Early History of the Computer Virus

By:
Craig Miles

For:
Prof. Dasgupta's History of Computer Science
The Center for Advanced Computer Studies
University of Louisiana at Lafayette

Table of Contents

1. Virus Prehistory (1948 – 1981)

It may be argued that the history of computer science begins with Charles Babbage's (b. 1791, d. 1871) pioneering work in the nineteenth century. If we assume that computer science did emerge around 1820 when Mr. Babbage began tinkering with his difference engine, then, as I write this today, the field of computer science is not far from its 200th birthday. It is interesting to note, then, that a major facet of modern computing, which causes worldwide economic damage to the tune of fifteen billion USD annually (1) and that has been referred to over a million times in the scholarly literature of computer science (2), did not appear until 1982. In that year, sixteen decades after the birth of computer science, a 9th grader at Mt. Lebanon High School in Pennsylvania wrote a computer program he called Elk Cloner. The major facet of modern computing to which I previously referred, of course, is the computer virus, and Richard "Rich" Skrenta's (b. 1967) Elk Cloner program was the first computer virus ever written (3).

Just because the first software implementation of a computer virus was not realized until 1982, however, does not imply that no one foresaw the ability of machines or software to automatically replicate themselves. Automatic self-replication, by the way, is what I consider to be the most important, and indeed necessary, property of a computer virus. I will provide a more formal definition of what it means to be a computer virus later on, but for now, the reader may simply assume that a computer virus is a malicious computer program with the ability to automatically self-replicate. We will come back to the Elk Cloner virus, as well as several others, in the next section, but first let us explore the time when the computer virus was merely theory; that is, the prehistory of the computer virus.

If the history of the computer virus begins with Rich Skrenta in a Pennsylvania high school, then the prehistory of the computer virus begins with John von Neumann (b. 1903, d. 1957) at the Institute for Advanced Study in Princeton, NJ. In 1948, von Neumann presented "The General and Logical Theory of Automata" at the Hixon Symposium on Cerebral Mechanisms in Behavior. The work was published in a journal of the same name in 1951 (4), and further works by von Neumann in the field of automata were forthcoming (5, 6). Automata are traditionally thought of as moving mechanical devices made in imitation of human beings; however, the phrase has also come to refer to a machine that performs a function according to a predetermined set of coded instructions. It should be apparent that the latter definition of automata sounds very much like that of computer software. Von Neumann's work laid the theoretical foundation for self-replicating automata, and therefore, for self-replicating software as well.

In von Neumann's first approach to modeling self-replicating automata, which he called the "kinematic model", he described a self-replicating physical machine with access to a *sea* of raw materials which it could use to create new copies of itself. The described machine consists in part of a memory tape on which there exists a set of coded instructions; that is, a program. This program contains the instructions necessary to retrieve parts from the *sea* using a manipulator, assemble those parts into a duplicate of the original machine, and then copy the contents of the memory tape onto the empty memory tape of the newly assembled machine. Though qualitatively sound, von Neumann's kinematic model of self-replicating automata does not lend itself nicely to being analyzed with any amount of mathematical rigor. Thus, in his subsequent work and on the suggestion of his friend, the mathematician and fellow Manhattan Project collaborator, Stanislaw Ulam (b. 1909, d. 1984), von Neumann further abstracted his model of self-replicating automata by describing it via the processes of cellular automation rather than the three-dimensional physical machines of the kinematic model.

Using the cellular automation processes, von Neumann described a *universal constructor*. While the technical details of the universal constructor are beyond the scope of this report, the reader should know that it is often viewed as a demonstration of the logical requirements necessary for machine self-replication. However, the crucial insight that allowed von Neumann to create his universal constructor (and his kinematic model as well) deserves treatment, because the very same insight is directly applicable to our subsequent study of computer viruses. This crucial insight is that the replicator component of the self-replicating machine is used in two distinct manners: first, it is an active component of the construction mechanism, and second, it is the target of a passive copying process. The same is true of computer viruses today.

Though subsequent mathematicians, scientists, and biologists have greatly built upon von Neumann's self-replicating automata work, to the extent even that an academic named Veith Risak (b. 1936) described a fully functional self-replicating program written in assembler language for the SIEMENS 4004/35 computer system in 1972 (7), I do not wish to further detail that work here. While that corpus of research may contain models which accurately describe the self-replication and transmission of computer viruses, academia has never been a major driving force in the design and development of real computer viruses. In fact, there is no evidence that the authors of the earliest computer viruses had even heard of the academic work on self-replicating automata. It is for that reason which I choose to quickly move past the prehistory and forge on ahead toward the history of the actual computer viruses.

