VIRUSES : HOW THEY WORK

Echo Azhar

The first use of the term virus to refer to unwanted computer code occurred in 1972 in a science fiction novel, *'When Harvey was one'*, by David Gerald. Fred Cohen, a graduate sutdent at the University of Southern California, formally defined the term computer virus in 1983. The idea of writing a 'computer virus' occurred to him, and in a week's time he put together a simple virus that he demonstrated to the class. His advisor Professor Len Adelman, suggested that he call his creation a computer virus.

A virus is a segment of program code that implants itself to one of your executable files and spreads systematically from one file to another. It is a segment of machine code (typically 200—4000 bytes) that will copy its code 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.

Viruses cannot spread by infecting pure data: pure data is not executed, however, some data, such as files with spreadsheet input or text files for editing, may be interupted by application programs. For instance, text files may contain special sequences of characters that are executed as editor commands when the file is first read into the editor. Under these circumstances, the data are "executed" and may spread a virus. Data files may also contain "hidden" code that is executed when the data is used by an application, and this too may be infected .

What is the function of a virus ?

Usually a virus has two distinct functions:

1) Spreads itself from one file to another without your input or knowledge. Technically, this is known as self-replication and propagation.

2) Implements the symptom or damage planned by the perpetrator. This could include erasing a disk, corrupting your programs or just creating havoc on your computer. Technically, this is known as the virus payload which can be benign or malignant at the whim of the virus creator.

A *benign virus* is one that is designed to do no real damage to your computer. For example, a virus that conceals itself until some predetermined date or time and then does nothing more than display some sort of message is considered benign. A *malignant virus* is one that attempts to inflict malicious damage to your computer, although the damage may not be intentional. There are a significant number of viruses that cause damage due to poor programming and outright bugs in the viral code. A malicious virus might alter one or more of your programs so that it does not work as it should. The infected program might terminate abnormally, write incorrect information into your documents. Or, the virus might alter the directory information on one of your system area. This might prevent the partition from mounting, or you might not be able to launch one or more programs, or programs might not be able to locate the documents you want to open.

Structure of a virus :

A virus may add itself to host code : as a shell, as an add-on and as intrusive code.

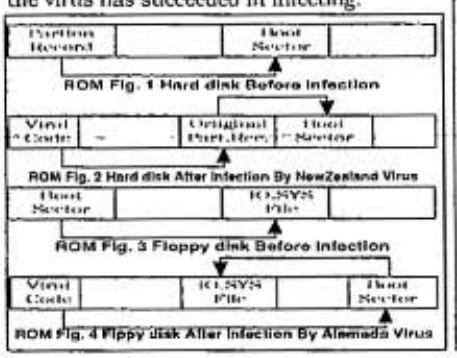
a) Shell viruses : A shell virus is one that forms a " Shell" around the original code. In effect, the virus becomes the program, and the original host program becomes an internal subroutine of the viral code. An extreme example of this would be a case where the virus moves the original code to a new location and takes on its identity. When the virus has finished executing, it retrieves the host program code and begins its execution.

Add-on Viruses : Most viruses are add-on viruses. They function by appending their code to the end of the host code, or by relocating the host code and adding their own code to the beginning. The add-on virus then alters the startup information of the program, executing the viral code before the code for the main program. The host code is left almost completely untouched; the only visible indication that a virus is present is that the file grows larger.

Intrusive Viruses : Intrusive viruses operate by replacing some or all of the original host code with viral code. The replacement might be selective, as in replacing a subroutine with the virus, or inserting a new interrupt vector and routine. The replacement may also be extensive, as when large portions of the host program are completely replaced by the viral code.

Once a virus has infected a program, it seeks to spread itself to other programs and eventually to other systems. Simple viruses wait for a specific triggering condition, and then perform some activity. The activity can be as simple as printing a message to the user, or as complex as seeking particular data items in a specific file and changing their values. Often viruses are destructive, removing files or reformatting entire disks.

The conditions that trigger viruses can be arbitrarily complex. This includes waiting for a specific date or time, determining the presence or absence of a specific set of files (or their contents), examining user keystrokes for a sequence of input, examining display memory for a specific pattern, or checking file attributes for modification and permission information. Viruses also may be triggered based on some random event. One common trigger component is a counter used to determine how many additional pro-grams the virus has succeeded in infecting.



How do they Spread?

