Advising users on computer systems technology

NCSL BULLETIN
August 1990

NCSL Bulletins are published by the National Computer Systems Laboratory (NCSL) of the National Institute of Standards and Technology (NIST). Each bulletin presents an in-depth discussion of a single topic of significant interest to the information systems community. Bulletins are issued on an as-needed basis and are available from NCSL Publications, National Institute of Standards and Technology, B151, Technology Building, Gaithersburg, MD 20899, telephone (301) 975-2821 or FTS 879-2821. If you wish to be placed on a mailing list to receive future bulletins, send your name, organization, and address to this office.

The following bulletins are available:

Data Encryption Standard, June 1990
Guidance to Federal Agencies on the Use of Trusted Systems

Technology, July 1990

Acknowledgements

John P. Wack, NIST, and Stanley A. Kurzban, IBM, are the principal authors of this bulletin. NIST appreciates the contribution of Mr. Kurzban, of IBM's System Research Education Center, whose paper, "Computer Viruses and Other Self-Copying Programs," is the basis for this bulletin. Mr. Kurzban's paper was based on an earlier, longer article [1].

This paper was developed in cooperation with the Computer and Telecommunications Security Council, a technical group sponsored by NIST to help identify and develop key issues affecting the computer security community in both government and industry. NIST acknowledges in particular the efforts of William J. Murray and Donn Parker, who contributed to earlier drafts of this paper.

COMPUTER VIRUS ATTACKS
by John P. Wack and Stanley A. Kurzban

INTRODUCTION

This paper discusses computer viruses and other related threats and their impact on computing. Much has already been written about computer viruses and how to deal with them; this paper attempts to highlight some of the issues surrounding the computer virus phenomenon that have been at times overshadowed by the extensive media attention and reactive mode forced upon many security professionals. The main issue that this paper brings forth is that despite current advice and controls, computer viruses may continue to be a serious problem. Larger, more complex issues regarding computer security may have to be addressed if the situation is to get better.

This paper is intended for management and those who need basic information about computer viruses and related threats. "Statement of the Problem" is a brief overview of computer viruses and related threats, dealing mainly with terminology. "What Can Be Done Now" describes current recommendations for preventing computer viruses. "Problem Extent and Future Impact" focuses on the extent to which viruses have affected computing and whether current recommendations for dealing with viruses will be adequate in the future.

STATEMENT OF THE PROBLEM

Computer viruses are program segments that copy versions of themselves into programs (targets) and thereby convert the targets into vehicles for further propagation. Viruses usually spread from program to program within a single system by thus reproducing every time any infected program runs. Moreover, they can spread from system to system whenever an infected program is introduced into another system.

As society now employs an increasing number of microcomputers to perform many complex and sensitive operations, it is interesting to note that all computer viruses found outside of controlled experiments have run only on microcomputers. Several factors may be relevant:

- Computer viruses are attractive vehicles of malice or profitless harm, and many malicious people may be presumed to have more access to microcomputers than larger systems.
- The uses for microcomputers have become more varied and sensitive, yet their system architectures (and absence of security mechanisms) have not changed appreciably. However, security measures have continued to evolve on larger systems, which are far more likely to be shared by many people.
- Users of microcomputers often view the need for measures that prevent or deter such things as viruses from program modification as unnecessary or inconvenient on single-user systems, whereas similar measures are usually embedded in larger systems.
- Microcomputers are often shared and serially reused in many environments without effective safeguards against viral infection.
- Sharing of program-containing media is far more common on microcomputers than on larger systems.

It should not be assumed that larger systems are immune to computer viruses and related threats. Rather, whatever actions a virus may effect on a microcomputer could be possible on a larger system.

The term, “computer virus,” has often been used imprecisely to refer to Trojan horses, worms, and logic bombs. More precise definitions of these terms are:

A Trojan horse [2] is a program that conceals harmful code. A Trojan horse usually resembles an attractive or useful program that a user would wish to execute.

A logic bomb [16] is code that checks for a certain set of conditions to be present. If these conditions are met, it may cause sudden and widespread damage. A time bomb is a logic bomb that is triggered by a certain date or time. Since a logic bomb is presumably something that a person would not wish to execute, it is likely to be concealed, that is, in a Trojan horse.

A worm [3] is a self-contained program that copies versions of itself across electronically connected nodes. The Internet worm [4] and the CHRISTMA EXEC [5] are two examples of worms, not computer viruses (however, the CHRISTMA EXEC required some user interaction to spread and possessed aspects of a Trojan horse).