Before we move on, however, we should quickly acknowledge a bit of prehistory trivium which perhaps provided some of the impetus for a self-replicating malicious program to be called a virus. I refer now to the usage of the word 'bug' to describe an error or fault in a computer system that produces an incorrect or unexpected result. A common, yet fanciful, etymology for 'bug' in the context of computer systems goes something like this (8):

> In 1946, when [Grace Murray] Hopper was released from active duty, she joined the Harvard Faculty at the Computation Laboratory where she continued her work on the Mark II and Mark III. She traced an error in the Mark II to a moth trapped in a relay, coining the term *bug*. This bug was carefully removed and taped to the log book. Stemming from the first *bug*, today we call errors or glitch's in a program a *bug.*

In truth, the word 'bug' predates the electronic computer by decades, having even been defined by Thomas Edison (b. 1847, d. 1931) in 1878 (9) as "little faults and difficulties" in an engineering project. For the record, it also was not Grace Murray Hopper (b. 1906, d. 1992) who found the moth in the computer, but rather a technician named Bill Burke (10) who removed the insect from the machine and taped it to a page in the log book with the annotation, “Moth in relay, first actual case of a bug being found.” Regardless of who found the moth, it is likely that Rear Admiral Hopper’s retelling of its story lead to the popularization and continued usage of the term in the context of computing. No matter how usage of the word ‘bug’ came about in the realm of computing, it is yet another example, along with von Neumann’s usage of cellular automaton processes, of the application of words and concepts from biology onto computer science. Most interesting, however, is that ‘bug’ is a synonym of ‘virus’, and this connection may not have been lost on Leonard “Len” Adleman (b. 1945) when he coined the phrase computer ‘virus’ in or around 1984.

2. The First Computer Viruses (1982 - 1986)

```
ELK CLONER

THE PROGRAM WITH A PERSONALITY

IT WILL GET ON ALL YOUR DISKS
IT WILL INFILTRATE YOUR CHIPS
YES IT'S CLONER!

IT WILL STICK TO YOU LIKE GLUE
IT WILL MODIFY RAM TOO
SEND IN THE CLONER!
```

- Output from Rich Skrenta’s 1982 Apple II computer virus, “Elk Cloner”

The above text is the message displayed by the first computer virus ever written (11). Elk Cloner, Rich Skrenta’s 1982 virus for the Apple II platform, was a boot sector virus. A boot sector is a region of a disk or storage device that contains machine code to be automatically loaded and executed by a computer system’s built in firmware (12). Since the code in a boot sector is automatically executed, it has become a common place to hide computer viruses, and Elk Cloner was the first of that ilk. In the case of Elk Cloner, when a computer was booted from a floppy disk infected with the virus, the virus’ code would be loaded into the system’s memory and executed. Upon every fiftieth boot of the system, the above message would be displayed to the user in lieu of whatever the expected behavior may have been. A computer program that merely displays a message to the user, however, is not a computer virus. What made Elk Cloner the first computer virus was its ability to self-replicate. Once a system was infected with Elk Cloner, anytime an uninfected floppy disk was inserted into the machine, the system would copy Elk Cloner into the disk’s boot sector thereby infecting it.

It is immediately evident that Elk Cloner was not malicious in the same manner as are modern computer viruses and malware. Rather than attempting to steal or destroy data, the virus was merely a practical joke to be played upon the young Mr. Skrenta’s friends. Though Elk Cloner was not destructive, the intent being to annoy is sufficient enough to be considered malicious behavior. Therefore, Elk

Cloner and other early viruses that merely annoy the user of the infected machine can be considered computer viruses under the informal definition of the phrase presented in the previous section.