Computer viruses can infect any form of writable storage, including hard disk, floppy disk, tape, optical media, or memory. Infections can spread when a computer is booted from an infected disk, or when an infected program is run. It is important to realize that often the chain of infection can be complex and convoluted. A possible infection might spread in the following way:

- A client brings in a diskette with a program that is malfunctioning (because of a viral infection).
- A consultant runs the program to discover the cause of the bug and then the virus spreads into the memory of the consultant's computer.
- A consultant copies the program to another disk for later investigation and then the virus infects the copy utility on the hard disk.
- A consultant moves on to other work preparing a letter and then the virus infects the screen editor on the hard disk.
- The system is switched off and rebooted the next day and then the virus is cleared from memory, only to be reinstalled when either the screen editor across a network link, thus infecting their own system.

How a virus becomes active :

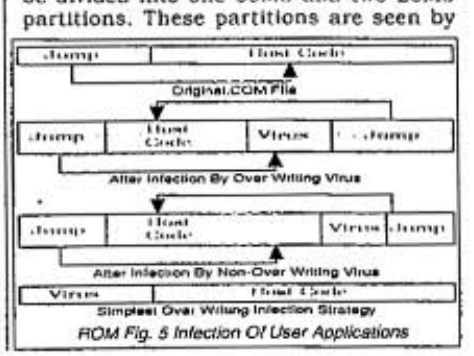
Let us first consider the IBM PC boot sequence. It has six components:

- *ROM BIOS routines*
- *Partition record code execution*
- *Boot sector code execution*
- *IO.SYS and MSDOS. SYS code execution*
- *COMMAND.COM command shell execution*
- *AUTOEXEC.BAT bauch file execution*

ROM BIOS : When an IBM PC or compatible PC is booted, the machine executes a set of routines in ROM These routines initialize the hardware and provide a basic set of input/output routines that can be used to access the disks, screen, and keyboard of the system. These routines constitute the basic input output system (BIOS).

ROM routines cannot be infected by viral code (except at the manunfacturing stage), since they are present in read-only memory that cannot be modified by software.

Partition record : The IBM PC disk operating system (DOS) allows a hard disk unit to be divided into upto four logical partitions. Thus, a 100Mb hard disk could be divided into one 60Mb and two 20Mb partitions. These partitions are seen by



ROM Fig. 5 Infection Of User Applications

DOS as separate derives: "C," "D," and so on.The ROM code executes a block of code stored at a well-known location on the hard disk (head 0, track 0, sector 1). This is the first physical sector of every hard disk that contains the disk's *Master Boot Record* and Partition Table. The Master Boot Record (MBR) has a small program within it called the Master Boot Program which looks up the values in the partition table for the starting location of the bootable partition, and then tells the system to go there and execute any code it finds. The size of each partition is stored in the partition record, as is a block of code responsible for locating a boot block on one of the logical partitions.

The partition record code can be infected by a virus, but the code block is only 446 bytes in length. Thus, a common approach is to hide the original partition record at a known location on the disk, and then to chain to this sector from the viral code in the partition record. This is the technique used by the *New Zealand* virus, discovered in 1988 (Fig. 2).

Boot sectors : The partition record code locates the first sector on the logical partition, known as the boot sector. (If a floppy disk is inserted, the ROM will execute the code in its boot sector, head 0, track 0, sector 1). Every logical drive, both hard disk and floppy, contains the boot sector. This sector contains specific information relating to the formatting of the disk, the data stored there and also contains a small program called the boot program (which loads the DOS system files). The boot program displays the familiar *"Non-system Disk or Disk Error"* message if the DOS system files are not present. It is also the program that gets infected by viruses. You get a boot sector virus by leaving an infected diskette in a drive and rebooting the machine. When the boot sector program is read and executed, the virus goes into memory and infects your hard drive. Remember, because every disk has a boot sector, it is possible (and common) to infect a machine from a data disk.

The boot sector contains a BIOS Parameter Block (BPB). The BPB contains detailed information on the layout of the filing system on disk, as well as code to locate the file IO.SYS. That file contains the next stage in the boot sequence (Fig. 3).

A common use of the boot sector is to execute an application program, such as a game, automatically; unfortunately, this can include automatic initiation of a virus. Thus, the boot sector is a common target for infection. Available space in the boot sector is too limited (a little over 460 bytes is available). Hence, the technique of relocating the original boot sector while filling the first sector with viral code is also used. A typical example of such a 'boot sector' virus is the *Alameda* virus. This virus relocates the original boot sector to track 39, sector 8, and replaces it with its own viral code (Fig. 4).

