

## A History of Viruses

*Jeremy Paquette* 2000-07-17

If you want a wall covered in graffiti, combine the following: a person who wants to vandalize, a can of spray paint, a wall, and some people to see the results. Wherever you find these four elements together, you'll see something resembling the interior of the New York City subway system, where as fast as maintenance workers can scrub the lettering from the walls and trains, taggers come back and leave new marks for all to see.

If you want a computer virus, just put together a programmer who wants to vandalize, a little programming knowledge, some computers for the virus to spread to, and some people to see the results. Until the late 1970s, the first two items were available in relative abundance. Computers had already been a Big Deal for more than two decades, and the latent human drive to create mischief (or worse) was similarly well established. To continue the analogy, we had the taggers, and the spray paint. By 1980, the wall, and the people to see the results, were not far behind.

The Apple ][ Personal Computer was introduced to the world at the West Coast Computer Faire in April of 1977. It was the first personal computer featuring color graphics, and cost US\$1300.

At the same show, Commodore Business Machines unveiled a one-off prototype of their PET 2001 computer. It sold for \$600.

A little over a year later, the Atari 400 and 800 personal computers were released. In June 1979, Texas Instruments released the TI-99/4 personal computer, and in July of the following year, Radio Shack brought the \$230 TRS-80 Color Computer to market.

By the 1982 introduction of the Commodore 64, the world had about a dozen small, inexpensive computer platforms to choose from.

And choose we did. In droves. It was the first time in history that any person could walk into a store, put down a modest amount of money, and return home with a complete computer system. Almost overnight, hordes of people were writing, buying and trading software. A collection of malignant code snippets with names like Elk Cloner, Festering Hate, Blackout and Burp would soon appear.

In 1982, a junior high school student named Richard Skrenta had recently been given his first computer for Christmas - an Apple ][. Recalls Skrenta, "I had always been mechanically curious, taking apart tube radios and telephones and wiring up O gauge Lionel train sets when I was young. When I got an Apple II in the 7th grade, I was in heaven.

"I had been playing jokes on schoolmates by altering copies of pirated games to self-destruct after a number of plays. I'd give out a new game, they'd get hooked, but then the game would stop working with a snickering comment from me on the screen (9th grade humor at work here)".

Soon, classmates were getting wary of letting Skrenta near their disks. He needed a way to alter their floppies to contain his "booby traps" without physically being able to get his hands on them. "I hit on the idea to leave a residue in the operating system of the school's Apple II. The next user who came by, if they didn't do a clean reboot with their own disk, could then be touched by the code I left behind. I realized that a self-propagating program could be written, but rather than blowing up quickly, to the extent that it laid low it could spread beyond the first person to others as well. I coded up Elk Cloner and gave it a good start in life by infecting everyone's disks I could get my hands on."

Skrenta's choice of platform is perhaps the only reason why Elk Cloner is not often seen "in the wild" today - in the history of viruses, non-PC-based malware has never flourished as widely as those written for IBM-compatible systems.

In 1986, the most common microcomputer in the world was the IBM PC. Basit Farooq Alvi and Amjad Farooq Alvi were running a computer store in Lahore, Pakistan. The name of their store was Brain Computer Services. Recently, it had occurred to Basit and Amjad that the start of a floppy disk contained program instructions which the computer executes on startup. Like so many programmers after them, they elected to alter these instructions to their own ends.

To this day, their exact motives are pure speculation, though some believe it was to create a virus which would inhibit American software piracy. Whatever their reason, they intended to create a program which would spread from PC to PC. In accomplishing this goal, they created the (c)Brain virus.

In 1987, users at the University of Delaware started seeing the "(c)Brain" label on diskettes. Luckily, this was the extent of (c)Brain's repertoire. It copied itself, and overwrote the diskette's

volume label with the name of the virus.

Like Elk, Brain was pretty benign, as computer viruses go. It doesn't delete your files. It replicates and moves on. Again, being harmless is a good strategy for a virus - for biological ones as well as malware. Compare your chances of catching the common cold to, say, Hantavirus. The more damage the pathogen does, the more it gets noticed. With a little self-restraint, a virus like (c)Brain can survive in the wild for a long, long time. The volume label behavior has been changed in subsequent programmer-induced "mutations" of (c)Brain. Most likely, this is because it's too easy for an "infected" user to discover the presence of the virus merely by simply noticing the changed volume label.

Other viruses soon followed (c)Brain. Among them were Alameda (Yale), Cascade, Jerusalem, Lehigh and Miami (South African Friday the 13th). These viruses exhibited some new tricks: while (c)Brain infected the boot sector, the new guys were able to infect .COM and .EXE files.

As well as looking for new targets of infection, virus programmers were starting to hide their work. Cascade was the first encrypted virus - a technique meant to deter disassembly and detection of the virus' program code.

Variable encryption made its debut in 1989 with the "1260" virus. This now meant that scanning for fixed strings common to all copies of a given virus would be less effective, as the majority of the virus' code appeared in a different form with each successive infection.

The same year, the first stealth techniques emerged. These viruses could change or conceal their location in memory, actively avoiding detection and removal by anti-virus tools.