Having described at a high level the technical details of the Elk Cloner virus, we now ask: Why would a 15 year old high school student write such a program? The answer, as was alluded to previously, is that Mr. Skrenta wanted to prank his friends. At the time, he and his group of friends often traded pirated copies of games via floppy disks. Mr. Skrenta had a penchant for modifying the games he was distributing to his friends such that upon the game reaching some particular condition, a message inserted by Mr. Skrenta would be displayed. This type of behavior, in present-centered language, is referred to as a logic bomb. Because his friends had grown wary of accepting disks from him, Mr. Skrenta was left to come up with a new way to annoy his friends, lest they go un-annoyed. It was this desire to annoy his friends while they were playing games, but without the ability to physically hand them the game disk, that lead Mr. Skrenta to look for ways to get his code onto their systems in a less direct fashion. What follows is Mr. Skrenta's recollection of the moment in which he realized he could continue to annoy his friends unabated (13):

> The *aha* moment was when I realized I could essentially get my program to *move around by itself*. I could give it its own *motive force*, by having it hide in the resident RAM of the machine between floppy changes, and hitching a ride onto the next floppy that would be inserted. Whoa. *That would be cool.* Insight without implementation is worthless, so to work I went.

Though Mr. Skrenta implies here that his ability to cause a program to automatically self-replicate was 'cool', he seems to have internalized a bit of cognitive dissonance about just how revolutionary his discovery was. In a posting to USENET in 1990, he stated the following (14):

> I don't even have my Apple II anymore, I gave it away. I wrote a lot of stuff for the Apple II--obscure adventure games, a small compiler, a toy multi-user operating system. The stupidest hack I ever coded generated the most interest, and lives on to this day.

Whether or not Mr. Skrenta believes his Elk Cloner program was revolutionary or stupid, however, is perhaps irrelevant. What matters is that Elk Cloner was the first instance of a class of software which has grown to include nearly fourteen million members as of November 2011 (as counted by the unique virus signatures in modern anti-virus products) (15).

Backing up for a moment, I should acknowledge that three programs, each independently written before Elk Cloner's release, have each been considered by some to be a computer virus. The first such program, called Creeper, was written by Robert Thomas in 1971. While Creeper was indeed self-replicating, it replicated itself across a closed network that was intentionally configured to allow a process on one node to write data into the memory of and execute code on another node (16). Furthermore, after displaying an initial message and replicating itself onto other network nodes, Creeper was never executed again. As such, Creeper fails to meet the criterion of maliciousness, as it was nothing more than proof-of-concept that the data copied across the previously described network could be the code of the program currently executing. Another such quasi-virus was Wabbit, author unknown. Wabbit was a program that was indeed self-replicating and malicious, but it only replicated itself on the local machine. Its maliciousness was derived from the fact that it continued to replicate itself until no room was left on the disk, thereby starving other processes of disk resources. However, Wabbit cannot be considered a computer virus because it had no ability to spread itself onto any system other than one on which it was originally executed. While it was certainly malicious software, Wabbit was not a virus because it required a human user to load it on each new system. The third program that some might consider to be an early example of a virus is ANIMAL, written by John Walker (b. 1950). ANIMAL was a game for the UNIVAC system in which the computer would ask the user simple questions about the animal he or she was thinking about before attempting to guess what animal the user had in mind. Upon being loaded, ANIMAL would copy itself into each directory writable by the user. Because users often shared directories with other users on the multiuser system, the game soon replicated itself across nearly all of the user accounts. Once a user with superuser privileges (root) executed the program, it would be copied to every single directory of every single user account on the system. Like Creeper, however, ANIMAL fails to meet the maliciousness criterion for being a computer virus. The author simply intended to spread his non-malicious game to as many users as possible. I offer as proof of ANIMAL's non-malicious intent comments from the source code of the module responsible for copying the game to new directories (17):

```
THIS PROGRAM IS A TOTALLY NEW WAY OF DISTRIBUTING VERSIONS OF
SOFTWARE THROUGHOUT THE 1100 SERIES USER COMMUNITY.  PREVIOUS
METHODS REQUIRED THE DELIBERATE AND PLANNED INTERCHANGE OF TAPES,
CARD DECKS, OR OTHER TRANSFER MEDIA.  THE ADVENT OF 'PERVADE'
PERMITS SOFTWARE TO BE RELEASED IN SUCH A MANNER THAT IF SOMEONE
CALLS YOU UP AND ASKS FOR A VERSION OF A PROCESSOR, VERY LIKELY
YOU CAN TELL THEM THAT THEY ALREADY HAVE IT, MUCH TO THEIR OWN
SURPRISE.
```