Other well-known boot sector viruses include *Brain (Pakistani), Search,* and *Italian* viruses. Boot sector viruses are particularly dangerous because they capture control of the computer system early in the boot sequence, before any anti-viral utility becomes active.

MSDOS.SYS, IO.SYS : The boot sector next loads the IO. SYS file, which carries out further system initialization, then loads the DOS system contained in the MSDOS.SYS file. Both these files could be subject to viral infection.

Command shell : The MSDOS.SYS code next executes the command shell program (COMMAND.COM). This program provides the interface with the user, allowing execution of commands from the key-board. The COMMAND.COM program can be infected, as can any other .COM or .EXE executable binary file. The COMMAND.COM file is the specific target of the Lehigh virus that struck Lehigh University in November 1987. This virus caused corruption of hard disks after it had spread to four additional COMMAND. COM files.

AUTOEXEC batch files : The COMMAND.COM program is next in the boot sequence. It executes a list of commands stored in the AUTOEXEC. BAT file. This is simply a text file full of commands to be executed by the command interpreter. A virus could modify this file to include execution of itself.

How do they infect a user program :

To infect a code file, the virus must insert its code in such a way that it is executed before its infected host program. These viruses come in two forms:

a) Overwriting : The virus writes its code directly over the host program, destroying part or all of its code. The host Program will no longer execute correctly after infection.

b) Non-overwriting : The virus relocates the host code, so that the code is intact and the host program can execute normally.

A common approach used for .COM files is to exploit the fact that many of them contain a jump to the start of the executable code. The virus may infect the programs by storing this jump, and then replacing it with a jump to its own code. When the infected program is run, the virus code is executed. When the virus finishes, it jumps to the start of the program's original code using the stored jump address (Fig. 5).

Notice that in the case of the overwriting virus, the more complex infection strategy often means that all but a small block of the original program is intact. This means that the original program can be started, although often it will exhibit sporadic errors or abnormal behavior.

Talking about others :

Memory Resident Viruses : The most 'successful' 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 potentially 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), rather than during a small period of time when the infected program is executing viral code. Thus, the two categories of memory-resident virus are:

a) Transient : The viral code is active only when the infected portion of the host program is being executed.

b) Resident : The virus copies itself into a block of memory and arranges to remain active after the host program has terminated. The viruses are also known as TSR (Terminate and Stay Resident) viruses. Examples of memory resident viruses are all known boot sector viruses, the *Israeli, Cascade,* and *Traceback* viruses.

File Infectors : These are viruses that attach themselves to (or replace) .COM and .EXE files, although in some cases they can infect files with extensions .SYS, .DRV, .BIN, .OVL and .OVY. With this type of virus, uninfected programs usually become infected when they are executed with the virus in memory. In other cases they are infected when they are opened (such as using the DOS DIR command) or the virus simply infects all of the files in the directory it was run from (a direct infector).

How to recognaze a viral infection

A common symptom is a sudden change in the size of programs or files, or a sudden decrease in the amount of space available on your disks. This is caused as the viral code is copied into program files and to disk. A sudden increase in the number of sectors marked unusable or bad may indicate a virus that hides itself on disk. A reduction in a available physical memory may signal the presence of a TSR virus.

A second common symptom of infection is odd behavior of system services. Resident viruses may not pass along system service requests correctly, or may alter those requests for their own purposes, thus leading to faulty behavior. Lost or garbled output to the screen or printers, corrupted images on the screen or access to the disks that faild may signal a TSR virus they also may signal a hardware problem or software bug. A system that suddenly seems slower may also signal the presence of a virus that is trapping service interrupts.

Since a virus needs to access disk to copy itself and to find new hosts to infect, excess or oddly-timed disk accesses can signal a viral infection. Newer viruses are more sophisticated in this regard as they piggyback their accesses on other, legitimate accesses.

A fourth and obvious symptom of a viral infection is the failure of some or all of your programs to work normally. This occurs when the viral code overwrites your application, or when it botches the jumps or code changes necessary to infect the code. In particular, if your code behaves differently from machine to machine, or from hard disk to diskette, you should suspect a virus.

References :

1) Web sites:
http://www.symantec.com/avcenter
http://www.netwell.com/~silicon/episteme.html
2 A.K.Dewdeney 1985.'A Core War bestiary of viruses,worms and other threates to computer memories',Scientific Americana.
3) Eugene h Spafford, Kathleen A. Heaphy, David J.Ferbrache.1989.' Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats'. ADAPSO.
4) P.J. Deanning.1988. Computer Viruses. Am. Sci.