Southern Illinois University Carbondale
OpenSIUC

Honors Theses

University Honors Program

5-1989

Computer Viruses

Roger L. Miller
Southern Illinois University Carbondale

Computer Viruses

by

Roger L. Miller

Senior Honors Project

CS 492

Dr. Nicholas Phillips

May 5, 1989

Preface

Before I present this copy of my project, I would like to take a moment to talk about how the nature of the project changed from its inception to its completion. Originally I had planned to disassemble the Pakistani Virus and write a program to attack it. A bold venture to be sure but one I thought was within reach. At the urging of my advisor, Dr. Phillips, I altered the description to make it at least partially a survey of computer viruses, as reported in the media and other sources. I also decided to add the part about a small case study of the attack at SIU.

As it turns out, experience again saved the day. I ran into numerous difficulties trying to take apart the virus, much less writing a program to counter it. With the lack of time and resources, degree of difficulty (the Pakistani virus is reputed to be the most technically sophisticated virus in the world), and the normal rigors of a college semester, the task proved too much. So I fell back onto the survey part of my project.

There were also problems in this. The resources that were available to me were rather limited. I had trouble obtaining the more comprehensive and technical reports concerning viruses, even through the inter-library loan system. I decided near the project's completion to keep the non-technical because of difficulty in obtaining resources, the technical aspects are very case-specific, and the readibility for non computer scientists would have been significantly decreased.

Computer Viruses

In a 1959 paper, computer pioneer John von Neumann suggested that computer programs might actually multiply, taking on a life all their own.[1] As so often happens, what once appeared fanciful science fiction has become a harsh reality. Programs creating other programs is an area of intense research and it has been successfully implemented in a limited scope. Programs that do seemingly have a perverse life of their own have stepped into the limelight in the past year. National attention was focused on so called viruses when ARPAnet, a military and research communications network, was overwhelmed by a viral attack in November 1988. In this paper, I plan to discuss viruses and related problems, take a look at one virus attack in particular, and examine what the future holds for computer security.

Around 1969, three programmers at AT&T's Bell Laboratories, perhaps acting on von Neumann's theories, took them one step farther and implemented self-replicating code, that is code that will make a duplicate of itself. Further, using the fact that a byte is a byte, they realized that systems using the same primary or core memory for program and data storage, left programs vulnerable to being consumed (as data) by other programs, or even by themselves. With all of this in mind, they designed a "game" that would pit two self-replicating programs against each other like gladiators, with core memory being their arena. These programs would then "battle to the death" by duplicating themselves and erasing or consuming the opposing program. The winner was the program that had destroyed the other program or controlled the most memory at the end of the allotted time. Soon the game caught on at other research facilities and was dubbed "Core Wars".[2]

Its creators realized the damage that could be done by their "organisms" if they were allowed to run rampant. The actual code wasn't as troublesome as the theory. There was the fear that someone with malicious intent could loose a program and cause untold destruction of data. In reality the threat was small because a machine with code gone wild could easily shut down. At the time most machines stood alone but as connectivity and computer access grew, so did the danger. For the most part, Core Wars and the idea of battling destructive code was kept quiet . . .until 1983.

At an Association for Computing Machinery banquet, Ken Thompson, creator of the original version of UNIX, was being given an award. In his speech, he told of core wars and how to create organisms. "If you have never done this, I urge you to try it on your own."[3] In 1984, "Scientific American" followed with an article on Core Wars and offered guidelines for creating your own battlefields and organisms. Fred Cohen presented a paper, Viruses: Theory and Experiments, to a computer security conference in 1984.[4] Soon after the name, computer virus, caught on and so did the practice of creating and releasing them.

Occasionally stories of viral epidemics appeared in the press but for the most part the public was unaware of what could happen. In 1986 sporadic stories about viruses and their potential danger were printed but they were ignored or dismissed even by many professional in the field. On Wednesday, November 2, 1988 the outbreak that many had feared and some even predicted occurred.