The Whale virus took this even further - it could rewrite its own instructions in such a way that it never consistently looks the same way twice. This marked the beginning of the end for string scanning and other techniques which relied on fixed, predictable elements appearing in each copy of a virus. As it happens, Whale was not particularly talented - it has been described as "too clumsy to work". Even still, its size and new, complex behavior made it famous.

Until this time, two classes of wild PC viruses had been reported. File infectors, which append themselves to executable programs, and boot sector viruses, which take advantage of the special segment of executable code at the beginning of a floppy or hard disk which is executed by the PC on startup.

By 1992, the prevalence of Windows had led to a shift in these numbers. While boot sector infectors coexisted peacefully with Windows 3.1, file infectors tailored to the DOS environment frequently crashed their hosts before they had a chance to spread effectively.

As a result, the inception of a large Windows-based computing community led to a decline in wild sightings of infector viruses, while the incidence rate for boot-sector infectors continued to increase.

As well as encouraging the decline of file infectors, the new, feature-rich Windows platform created a niche for a distinct category of malware: the macro virus.

A macro is a collection of user commands which can be stored and run as a group. Usually, macros are meant for automating repetitive tasks within a particular application (ie, search for instances of the word "abc" and replace with "xyz".) In the case of a macro virus, this language is used to create the routines which allow the macro virus' instructions to spread to and infect other documents.

The first reported macro virus, "Concept", was seen in the wild by AV researcher Sarah Gordon in summertime of 1995. A set of five macros designed only to replicate, Concept's payload displays the virus author's ominous message: "That's enough to prove my point".

These viruses exploited a new channel between users: document sharing. Until the inception of macro viruses, users could only spread viral infections via bootable diskettes or infected programs. By infecting data files, Concept and its descendents were able to leverage the fact that most users tend to exchange documents much more prolifically than they exchange programs.

In late July, 1996, Gordon received a wild copy of another macro virus, this time for MS Excel. Dubbed "XM.Laroux", it exploited the same type of macro language in which Concept was written, but it worked in MS Word's number-crunching cousin. Now spreadsheets, too, could contain harmful malware.

Again, the increasing tendency of users to exchange documents as part of their everyday work allowed this virus to spread. Moreover, by exploiting the emerging trend toward interoperability, the macro virus broke down the barriers between different computing platforms. A macro virus infecting a Microsoft Word or Excel document on a PC could now

spread to a Macintosh, or any other computer for which the application was available.

An excerpt from the source code of Melissa says it all:

```
"WORD/Melissa written by Kwyjibo  
Works in both Word 2000 and Word 97  
Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!  
Word -> Email | Word 97 <-> Word 2000 ... it's a new age! "
```

"A new age" is right. We've already come a long way from the Elk Cloner virus, which infected only one type of file (the boot sector) on one particular operating system (Apple DOS). Several developments would converge to increase the scope of the modern virus threat still further. Microsoft answered customer requests for more flexible, automation-friendly features with the 1997 introduction of the Windows Scripting Host. This technology gave different Windows applications a common programming language (VBScript), allowing new features to be added to applications as required. Unfortunately, it is also possible to use this language to write highly viable malware.

At the same time, greater connectivity between users (through the growing Internet and corporate networks) increased the widespread sharing of documents or spreadsheets as e-mail attachments. If an attachment was actually a Visual Basic script, concealed as a document, the Windows Scripting Host would still execute it in much the same way that Word or Excel would run a macro.

Microsoft's own comment says it best: "It's important to note that the virus payload cannot run by itself. In order for it to run, the recipient must open the mail, launch the payload by double-clicking on it, and answer 'yes' to a dialogue that warns of the dangers of running untrusted programs."

Surprisingly, when the "Love Bug" hit, many, many thousands of people did just that, accepting an attachment called "Love Letter" at face value. Within four hours of its release into the wild, VBS.LoveLetter was the most successful virus of all time. By May, 2000, roughly thirty variants of the virus had been identified.

Dozens of new viruses have made their debut since the Melissa and I.Love.You incidents. These descendants of the recent macro and VBScript threats have earned headlines of their own, and

despite the wide release of software patches and instructions for avoiding these threats, users are still reporting infections by the hundreds. Given that the most prevalent virus on record spreads only with the "permission" of each infected user, it seems that human factors are beginning to outweigh technical ones as the primary genesis of viral security threats. The problem of teaching the user population to practice safe computing is a challenge at least equal to the technical question of making computers safe from malicious code.

Clearly, it's anyone's guess what the next virus threat will look like. But no matter how the next chapter in virus history may unfold, it will always come down to the malicious programmer and his desire for an audience.

## Relevant Links

[Guidelines for an Anti-Virus Policy](#)

*Dr. Solomon's*

[Virus Glossary](#)

*McAfee*

[Worms](#)

*Dr. Solomon's*

[Who Writes This Stuff?](#)

*Sarah Gordon*

[Virus naming conventions](#)

[Microsoft on VBS/Loveletter](#)

[Privacy Statement](#)

Copyright 2006, SecurityFocus