Fighting Computer Viruses

Jeffrey Kephart, Gregory Sorkin, David Chess, Steve White

Scientific American
November 1997

Computer viruses have pervaded popular culture at least as successfully as they have the world's computer population. Capitalizing on the same fearful fascination with man-made life-forms that Mary Shelley tapped in Frankenstein, viruses have become the subject of widespread urban legends and hoaxes, popular television shows and movies. Yet they have not received much scientific scrutiny.

Much of their popular presence is attributable to an obvious but deep biological analogy: computer viruses replicate by attaching themselves to a host (a program or computer instead of a biological cell) and co-opting the host's resources to make copies of themselves. Symptoms can range from unpleasant to fatal. Computer viruses spread from program to program and computer to computer, much as biological viruses spread within individuals and among individual members of a society. There are other computer pathogens, such as the "worms" that occasionally afflict networks and the "Trojan horses" that put a deceptively friendly face on malicious programs, but viruses are the most common computer ill by far.

We and our colleagues at the IBM Thomas J. Watson Research Center have found the biological analogy to be helpful in understanding the propagation of computer viruses on a global scale and inspirational in our development of defenses against them. Building on decades of research by mathematical epidemiologists, we have obtained some understanding of the factors that govern how quickly viruses spread. Our efforts to find efficient methods of detecting viruses and the relations among them owe much to pattern-matching techniques developed by computational biologists. Furthermore, we have also drawn inspiration for defenses against pathological software from the vertebrate immune system and its astounding ability to repel or destroy pathogens.

Computer viruses can trace their pedigree to John von Neumann's studies of self-replicating mathematical automata in the 1940s. Although the idea of programs that could infect computers dates to the 1970s, the first well-documented case of a computer virus spreading "in the wild" occurred in October 1987, when a code snippet known as the "Brain" virus appeared on several dozen diskettes at the University of Delaware. Today viruses afflict at least a million computers every year. Users spend several hundred million dollars annually on antivirus products and services, and this figure is growing rapidly.

Most viruses attack personal computers (PCs). More than 10,000 viruses have appeared so far, and unscrupulous programmers generate roughly another six every day. Fortunately, only a handful have been detected far afield. There are three main classes of PC viruses (and the categories for other systems are analogous): file infectors, boot-sector viruses and macro viruses. Roughly 85 percent of all known viruses infect files containing applications such as spreadsheet programs or games. When a user runs an infected application, the virus code executes first and installs itself independently in the computer's memory so that it can copy itself into subsequent applications that the user runs. Once in place, the virus returns control to the infected application; the user remains unaware of its existence. Eventually a tainted program will make its way to another computer via a shared diskette or network, and the infection cycle will begin anew.

Boot-sector viruses, which account for about 5 percent of known PC virus strains, reside in a special part of a diskette or hard disk that is read into memory and executed when a computer first starts. The boot sector normally contains the program code for loading the rest of a computer's operating system (hence the name, a reference to lifting oneself up by one's own bootstraps). Once loaded, a boot-sector virus can infect any diskette that is placed in the drive. It also infects the hard disk, so that the virus will be loaded into memory whenever the system is restarted. Boot viruses are highly effective: even though there are fewer strains, they were for a time much more prevalent than file infectors were.

The third category, macro viruses, are independent of operating systems and infect files that are usually regarded as data rather than as programs. Many spreadsheet, database and word-processing programs can execute scripts--prescribed sequences of actions--embedded in a document. Such scripts, or macros, are used to automate actions ranging from typing long words to carrying out complicated sequences of calculations. And virus writers have created scripts that insert copies of themselves in other documents. Macro virusescan spread much more rapidly than other kinds of viruses because many people share "data" files freely--consider several workers swapping drafts of a jointly written report. "Concept," the first macro virus observed in the wild, infected its first Microsoft Word document late in 1995 and is now the most prevalent virus in the world. Today more than 1,000 macro viruses are known. As well as basic replication code, viruses can contain whatever other code the author chooses. Some virus payloads may simply print a message or display an image, but others will damage programs and data. Even those without malicious payloads can cause damage to systems whose configuration differs from what the virus designer expected. For instance, the "Form" virus, which usually produces only a slight clicking noise once a month, overwrites one disk directory sector in a way that is harmless to older PCs but lethal to newer ones that arrange disk information differently.