At about 6pm Wednesday the infectious code (technically it was a worm) was first noticed at several computer centers connected by Internet and began attracting a great deal of attention a few hours later.[5] The worm was reproducing so rapidly, it slowed down what ever system it infected. Because of its crippling effects and sophistication many talented computer scientists were worried but intrigued by the worm. People all along Internet, which is connected to several premiere research networks such as BAR and ARPAnet, began to dissect the worm and work on a fix.[6] Graduate students, researchers and system operators along the network battled around the clock; by Friday night, the worm was under control and had nearly been eliminated, barely two days after it had been unleashed. It had no lasting effects except to raise a flag of warning about what could have happened had the worm not been benign. If not for a flaw in the code, the worm would replicated at a significantly slower rate and probably could have gone unnoticed for months. It's ironic that the creator, Robert T. Morris Jr., made his mistake when adding code to increase his worm's longevity in the network and avoid defenses aimed at it.[7] What is even more ironic is that Robert T. Morris Sr. was one of the programmers who came up with the concept of Core Wars.[8]

The programs written and used for core wars are a far cry from the code that allowed the worm to infect and estimated 6000 computers world wide. The worm was designed to exploit flaws in a UNIX operating system, and then only in certain types of machines.[7] This in turn differs from the dozens of viruses that have plagued personal computer users everywhere. When the media started to report stories of computer epidemics, everything was glazed with the generalized name virus. Actually there are several different classifications of replicant code. As with most topics in computer science, there aren't any sharp lines drawn to distinguish types but several generally accepted guidelines are used below. One thing that can be generalized is that they are all computer programs, usually written with mischievous or malicious intent. During some of the initial media reports, people were fearful that they could catch and get sick from computer viruses. This is, of course, totally ridiculous because the viruses are only programs and not biological organisms.

A real virus, which is a living organism, attaches itself to a cell and forces it to duplicate itself over and over again. A computer virus is so named because it behaves in much the same manner, embedding itself in another program or file. Once a virus comes in contact with a system, it typically attacks by altering the operating system, the master program that drives a computer. The corrupted operating system places copies of the virus into other programs that it comes into contact with. If this other software is run again, it will have the same ability to corrupt the operating system and infect other software. When possible the virus also corrupts the master copy of the operating systems so that the computer system will be infected as soon as it is started up.

One common strategy used to spread a virus is to hide the code within another program. This is known as the Trojan Horse method. Naturally, users won't operate on a system they know is infected. Therefore to get the bug into other systems, they place the virus inside a very attractive package, say a word processor or a game. The new user doesn't think anything of using the new program and soon the virus has spread throughout his entire library of software. Several hackers were especially devious in their choice of a trojan horse program. A program called flushot3 was designed to fight/detect viruses. Rather then being commercially available, it used the concept of shareware distribution and was readily available on many bulletin boards. The problem was that vandals modified copies of flushot3 and inserted viruses in them.[10] Then instead of protecting their systems, people were actually infecting them.

A worm, like the one that attacked Internet, differs from a virus because it is a self contained program. This means that it doesn't attach itself to other software. Once in a system, it remains a separate entity and survives by living off of flaws in the host system's logic. In the Internet infection, several computer labs remained uneffected because they were using modified versions of UNIX.[11] These nonstandard versions had eliminated the well known weaknesses of UNIX, weaknesses that have been recognized for years but often ignored.

A bacterium is a program that is identified more by its results than its methods. It keeps duplicating itself, usually by exploiting a weakness in the host system. Eventually the system is slowed down to a snails pace just by the sheer magnitude of jobs created by the bacterium. It doesn't actually alter or damage anything but the system is rendered ineffective because most of the processor time is used to create and send out clones of the program. A case of this occurred around Christmas 1987. Somehow a "Christmas Card" got into the BITnet network. Aside from the seasons greeting, it drew a picture of a christmas tree on the screen. At the same time, it sent a copy of itself to everyone on the current users mailing list. It propagated very rapidly and bogged down the network.[12] It was necessary to shut down the network to clear the forest.

Both worms and viruses potentially pose different problems than bacterium because they may include routines that perform special functions, rather than just survival. Their purpose may be something as playful and harmless as to display a message asking for cookies; its purpose may be something as potentially harmful as wiping out a data base. Often this hidden routine is constructed so that it executes at a predetermined date, after a given number or repetitions, or whenever some other specified conditions are meet. This "time bomb" effect is what makes infections particularly worrisome.