I hope we may now agree that Elk Cloner was indeed the world's first computer virus. If, in fact, that is the case, then what came next? Brain did. Brain holds the distinction of being the world's first computer virus for the MS-DOS system, and also the first IBM compatible virus. Like Elk Cloner, Brain also infected the boot sector of storage media. The primary difference between the two viruses, however, was that Brain was even more pervasive than its authors originally intended. In an interesting turn of events that has reoccurred multiple times throughout the history of computer viruses, Brain spread itself onto more systems than its authors ever intended. You see, Brain was only ever intended (or so its authors have said) to be malicious towards one specific class of people: those who had pirated the author's commercial software. Like Elk Cloner, the malicious act of the Brain virus was simply to annoy; it did so by nagging software pirates to contact the authors in order that they could disinfect the system and demand payment. To that end, the authors of the Brain virus, two brothers named Basit and Amjad Farooq Alvi from Lahore, Pakistan, actually included their names, address, and telephone number within the virus. What they could not foresee, however, was the extent to which their Brain would spread. An article in Time Magazine from 1988 stated that the virus had replicated itself onto over 100,000 floppy disks around the world (18), a much larger number than there were installations of the brother's commercial software.

We see that the age of the first computer viruses is marked by innocence. These forerunners of what has become a modern scourge were remarkably benign in comparison to their newer counterparts. In an interview (19) with a computer analyst who had analyzed the Brain virus twenty-five years prior, the Alvi brothers stated that modern malware is unrecognizable compared to what they created, and this new generation of destruction and theft is nothing but criminal activity. When asked how they perceived their own virus, the brothers responded, "friendly."

3. The Response (1984)

While the earliest computer viruses could be characterized as being relatively innocent and benign in nature, at least one person saw the potential for them to become a major threat. Frederick “Fred” Cohen (b. 1957) was the first academic to publish work in computer science’s scholarly literature relating to computer viruses and methods for defending computer systems against the threats that they pose. His definition of a computer virus, the first such definition, was published in the paper “Computer Viruses Theory and Experiments (20)” in 1984. The definition follows:

> We define a computer ‘virus’ as a program that can ‘infect’ other programs by modifying them to include a possibly evolved copy of itself. With the infection property, a virus can spread throughout a computer system or network using the authorization of every user using it to infect their programs. Every program that gets infected, may also act as a virus and thus the infection grows.

Conspicuously missing from Cohen’s inceptive definition of a computer virus is any mention of maliciousness. In fact, Cohen did not consider maliciousness a requisite feature of a computer virus, as the following quotation attests (20):

> It should be pointed out that a virus need not be used for evil purposes or be a Trojan horse.

Of course, definitions must evolve over time to keep up with the concepts they seek to describe. Today, in this author’s opinion, one would have great trouble in finding a person knowledgeable in the field of computing who does not implicitly equate the phrase computer virus with software of malicious intent. For that reason, and as was discussed in the previous sections, this report considers an appropriate definition of a computer virus to be Dr. Cohen’s original definition supplemented by the requirement of maliciousness.

Though Dr. Cohen’s paper was the first to make use of the phrase ‘computer virus’ in a scholarly work, he gives credit for the coining of the phrase to Leonard “Len” Adleman (b. 1945). I mentioned in a previous section that Dr. Adleman’s usage of ‘virus’ was likely inspired by a perceived similarity between biological systems and computer systems. Providing further evidence to this suggestion is that Dr. Adleman is not only a professor of computer science as the University of Southern California, but he is also a professor of molecular biology. However, other members of the computing community, as we shall soon see, were not so quick to draw the same parallel.

Doing research costs money, thereby necessitating researchers to write grant proposals. The committee who reviews and determines whether or not such grant proposals should be approved is generally hoped to be populated by members at the forefront of the academic community towards which the proposed research is targeted. In a keynote address to one of the first academic conferences on computer viruses (21), Dr. Cohen shared some of the reviews he had received from a National Science Foundation committee for a proposal requesting funds to research practical defenses to computer viruses. Particularly interesting is the following quote:

> …In fact, the whole area of ‘virus’ is a lot of hype. Sound engineering practices are not being considered. Comparison to biological systems is at best silly. The PI is not likely to produce any interesting basic research result.

