Computer Viruses as Artificial Life

Following papers by Eugene H. Spafford *

A computer virus is a segment of machine code (typically 200-4000 bytes) that will copy itself into one or more larger "host" programs when it is activated. When these infected programs are run, the viral code is executed and the virus spreads further.

Computer viruses flourish in the weaker security environment of the personal computer. Personal computers were originally designed for a single dedicated user. The systems contained no security facilities, and there was minimal amount of security-related software available to safeguard data. Today, however, personal computers are being used for tasks far different from those originally envisioned, including managing company databases and participating in networks of computer systems. Unfortunately, their hardware and operating systems are still based on the assumption of single trusted user access, and this allows computer viruses to spread and flourish on those machines.

Viruses have two major components: one that handles the spread of the virus, and a "manipulation" task.

Classification of computer viruses based on the ways a virus may add itself to host code:

- Shell viruses: The virus "becomes" the program, and the original host program becomes an internal subroutine of the viral code.
- Add-on viruses: Most viruses are add-on viruses. They function by appending their code to the host code. The add-on virus then alters the startup information of the program, executing the viral code before the code for the main program.
- Intrusive viruses - Intrusive viruses operate by overwriting some or all of the original host code with viral code. When large parts of the host program are completely replaced by the viral code, the original program can no longer function properly.

Another form of classification is related to how viruses are activated and select new targets for alteration. The simplest viruses are those that run when their "host" program is run, select a target program to modify, and then transfer control to the host. These viruses are "transient" or "direct" viruses, known as such because they operate only for a short time, and they go directly to disk to seek out programs to infect.

The most "successful" PC viruses to date exploit a variety of techniques to remain resident in memory once their code has been executed and their host program has terminated. This implies that, once a single infected program has been run, the virus can spread to any or all programs in the system. This spreading occurs during the entire work session (until the system is rebooted to clear the virus from memory). These viruses are "resident" or "indirect" viruses, known as such because they stay resident in memory, and indirectly find files to infect as they are referenced by the user. How do "Resident" viruses remain in memory? The answer for personal computers running software such as MS-DOS is that the virus alters the standard interrupts used by DOS and the BIOS.

*Submitted by Doron Chen

The PC uses many interrupts to deal with asynchronous events and to invoke system functions. When an interrupt is raised, the operating system calls a routine, the address of which it finds in a special table known as the "interrupt table". A virus can modify this table so that the interrupt causes viral code (resident in memory) to be executed.

Common viruses wait for a specific triggering condition, and then perform some activity. This activity can be as simple as seeking particular data items in a specific file and changing their values. Often, viruses are destructive, removing files or reformatting entire disks. Many viruses are also faulty and may cause unintended damage. The conditions that trigger viruses can be arbitrarily complex.

Evolution of viruses

Since the first viruses were written, we have seen what may be classified as five "generations" of viruses. This list describes increasing levels of sophistication and complexity represented by computer viruses in the MS-DOS environment.

- First generation : Simple. Viruses that do nothing very significant other than replicate.
- Second generation: Self-recognition. One problem encountered by viruses is that of repeated infection of host, leading to depleted memory and early detection. To prevent this unnecessary growth of infected files, second-generation viruses usually implant a unique signature that signals that the file or system is infected. The virus will check for this signature before attempting infection; if the signature is present, the virus will not reinfect the host.
- Third generation : Stealth. Most viruses may be identified on a contaminated system by means of scanning the secondary storage and searching for a pattern of data unique to each viruses. To counteract such scans, some resident viruses intercept selected system calls which would expose the presence of the virus and return false information.
- Fourth generation : Armored. As anti-virus researchers have developed tools to analyze new viruses and craft defenses, virus authors have turned to obfuscate the code of their viruses. This "armoring" includes adding confusing and unnecessary code to make it more difficult to analyze the virus code. The defenses may also take the form of directed attacks against anti-virus software, if present on the affected system.
- Fifth generation : Polymorphic. These are viruses that infect their targets with a modified or encrypted version of themselves. The virus in the altered file will not be identifiable through the use of simple byte matching.

Defense Against Viruses

- Activity monitors - programs that are resident on the system, monitor activity, and either raise a warning or take special action in the event of suspicious activity. Thus, attempts to alter the interrupt tables in memory, or to rewrite the boot sector would be intercepted by such monitors.
- Scanners - A scanner operates by reading data from disk and applying pattern matching operations against a list of known viruses patterns.
- Integrity checkers - programs that generate checkcodes for monitoring files. Periodically, these checkcodes are recomputed and compared against saved versions. If the comparison fails, a change is known to have occurred to the file, and it is flagged for further investigation.

Viruses as Artificial Life

Here is a reasonable list of properties associated with life, and an analysis to determine whether computer viruses conform to these properties.

- Life is a pattern in space-time : Viruses are represented by patterns of computer instructions that exist over time on many computer systems. It is questionable if these patterns exist in space, unless one extends the definition of space to "cyberspace".
- Self-reproduction : One of the primary characteristics of computer viruses is their ability to reproduce themselves.
- Information storage of a self-representation : The code that defines the virus is a template that is used by the virus to replicate itself. This is similar to the DNA molecules of what we recognize as organic life.
- Metabolism : This property involves the organism taking in energy or matter from the environment and using it for its own activity. Computer viruses make use of the electrical energy present in the computer to traverse their pattern of instructions and infect other programs. However, the expenditure of energy is not by the virus, but by the underlying computer system. Thus, we must conclude that viruses do not actually have a metabolism.
- Functional interaction with the environment : Viruses alter their environment to support their existence.
- Interdependence of parts : Living organisms cannot be arbitrarily divided without destroying them. The same is true for computer viruses. However, the virus can be reassembled later and regain its functional status. In this sense, computer viruses are more like simple machines rather than instances of living things.
- Stability under perturbations : Computer viruses run on a variety of machines under different operating systems. Many of them are able to compromise (and defeat) anti-virus and copy protection mechanisms.
- The ability to evolve : No computer viruses evolve as we commonly use the term. There are variants of many known viruses. However, the source of these variations appears to be programmers (the original virus authors or otherwise) who alter the viruses to avoid anti-viral mechanisms, or to cause different kinds of damage.
- Growth : Viruses certainly do exhibit a form of growth, in the sense that there are more of them in a given environment over time.

Concluding Comments

Our study of computer viruses at first suggests they are close to what we might define as "artificial life". However, upon closer examination, a number of significant deficiencies can be found. These lead us to conclude that computer viruses are not "alive", nor is it possible to refine them so as to make them "alive", without drastically altering our definition of "life". To suggest that computer viruses are alive also implies that some part of their environment - the computers, programs, or operating systems - also represent artificial life.

Computer viruses provide an interesting means of modeling life. Research into competition among computer viruses and other software, including anti-viral techniques, is of practical interest as well as scientific interest. The problem with research on computer viruses is their threat. Experience shows that viruses cannot be recalled or controlled, and they may cause extensive losses over the years.

Even viruses written to be benign cause significant damage because of unexpected interactions and bugs. Ultimately, it would be disappointing if there would be a widespread acceptance of computer viruses as a form of artificial life. It would be especially dangerous to attract the untrained, the careless, and the uncaring to produce them. Self-reproducing code is easier to write than to control, and encouraging its production in uncontrolled environments is irresponsible.

In conclusion, Spafford suggests that there is something of great importance to be learned from the study of computer viruses: the realization that experimentation with systems in some ways (almost) alive can be hazardous.