A classic time bomb was incorporated by a program dubbed the PLO virus. It turned up at the Hebrew University and other sites throughout Israel. It included a couple of time linked functions. On the thirteenth of every month, it would reproduce madly. Its primary and most destructive function was set for Friday, May thirteenth, 1988. On this date it would erase all information stored in memory and on all accessible disks. This virus reportedly spread to computers used by the Isreali Defense Force and at the ministry of educations' educational center, it destroyed fifteen thousand dollars worth of software and over seven thousand man hours of research.[13]

Another prolific virus that uses a time bomb, though not with that regularity, was the (C) BRAIN virus, also known as the Pakistani virus. This virus was developed by two brothers who were self taught programmers. They ran a computer store in Lahore, Pakistan. Originally they inserted the virus only in software of their own creation. If anyone attempted to illegally copy their programs, the bootlegged version would eventually malfunction. The pirate would then be forced to come to them to get it fixed, if at all. Soon the Alvi brother began running their own pirating operation, though they claimed it was legal due to a loophole in Pakistani law. They sold versions of popular programs such as Lotus 1-2-3 and Wordstar at cut-rate prices. But they included the virus in versions sold to foreigners, particularly Americans. They reasoned that copy rights didn't include software under Pakistan's laws, therefore local people who bought the software weren't breaking the law.

Foreigners, however, were pirates and deserved to be punished and got contaminated versions. [14]

The Pakistani virus and altered versions of it have been found all over the world. It gained a lot of attention when a reporter for the Providence Journal-Bulletin discovered that her disks had been infected by the virus. Froma Joselow, a financial reporter, was preparing to write a story and tried to access her disk that contained six months of notes and interviews; when she kept getting disk errors, she took the disk to the newspaper's computer center. The systems analyst found a message hidden in the jumble of data: "WELCOME TO THE DUNGEON . . . CONTACT US FOR VACCINATION." It also had the address and phone number of the Alvi brothers' computer store in Pakistan.[15]

The message is the same one that has greeted thousands of university students across the country. Because students were the most frequent customers at Brain Computer Services, there is a higher concentration of computer usage on campuses, and not much consideration given to borrowing and copying software in the student environment, universities have been the sites of several epidemics of the Brain virus. The University of Miami at Ohio state was the site of one such outbreak. Another campus that was struck by the virus was Southern Illinois University at Carbondale.

In the middle of the fall 1988 semester, students began having problems with their software. There were numerous complaints of data being lost, especially from business students and others in the college of business. In the main computer lab in Faner Hall, students are able to check out software from a library which includes Lotus 1-2-3, Wordstar and many other programs. Many of the students affected were working on a Lotus 1-2-3 project. It was estimated that two hundred students in that class alone had their software exposed to the Pakistani virus. Evidently someone had a bootlegged version of Lotus or some other program and used it or an infected data disk while using software checked out from Faner Lab. In this way, someone managed to infect the library's software. Then another student checked it out and caught the virus; the cycle just went on and on from there.

Bill Baron, lab director for Computing Affairs at SIU, said that he had heard talk of viral epidemics but had no reason to expect one at SIU. He also said its severity was partly Computing Affairs fault. "Our disks weren't write protected (in the software library). We were being overly benevolent. Many people who come in and use programs like PC Write don't even have a working disk. So they put their working file on our the disk so they can print their paper." He added that not having the write protect tabs (which would prevent the virus from altering the disk) also made it easier when lab workers went to reconfigure the disks. The epidemic was severe enough that computing affairs shut down the software library.

The library was shut down for three days, in which they implemented a three part plan to clean up the Pakistani virus at SIU. They consider there to be three types of software: computing affairs, faculty for instruction, and user(student).

It was decided to clean up computing affairs first, since they provide the majority of software on campus. They had to completely rebuild their libraries from the manufacturers originals. Normally copies are made from masters, copies of the originals that are configured for SIU's particular terminals, but even the masters had been corrupted.