It is interesting, entertaining, and perhaps saddening to note that while numerous agencies, including the NSF, today provide millions of dollars to conduct research with the intent of stemming the flow of malware, twenty-five years ago those same agencies and the supposed experts within could not even believe such an intractable threat could possible exist.

Thankfully, Dr. Cohen was not deterred from his work, as he continued to publish on the topic well into the 1990s. Over time, a majority of computer scientists have come to agree with his characterizations and theories regarding computer viruses, and his work later served as the theory behind much of the development of anti-virus software.

4. Concluding Remarks

Though the computer virus and other types of malicious software are relative newcomers to the field of computing, they affect every computer user. As such, it is interesting to see how the modern scourge that is malicious software evolved from the relatively innocent tinkering of early computer hobbyists. That the computer virus affects every single person who makes use of computers today is evidence enough that the story of those involved in its earliest development should be told. I hope to have accomplished that through this report.

Certainly much has happened in the world of computer viruses since the point where I have left off: Viruses and other malicious software have attacked millions of computer systems, anti-virus has become a multi-billion dollar industry, and the newest generations of malicious software are capable of targeting whole industrial facilities rather than individual computer systems. But all of that had to start somewhere. This has been the story of that beginning.

5. Bibliography

1. Computer-Economics. Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets, and Other Malicious Code. Tech. Rep., Jun, 2007.
2. Google. Google Scholar Search Results for Phrase: Computer Virus. 2011 [cited 2011 November 30]; Available from: http://scholar.google.com/scholar?q=computer+virus.
3. Szor P. The art of computer virus research and defense: Addison-Wesley Professional; 2005.
4. Von Neumann J. The general and logical theory of automata. Cerebral mechanisms in behavior. 1951:1-41.
5. Von Neumann J. Theory and organization of complicated automata. Burks (1966). 1949:29-87.
6. Von Neumann J. The theory of automata: Construction, reproduction, homogeneity. of:(Burks 1966) Based on an unfinished manuscript by von Neumann Edited for publication by AW Burks. 1966:89-250.
7. Risak V. Selbstreproduzierende Automaten mit minimaler Informationsübertragung. Elektrotechnik und Maschinenbau. 1972;89(11):449-57.
8. Danis SA. Rear Admiral Grace Murray Hopper. 1997 [updated February 16, 1997; cited 2011 November 11] Available from: http://ei.cs.vt.edu/~history/Hopper.Danis.html.
9. Edison T. Edison to Puskas. Edison papers, Edison National Laboratory, U.S. National Park Service, West Orange, N.J.1878.
10. Lee JAN. Howard Aiken's third machine: the Howard Mark III calculator or Aiken-Dahlgren electronic calculator. Annals of the History of Computing, IEEE. 2000;22(1):62-81.
11. Wikipedia. Elk Cloner --- Wikipedia, The Free Encyclopedia. 2011 [cited 2011 November 23]; Available from: http://en.wikipedia.org/w/index.php?title=Elk_Cloner&oldid=450328896.
12. Wikipedia_Collaborators. Boot Sector. 2011 [updated November 19, 2011; cited 2011 November 26]; Available from: http://en.wikipedia.org/wiki/Boot_sector.
13. Skrenta R. The joy of the hack. 2007 [updated January 26, 2007; cited 2011 November 20]; Available from: http://www.skrenta.com/2007/01/the_joy_of_the_hack.html.
14. Skrenta R. Re: A (long) story about an (old) Apple][virus. 1990 [cited 2011 November 21]; Available from: http://www.skrenta.com/cloner/clone-post.html.
15. Triumfant. The Worldwide Malware Signature Counter. 2011 [cited 2011 November 23]; Available from: http://www.triumfant.com/Signature_Counter.asp.
16. Schantz RE. BBN's network computing software infrastructure and distributed applications (1970-1990). Annals of the History of Computing, IEEE. 2006;28(1):72-88.
17. Walker J. ANIMAL - PERVADE. 1975.
18. PHILIP ELMER-DEWITT RM. Technology: You Must Be Punished. Time Magazine. 1988.
19. F-Secure. http://campaigns.f-secure.com/brain/. 2011 [updated February, 2011; cited 2011 November 24]; Available from: http://campaigns.f-secure.com/brain/.
20. Cohen F. Computer viruses:: Theory and experiments. Computers & security. 1987;6(1):22-35.
21. Cohen FB. Fred's Papers: ASP Press; 1988.