A virus [7] is code that plants a version of itself in any program it can modify. The virus may append or otherwise attach itself such that the program executes after the virus code, making it appear as if the program were functioning as usual, or the virus may overwrite the program such that only the virus will function. A Trojan horse program could initiate the spread of a virus, as could a worm.

In addition to propagation mechanisms, viruses and worms have “missions,” for example, to cause harm via a logic bomb. Note that the existence of a mission does not necessarily connote harm [6]. In theory, it could be beneficial (the concept of a worm was introduced [3] in the context of a useful application). Since we deal here with “attacks,” however, we assume that every virus and every worm is harmful.

Note that the potential for harm from a computer virus is great because:

- Viruses can spread from program to program within systems and from system to system without limit. (Worms can do the latter as well.)
- It is extremely difficult, if not impossible, to trace a virus back to its creator, so fear of punishment is unlikely to restrain anyone who looses a virus.
- A virus or worm or Trojan horse can contain virtually any type of harmful code, and such code can be extremely difficult to identify in advance as harmful.
- The only foolproof defense against them is to use no software with any function that is not thoroughly understood by the user. However, that is not practical; use of programs whose working is not thoroughly understood is the very cornerstone of computers’ value.

WHAT CAN BE DONE NOW

The most important defense against any harm that software might do is prudent acquisition and use of software. Unfortunately, in the case of computer viruses, one cannot fault most victims, as they simply trusted that the software they used did not contain harmful code (unfortunately, victims bear the onus of having been sloppy with regard to security, which is not true in all cases). As a consequence, the threat of viruses now makes it prudent to avoid placing any degree of trust in software unless one has been reasonably careful in acquiring it from a reliable source. At the same time, viruses have been found in shrink-wrapped diskettes and the size or reputation of a software house is not a guarantee of protection.

Anti-viral software may be of significant assistance in verifying whether software is safe to use and preventing viruses and related threats from causing damage. There are three types [12] [13]:

1. Preventive software may prevent viruses from spreading within systems by placing barriers in the path of program modification. The barriers might be controls such as common access control or encryption of programs or unbypassable messages to users whenever program modification is attempted.
2. Detective software monitors events in a computer and reports suspect ones to the user. While program modification is one such event, so too are many others, like modification of system control blocks, that have been used by known viruses.
3. Virus-specific, or scanning software simply tests for the occurrence of signatures of known viruses (a signature being a string of bits known to occur in a virus and to be relatively unlikely to occur elsewhere by chance).

Some drawbacks associated with anti-viral software are (1) virus-specific software may fail to detect viruses more recent than the software and (2) detective software may fail to detect some viruses that are already resident in memory when the software is loaded. However, known viruses have survived for many years and have infected systems in very widely distributed systems; most damage that has been done to date has been done by viruses that had first been detected and studied long before the damage in question was done. Thus, anti-viral software can protect systems from this large body of known viruses, while at the same time providing a level of deterrence against newer viruses.

The next step in defense lies in protection against damage that might be caused by harmful code carried within viruses. In its most simple form this means regular backups of data; in more evolved forms it includes system maintenance, physical security, risk analysis, contingency plans, and teams of experts who can respond quickly to virus attacks. Depending on the environment, these measures can be employed in varying degrees, with backup being the most important.

What is clear about viruses is that they have not so much created new vulnerabilities as they have exposed and exploited long-standing ones, with the arguable exception being modification of programs on personal computers [14]. Thus, prudence in acquiring software combined with measures such as described here can provide a significant level of deterrence and thus protection from viruses and related threats.

PROBLEM EXTENT AND FUTURE IMPACT

While current measures for dealing with computer viruses have proven to be effective, one should not be left with the impression that the problem of viruses has been solved or that by using these measures one can eliminate viruses. Evidence shows that the numbers of new viruses and virus incidents are increasing each year and that viruses are becoming more sophisticated and malicious. Moreover, most victims continue to be hit by “older” viruses that are well understood and for which detectors and vaccines have been developed. Governmental and commercial organizations, academia, and users have all responded in some form to the problem [8], [9], [10], [11], however, specialized defenses are still in their infancy [15]. This situation, combined with the factors of increased dependence on computers and more variety and complexity in software (making its quality and trustability more difficult to ascertain), could result in well-orchestrated incidents of harmful software’s wreaking havoc.