The second phase was to verify the integrity of instructor supplied software - special software that professor leave to be checked out by students. They notified all faculty that their software was quarantined until they came and personally verified that it was free of infection and signed a letter to that effect.

Phase three was to clear up, as much as possible, user software - that is software that students carry around. To achieve this goal, a check station was set up in Faner lab. At the station, lab workers would check anyone's software for viruses and if requested, to eliminate it. Mr. Baron said they assumed most computer science majors and other with computer knowledge would have already taken care of their software; the station, which was operated for two weeks, was for everyone else. The service was provided free to students but not to computing affairs. It cost about six-hundred additional dollars in salaries to man the station.

Measures have been taken to insure that this won't happen again. All of computing affairs disk are specially write protected. Rather than the normal tabs that can be peeled on and of, special labels were attached. If anyone removes the tab, it will probably rip, or at least be noticed. lab assistants set aside any software that appears to have been tampered with, to be examined later. Also a policy has been instituted that anyone who removes a write-protect tab will temporarily lose lab privileges. While Mr. Baron has faith in these measures, he knows that SIU isn't immune. Currently a virus that infects Macintoshes is plaguing computing affairs. This is a virus that, because the system it attacks is very unusual, will take quite some time to eliminate.

In the case of the viral attack at SIU-C, the real risk of doing any widespread damage was limited because the virus attacked personal computers. A person's danger was limited to how much he used someone else's software and how careful he was about backing up his own. With a few simple, common sense precautions, the chance of infection was slim. To further insure the integrity of your personal computer, there are many programs available that can aid in countering viruses. As always, when there is a demand for a product, business world is ready to respond. After viruses gained wide notoriety in the fall on 1988, the software industry came to the rescue. Within the span of several months where,there had been a void, there were suddenly dozens of programs ready to end your virus woes.

With such reassuring names as Disk Watcher and Guard Dog, people were sure that their virus worries were over; but in a recent test conducted by PC Magazine found that no software was completely successful against viruses. They tried out eleven of the most popular anti-viral products. As a test, three viruses that attack in different ways were used against the packages; no program detected all three but a couple did do very well.[16] Nothing, aside from living in a glass house and writing all of your own software can absolutely guarantee your computer's security. The problem with developing technical solutions against viruses is that the people who create viruses are just as ingenious as those who defend against them. It can be seen as a tit-for-tat war; someone writes a virus - someone else develops a defense; another figures out a way to breach that defense - yet another finds a way to improve the defense. The cycle doesn't end.

If technical solutions are temporary fixes at best, what can be done to stem the tide of virus attacks? A idea that is more applicable at the industrial/commercial level is more emphasis on physical security - that is restricting physical access to the computer systems and placing tight checks and usage requirements. There are also methods to prevent remote access from unauthorized locations. The government's data transmission network is the ultimate example of this. They employ private communication lines in gas filled tubes;[17] no one could causally reach their computers and if they tried to tap the lines, an alarm would be sounded. This level of prevention is too costly to be practical in most other situations. There are additional problems in restricting access and causing legitimate users untold headaches just trying to logon. A final consideration is that the viruses that have done the most real damage in terms of data lost have been loosed by someone on the inside, usually by disgruntled former employees. All of the security is for naught if the culprit is/was a legitimate user. There may be ways to limit what an employee can do but these are case specific.

An old tool that is only beginning to be utilized in the fight against viruses and computer crime in general is the law. People feel that if there were strict punishments associated with loosing viruses, this would be a sufficient deterrent. Over the past three years, legislators have scrambled to make laws that would deal with the problems. A problem arises in that the problems are coming faster than the laws. They're playing catch up but as far back as 1979, the American Bar Association has been on record in favor of a uniform federal computer crime legislation.[18] There are laws dealing with computer crimes in most states and in 1987, Congress passed the Computer Security Act[19] and the Federal Computer Crime Act in 1988.[20] More laws are undoubtedly on the way. So far there has been only one test case involving a virus. In a civil suit in Texas, a programmer was required to pay $12,000 to his former employer after destroying over 100,000 records of sales commissions. The case also went to criminal court where he could face up to ten years in prison.[21]