Antivirus Technology

Antivirus software has existed since shortly after computer viruses first appeared. Generic virus-detection programs can monitor a computer system for viruslike behavior (such as modification of certain crucial files or parts of main memory), and they can periodically check programs for suspicious modifications. Such software can even detect hitherto unknown viruses, but it can also be prone to false alarms because some legitimate activities resemble viruses at work.

Scanning programs, in contrast, can search files, boot records and memory for specific patterns of bytes indicative of known viruses. To stay current, they must be updated when new viral strains arise, but they only rarely raise false alarms. The viral signatures these programs recognize are quite short: typically 16 to 30 bytes out of the several thousand that make up a complete virus. (Similarly, biological immune receptors bind to sequences of eight to 15 amino acids out of the thousands in a viral protein.) It is more efficient to recognize a small fragment than to verify the presence of an entire virus, and a single signature may be common to many different viruses. Most computer-virus scanners use pattern-matching algorithms that can scan for many different signatures at the same time: the best can check for 10,000 signatures in 10,000 programs in under 10 minutes.

Once a virus has been detected, it must be removed. One brutal but effective technique is simply to erase the infected program, much as certain types of immune cells destroy an infected cell. Body cells are generally easy to replace, but computer programs and documents are not so expendable. As a result, antivirus programs do their best to repair infected files rather than destroy them. (They are aided in this endeavor by the fact that computer viruses must preserve their host program essentially intact to remain undetected and multiply.)

If a virus-specific scanning program detects an infected file, it can usually follow a detailed prescription, supplied by its programmers, for deleting viral code and reassembling a working copy of the original. There are also generic disinfection techniques that work equally well for known and unknown viruses. One method we developed gathers a mathematical fingerprint for each program on the system. If a program subsequently becomes infected, our method can reconstitute a copy of the original.

Virus-specific detection and removal techniques require detailed analysis of each new virus as it is discovered. Experts must identify unusual sequences of instructions that appear in the viral code but not in conventional programs -- a process that relies on carefully developed knowledge and intuition. They also must develop a prescription for verifying and removing the virus from any infected host. To keep up with the influx of half a dozen new viruses a day, antivirus technologists have developed automated tools and procedures to assist human virus experts or even replace them.

We have developed a brute-force statistical technique to extract high-quality signatures very quickly. We started by measuring the frequencies of short byte sequences in a large group of legitimate programs. When a new virus is sent to us, our software finds the sequence of viral bytes that is statistically least likely to appear in a legitimate program. This method is much faster than analysis by hand, and tests suggest that it produces signatures that are less prone to false alarms than those selected by expert humans. Our signature-extraction method is somewhat analogous to the outmoded "template" theory of the immune system, according to which antibodies mold themselves to a particular foreign invader -- our signatures are made specifically for each new virus we encounter.

Stephanie Forrest of the University of New Mexico and her collaborators at Los Alamos National Laboratory have developed an alternative that is more faithful to the currently accepted "clonal selection" theory of the immune system, in which the body generates an enormous range of immune cells and then mass-produces the ones that turn out to recognize a pathogen. Their scheme generates code signatures randomly, without reference to any particular virus. Each signature is checked against existing code on the system; if it does not match anything, it is retained in a huge database. Finding one of these signatures in a program is a sure sign that the program has been modified, although further analysis is required to determine whether a virus is at fault.

In another twist on the biological metaphor, virus hunters have learned to exploit the fact that programmers often make new computer viruses from key parts of existing ones. These viral "genes" enable us to trace the evolutionary history of computer viruses, in the same way that biologists determine the family trees of related species. By processing large collections of viral code, we can automatically derive a set of family signatures that catches all the different members of a viral family, including previously unknown variants. This technique reduces signature storage requirements substantially: a single 20-byte family signature can recognize dozens of distinct viruses.

We have also developed a neural-network technique to recognize viruses by scanning for several, very short patterns, each only three to five bytes long. These tiny fragments represent computer instructions that carry out tasks specific to viral infection. Although conventional software might occasionally contain one of these fragments, the presence of many of them is an almost certain viral hallmark. Antiviral software can check for such short sequences very quickly; even more important, because these patterns of data are directly linked to the virus's function, we can now recognize a wide variety of viruses without ever having seen them before.