An unfortunate aspect of computer viruses is that they cannot be assumed to have been eradicated until there are no suitable remaining “hosts” (computers) for the software to infect. The microcomputers in widespread use do not contain built-in security measures such as those on larger systems, thus the preventive measures for viruses and related threats depend on users’ willingness to purchase, install, and use them. Moreover, the process of educating users and helping them to be more aware of the problem is slow, and people are, as in other things, prone to lapses of good judgment where computer security is concerned. Consequently, there is no reason to assume that current defenses against viruses will be generally more effective in the future than they are at present.

Perhaps as part of the effort to develop better defense measures, we need to change our attitudes towards the use and abuse of computer systems. We need to understand that computer crime is no different from theft, damage to property, or fraud. We need to understand that viruses alone are not the problem, but rather the authors of viruses. These individuals have caused significant damage to systems and data, but they have possibly caused more damage to the fabric of society's trust in the usefulness of computing. If people are unable to use programs without fear that those programs may do more harm than good, then the entire foundation of useful computing is undermined. People who write viruses are doing a vast disservice to the computing profession; they are not computer professionals, hackers, or "whiz kids," they are at best criminals and vandals. We need to instill that message into the rest of society.

At the same time, computer users should continue to learn more about viruses and how to prevent them. We need to promote the use of current defense methods and measures as applicable, and support efforts for systems and measures that offer better security and integrity. We need to understand that the solution to the computer virus problem is complex, involving many issues that fall under the broad category of computer security. With this attitude, we need to continue our use of computing systems, given reasonable safeguards and protection, to accomplish purposeful and useful aims.

REFERENCES

[1] Kurzban, Stanley A., "Viruses and Worms—What Can You Do?," ACM SIG Security, Audit, & Control, Volume 7 Number 1, Spring 1989.

[2] Anderson, James P., "Computer Security Technology Planning Study," ESD-TR-73-51, Volumes I and II, USAF Electronic Systems Division, Bedford, Massachusetts, October 1972.

[3] Shoch, John F., and Jon A. Hupp, "The Worm Programs—Early Experience with a Distributed Computation," Communications of the ACM Volume 25, Number 3 (March 1982), Pages 172-180.

[4] Spafford, Eugene H., "The Internet Worm: An Analysis," Purdue Technical Report CSD-TR-823, November 28, 1988.

[5] McLellan, Vin, "Computer Systems Under Siege," The New York Times, January 17, 1988.

[6] Murray, William H., "Epidemiology Application to Computer Viruses," Computers and Security, Volume 7, Number 2, April 1988.

[7] Cohen, Fred, "Computer Viruses," Proceedings of the 7th DoD/NBS Computer Security Conference 1984, Pages 240-263.

[8] International Business Machines Corporation, "IBM Systems Security," Programming Announcement 289-581, October 24, 1989.

[9] White, Steve R., David M. Chess, and Chengi Jimmy Kuo, "Coping with Computer Viruses and Related Problems," Research Report Number RC 14405, International Business Machines Corporation, Yorktown Heights, New York, 1989; adapted and distributed as "Coping with Computer Viruses and Related Problems," Form G320-9913, International Business Machines Corporation, September 1989.

[10] National Computer Security Center, "Proceedings of the Virus Post-Mortem Meeting," Fort George G. Meade, Maryland, November 8, 1988.

[11] Wack, John P., and Lisa J. Carnahan, "Computer Viruses and Related Threats: A Management Guide," NIST Special Publication 500-166, National Institute of Standards and Technology, August, 1989.

[12] McAfee, John, "The Virus Cure," Datamation, Volume 35, Number 4, February 15, 1989, Pages 29ff.

[13] Marc Adler, "Infection Protection: Antivirus Software" PC Magazine, April 25, 1989, Pages 193ff.

[14] Denning, Peter J., "Computer Viruses," American Scientist, Volume 766 (May-June 1988), Pages 236-238.

[15] Pozzo, Maria M., and Terence E. Gray, "An Approach to Containing Computer Viruses," Computers and Security, Volume 6, Number 4, August 1987.

[16] Spafford, Eugene H., Kathleen A. Heaphy, and David J. Ferbrache, "Computer Viruses - Dealing with Electronic Vandalism and Programmed Threats," ADAPSO Software Industry Division Report, 1989, Page 8.