In many cases finding and proving beyond a reasonable doubt that someone created a virus will be difficult, to say the least. And once again there is an additional problem. Companies, especially those who handle data storage and processing for others, may be reluctant to admit that they have been breached by a virus. Having a long public trial about the gaps in their security is not in their best interest. Most companies simply cover it up deny that there was ever a problem. Even when a former employees are the perpetrators, they are sent off with a pat on the back rather than a date in court. One company even gave a going away party to a former employee to smooth things over.[22]

Even Dr. Harold Highland, the editor-in-chief of Computers and Security magazine encouraged cover ups. "My recommendation to a corporate entity would be to deny it immediately. I have advised industry that if anything like this happens and you can kill it by denying it, kill it."[23] This is reasonable from one perspective - a lot of publicity only puts the spotlight on vulnerable companies; There is also the fear of copycat crimes if media exposure is too great. It is open to debate though whether the fear of punishment after several successful prosecutions would offset the chance of copycats. Other companies and the public in general could benefit by being made aware of the potential dangers that lie in wait for them.

Where the real and potentially life-threatening danger lies is in viral attacks on networks. Untold harm could be done if a virus got into a hospital's records or managed to disrupt an air traffic control network. The risk grows greater and greater every day, as computers become more interconnected and more compatible and access easier to gain. Robert Morris Jr.'s virus, although its effects were felt worldwide, was only an inconvenience. He was playing a game and didn't want to hurt anyone; the stakes might be higher in the next game. For the most part, luck has kept the computer industry from a major disaster. The Internet attack served as a wake up call to experts in the field. This time there was no permanent damage. Will we be so lucky next time ?

Notes

[1] Phillip Elmer-Dewitt, "Invasion of the Data Snatchers!" Time, 26 Sept. 1988, p. 65.

[2] Ibid., pp. 65-66.

[3] Ibid., p. 66.

[4] Phillp Fites, Peter Johnson and Martin Kratz, The Computer Virus Crisis (New York: Van Nostrand Reinhold), p. 17.

[5] Eliot Marshall, "Worm Invades Computer Networks," Science, 11 Nov. 1988, p. 855.

[6] Ibid., p. 855.

[7] Eliot Marshall, "The Worm's Aftermath," Science, 25 Nov. 1988, p. 1121.

[8] Elmer-Dewitt, p. 65.

[9] Marshall, p. 855.

[10] Fites, p. 137.

[11] Marshall, p. 855.

[12] Elmer-Dewitt, p. 64.

[13] Fites, pp. 122-24.

[14] Elmer-Dewitt, p. 66.

[15] Ibid., p. 62.

[16] Neil J. Rubenking, "Infection Protection," PC Magazine, 25 April 1989, p. 193-228.

[17] Katherine M. Hafner, "Is Your Computer Safe?" Business Week, 1 Aug. 1988, p. 72.

[18] I. Peterson, Science News, 23 June 1984, p. 390.

[19] Hafner, p. 70.

[20] Fites, p. 113.

[21] Elmer-Dewitt, p. 63.

[22] Hafner, p. 67.

[23] Fites, p. 110.

Bibliography

Denning, Peter J. "Computer Viruses." American Scientist, May-June 1988, pp. 236-38.

Elmer-Dewitt, Phillip. "Invasion of the Data Snatchers!" Time, 26 Sept. 1988, pp.62-67.

Fites, Phillip, Peter Johnson and Martin Kratz. The Computer Virus Crisis. New York: Van Nostrand Reinhold, 1989.

Hafner, Katherine M. "Is Your Computer Safe?" Business Week, 1 Aug. 1988, pp. 64-72.

Marshall, Eliot. "The Worm's Aftermath." Science, 25 Nov. 1988, pp. 855-56.

Marshall, Eliot. "Worm Invades Computer Network." Science, 11 Nov. 1988, pp. 1121-22.

Morrison, Perry R. "Computer Parasites." The Futurist, March-April 1986, pp. 36-38.

Peterson, I. Science News, 23 June 1984, p. 390.

Rubenking, Neil J. "Infection Protection." PC Magazine, 25 April 1989, pp. 193-228.