Hunting Viruses in the Wild

Since 1990 we have been collecting virus statistics from a population of several hundred thousand PCs among our corporate customers. We record the location and date of each incident along with the number of infected PCs and diskettes and the identity of the virus. These statistics have permitted us to infer a good deal about the behavior of viruses in the wild, including the fact that only a small fraction of viruses are genuinely problematic. Only about 5 percent of all known viruses have been observed within the population we have studied, many of them just once. The 10 most common viruses account for two thirds of all incidents. In addition, the prevalence of these successful viruses appears to follow a common pattern: a virus will spread over the course of a year or so, increasing its numbers in a roughly linear fashion until it reaches a plateau. After that, it will continue to appear in computers at a roughly constant level, although sometimes its numbers decline to near extinction.

In an effort to understand these characteristics, we have borrowed from mathematical models of biological epidemics. The simplest models predict the behavior of a disease from a few parameters--most significantly, the "birth rate" at which sick individuals infect others and the "death rate" at which the sick either die or are cured. If the ratio between these two rates is less than a critical value, any infection will quickly die out. The larger the ratio, the more likely an epidemic, and (if there is no immunity) the greater the fraction of the population that will be infected at any one time.

Our observations suggest that such a simplistic view is inadequate. Unless the ratio of the birth and death rates just happens to be close to the critical value, a virus should either die out completely or spread exponentially and become almost universal. Instead many viruses persist steadily at levels that are a small fraction of the overall population. One crucial error in this simple model appears to be in assuming uniform chances of contact among everyone in the population at risk. More sophisticated models take into account the extraordinary cliquishness of typical patterns of software exchange. Each person shares software and data only with a few other people, on average, and most of the sharing takes place within groups. If Alice shares with Bob and Bob shares with Carol, then Alice and Carol are reasonably likely to share with each other.

Computer simulations have shown that locality of contact slows the initial growth in a way that is qualitatively consistent with our observations. Sparse sharing reduces the likelihood of an epidemic and lowers the plateau, but not by enough to explain the data.

Evolution in Action

Just as external factors such as drought, sanitation and migration have a strong influence on biological epidemics, changes in the computing environment are responsible for the presence of several distinct epochs in viral infection. Until 1992, reported sightings of file-infecting viruses and boot viruses occurred at roughly equal (and steadily rising) rates. Then the incidence rate for file infectors began to fall dramatically, whereas that for boot-sector infectors continued to rise. Between late 1992 and late 1995, boot-sector infectors reigned supreme. Why did the file infectors essentially become extinct?

We believe the cause was the widespread acceptance of Windows 3.1, an enhancement to MS-DOS -- the operating system used on most computers -- that became popular around 1992. Windows crashes readily in the presence of typical file viruses, and so necessity will lead afflicted users somehow to eliminate the virus from their systems (perhaps by wiping out the hard disk and reinstalling all the software), regardless of whether they know that the symptoms are caused by a virus. Boot viruses, in contrast, tend to coexist peacefully with Windows 3.1; they do not kill their hosts before the infection has a chance to run riot.

The wide use of Windows 95, yet another new operating system, has now led to a precipitous decline in the prevalence of boot viruses. Windows 95 warns the user about most changes to boot sectors, including many of those caused by viruses, and most boot viruses cannot spread under Windows 95. We have already seen a handful of viruses specifically designed for Windows 95 and other 32-bit operating systems, although the ones we have seen are unlikely to become widespread.

We are now in the era of the macro virus. Because users tend to exchange documents and other data files capable of harboring macro viruses more frequently than they exchange programs, macro viruses enjoy a higher birth rate and thus spread faster than the traditional boot or file infectors. Sophisticated mail and file-transfer functions now permit users to share documents or programs more quickly and easily than before, exacerbating the problem.

Macro viruses are also the first viruses to exploit the growing trend for interoperability among computers. A DOS file infector can never endanger a Macintosh, for instance, but a macro virus can infect any computer that supports a vulnerable application program. The fact that Microsoft Word runs on many different kinds of computers enables Concept and other macro viruses to move beyond traditional system boundaries.

A Digital Immune System

Today viruses mainly travel from one computer to another through intentional, manual exchange of programs, and human response time is generally sufficient to cope with them. A successful new virus typically takes months or even years to gain a foothold. In the densely connected world of the near future, viruses might be able to propagate much faster. As early as 1988, Robert Tappan Morris launched what came to be known as the "Internet Worm," a program that exploited security holes and invaded hundreds of computers around the world in less than a day.

