions are that the number and complexities of viruses will continue to increase, and that viruses will be designed that target Microsoft's new high-end operating systems.

Author Biographies

Richard Ford

Dr. Richard Ford is Technical Director of Command Software Systems, Inc., located in Jupiter, Florida. Dr. Ford is an industry-wide recognized authority on computer viruses and antivirus products.

Robert Morrisâ€™s worm infected 10% of computers online at the timeâ€“around 6,000 machines. Morris built the virus to test the size of the Internet, when he was a grad student at Cornell. The bug slowed infected computers to a halt, prompting the government to sue. Under the Computer Fraud and Abuse Law, in December 1990, Morris became the first virus-maker convicted in U.S. court.Â By the time the 30-year-old was arrested, Melissa was the worst computer virus outbreak to date. The first ever email-aware virus hid inside an attachment called â€œList.DOC,â€ which contained a list of 80 passwords to porn sites.Â Taylor Beck writes about brains & the future for Fast Company. In his past jobs he studied memory, dreaming, and Japanese in neuroscience labs from Kyoto to St. More. A computer virus is a type of computer program that, when executed, replicates itself by modifying other computer programs and inserting its own code. If this replication succeeds, the affected areas are then said to be "infected" with a computer virus. Computer viruses generally require a host program. The virus writes its own code into the host program. When the program runs, the written virus program is executed first, causing infection and damage. A worm does not need a host program, as it is an You know computer viruses are bad news, but do you know where they come from? Join AVG in uncovering the true history of viruses and malware.Â It was crafted as a prank by future entrepreneur Rich Skrenta as a 15-year-old high schooler, and all you really had to do was reboot the computer to continue using it as normal. Although apparently he had a habit of doing these kinds of things, as his friends soon learned to stop trading floppy disks with him. A Pathology of Computer Viruses. SPRINGER-VERLAG London . Berlin .Â DOI: 10.1007/978-1-4471-1774-2. British Library, Cataloguing in Publication Data Ferbache, David 1965--A pathology of computer viruses. 1. Computer. Viruses 1. Title 004. Library of Congress Cataloging-in-Publication Data Ferbrache, David, 1965--A pathology of computer viruses / David Ferbrache p.cm. Inc1udes index. 1. Computer viruses. 1. Title. QA76.76.C68F451991 005.9--dc20.