New technologies (such as Web browsers that use "ActiveX") for silently downloading software and data to a user's computer make the problem even more pressing. Already modern-day mail programs permit text documents or spreadsheets to be sent very simply as e-mail attachments. Opening the attachment can cause the appropriate application to start up automatically, and any macro viruses contained in the attachment may be executed. Soon software agents may be routinely authorized to send and open mail containing attachments. With humans no longer participating in the replication cycle, viruses could be free to spread orders of magnitude faster than they do now.

These changes in the digital ecosystem suggest that a more automatic response to computer viruses is needed, one that is not limited by human response times or by the rate at which humans can dissect novel viruses. IBM, Symantec Corporation and McAfee Associates are among the companies developing technology to help respond quickly and automatically to new viruses.

At IBM, we are creating what may be thought of as an immune system for cyberspace. Just as the vertebrate immune system creates immune cells capable of fighting new pathogens within a few days of exposure, a computer immune system derives prescriptions for recognizing and removing newly encountered computer viruses within minutes. In a current prototype, PCs running IBM AntiVirus are connected by a network to a central computer that analyzes viruses. A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signatures to infer that a virus may be present. The monitoring program makes a copy of any program thought to be infected and sends it over the network to the virus-analysis machine.

On receiving a putatively infected sample, the machine sends it to another computer that acts as a digital petri dish. Software on this test machine lures the virus into infecting specially designed "decoy" programs by executing, writing to, copying and otherwise manipulating the decoys. To replicate successfully, a virus must infect programs that are used often, and so the decoy activity brings the viral code out of hiding. Other behavioral characteristics of the virus can be inferred during this phase as well.

Any decoys that have been infected can now be analyzed by other components of the immune system, which will extract viral signatures and produce prescriptions for verifying and removing the virus. Typically it takes the virus analyzer less than five minutes to produce such prescriptions from an infected sample. The analysis machine sends this information back to the infected client PC, which incorporates it into a permanent database of cures for known viruses. The PC is then directed to locate and remove all instances of the virus, and it is permanently protected from subsequent encounters.

If the PC is connected to other machines on a local-area network, it is quite possible that the virus has invaded some of them as well. In our prototype, the new prescription is sent automatically to neighboring machines on the network, and each machine checks itself immediately. Because computer viruses can exploit the network to multiply quickly, it seems fitting that the antidote should use a similar strategy to spread to machines that need it. By allowing the latest prescriptions to be propagated to subscribers at uninfected sites, it is possible in principle to immunize the entire PC world against an emerging virus very rapidly.

Regardless of how sophisticated antivirus technology may become, computer viruses will forever remain in an uneasy coexistence with us and our computers. Individual strains will wax and wane, but as a whole, computer viruses and antivirus technology will coevolve much as biological parasites and hosts do. Both will also evolve in response to such changes in the computing environment as itinerant software agents--which will have to be protected from corruption by the computer systems they traverse even as those systems guard themselves from agent malice. Perhaps computer viruses and computer immune systems are merely precursors of an eventual rich ecosystem of artificial life-forms that will live, die, cooperate and prey on one another in cyberspace.

Further Reading

- ROGUE PROGRAMS: VIRUSES, WORMS AND TROJAN HORSES. Edited by Lance J. Hoffman. Van Nostrand Reinhold, 1990.
- COMPUTERS AND EPIDEMIOLOGY. J. O. Kephart, S. R. White and D. M. Chess in IEEE Spectrum, Vol. 30, No. 5, pages 20-173;26; May 1993.
- A SHORT COURSE ON COMPUTER VIRUSES. Second edition. Frederick B. Cohen. John Wiley & Sons, 1994.
- ROBERT SLADE'S GUIDE TO COMPUTER VIRUSES. Robert Slade. Springer-Verlag, 1994.
- BIOLOGICALLY INSPIRED DEFENSES AGAINST COMPUTER VIRUSES. Jeffrey O. Kephart, Gregory B. Sorkin, William C. Arnold, David M. Chess, Gerald J. Tesauro and Steve R. White in Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, August 20-173;25, 1995. Distributed by Morgan Kaufmann Publishers, Inc.
- A Biologically Inspired Immune System for Computers
- Computer Virus Handbook by David Stang of Quarter Deck
- The Crypt Newsletter