Modern Methods of Detecting and Eradicating Known and Unknown Viruses

Dmitry Mostovoy
5th international conference Virus Bulletin - 95
September 1995

Abstract

Viruses are growing in number from day to day, so it is obvious that soon anti-virus programs like NAV or MSAV will not be quite efficacious. Therefore, we started designing a program that would annihilate not individual infectors, but viruses in general, regardless of whether a virus is known or not, or whether it is old or new.

The first outcome of our efforts in this direction, ADinf (Advanced Diskinfoscope), is a forecasting center which alerts the user in advance with great reliability about the intrusion of viruses, even HITHERTO unknown infectors. As distinct from all other data integrity checkers, ADinf inspects a disk by scanning the sectors one by one via direct addressing of BIOS without the assistance of the operating system and takes under check all vital parts of hard disk. To evade such a detection tactics is almost impossible.

ADinf alerts the user in time about virus intrusion and restores infected boot sectors. How to restore the infected files automatically? Our next step was to produce a curing companion to ADinf. The new tool, ADinf Cure Module, deploys a novel strategy. Paradoxically, ninety seven percents of the viruses in our collection fall under few standard groups by the types of infection methods. New viruses are as a rule designed on one of these common infection principles, and therefore ADinf Cure Module will be about 97% efficient in its performance also in the future.

ADinf and ADinf Cure Module are parts of DialogueScience anti-virus kit - the most popular anti-virus in Russia.

Integrity Checking

The basic classes of anti-virus programs are well known. They are scanners/removers, monitors, and vaccines. I would like to discuss the development of programs to which, in my opinion, anti-virus designer pay undeservedly little attention. This class of anti-virus programs is known as ``integrity checkers'', though the name does not fully characterize the program's policy which we describe below. This is the only class of purely software means of anti-virus protection, which permits the detection of known and unknown viruses with reliability approaching 100% and eradication up to 97% file infectors, even new hitherto unknown viruses.

The operation of integrity checkers is based on a simple fact: even though it is impossible to know all information about potentially infinite number of viruses, it is quite possible to store a finite volume of information about each logical drive in the disk and to detect virus infection from the changes taken place in files and system areas of the disk. As already mentioned, the name "integrity checker" does not fully reflect the essence of these programs. Infection techniques is not restricted to a simple modification of the program code. Other paths for infection either

already exist or are also possible; for example, companion viruses [1]. A disk can be corrupted by restructuring the directory tree, say, by renaming the directories and creating new directories, and by other such manipulations. Consequently, to provide reliable protection integrity checkers must take care of far more number of parameters that the mere changes in the size and CRC of files as is done by most programs of this class. Thus, master boot record (MBR) and boot sectors of logical drives, a list of bad clusters, directory tree structure, free memory size, CRC of Int 13h handler in BIOS and even the Hard Disk Parameter Tables, all must be under the control of integrity checkers. Changes in the size and CRC of files, creation of new files and directories and removal of old files and directories are obviously objects for strict control. A designer of integrity checker must be one step ahead of virus designers and block every possible loophole for parasite intrusion.

Despite the large amount of controlled information, an integrity checker must nonetheless be user-friendly, simple in usage, and quick in checking disks. It must at the same time be user-customizable as regards the levels of messages displayed on the changes occurred in the disk and be capable of conducting a preliminary analysis of the changes, particularly the suspicious modifications such as

- changes in size and CRC of files without any change in datestamp,
- illegal values of hours, minutes or seconds in the datestamp of infected files (for example, 62 seconds),
- year greater than the current year (certain viruses mark infected files by increasing the year of creation by 100 years, which cannot be detected visually because ``dir'' command only displays the last two figures of the year,
- any changes in files specified in the ``stable'' list,
- change in master boot record or boot sector,
- appearance of new bad clusters on the disk and others.

Let us now discuss the main problems faced by a designers of ``integrity checkers''. First, this is the dodging ability of viruses based on stealth-mechanism. Integrity checkers that rely on operating system tools in their scanning mission are absolutely helpless against this class of viruses. They have stimulated the development of an integrity checker that checks disks by reading the sectors via direct addressing through BIOS. Stealth viruses cannot hide the changes in an infected file size; on the contrary, under such a scanning technique the stealth-mechanism betrays the presence of known and hitherto unknown stealth viruses through the discrepancy between the information given out by DOS and the information obtained by reading via BIOS. Such algorithms have been created and successfully detect the appearance of stealth-viruses.

Scanning a disk by reading the sectors by direct addressing of BIOS has one more important merit which is often overlooked. If a computer is infected by a so-called ``fast infector'' [1], i.e., a virus that infects files not only when they are started, but also when they opened, such an integrity checker will not spread the infection to all files in the disk, because it does not at all address the operating system for reading a disk via sectors and uses an independent file opening system, and the viruses does not get any control.

Finally, an integrity checker utilizing direct reading of sectors is twice faster in checking a disk than any other program than relies on the operating system tools, because a disk scan algorithm can be created that reads each sector only once and optimizes the head movements.

Disk handling via BIOS has its own hurdles. The foremost problem is the compatibility with innumerable number of diverse hardware and software, including disk compactors (Stacker, DoubleSpace), specialized drivers for accessing large disks (Disk Manager), SCSI disk drivers etc. Furthermore, there are

many MS-DOS compatible operating systems that have imperceptible but quite important features in partitioning logical drives. Integrity checkers must pay due attention to these fine factors.

Virus Removal Techniques

Modern integrity checkers are useful not only in detecting infection, but are also capable of removing viruses immediately with the help of the information they retrieve from an uninfected machine at the time of installation. An integrity checkers can kill known viruses as well as the viruses which were unknown at the time of creation of the integrity checker.

How this is done? Obvious are the methods for removing viruses from the master boot record and boot sectors. Integrity checker stores images of uninfected boot sectors in its tables and in case of damage can instantly restore them. The only restriction is the restoration must also be effected via direct addressing of BIOS and after restoration the system must be rebooted immediately in order to prevent the active virus from reinjecting infection while accessing the disk via INT 13h.

Removal of file viruses is based on a surprising fact, namely, despite the vast number of diverse viruses, there are only a few techniques by which a virus is injected into a file. Here we only briefly outline the file restoration strategy. Figure 1 shows a schematic diagram of a usual EXE file.

For each file integrity checker keeps a header (area 1), relocation table (area 2) and the code at the entry point (area 4). Strings (area 3 and area 5) are vital because they are the keys to identifying the mutual locations of various areas in an infected file when a virus writes its tail, not at the file end, but at the file beginning or in the file body (after the relocation table or at the entry point). In an infected file, after determining the area that coincides with the imaged areas in the table, the displacement of a block (for example, the block for area 3 begins at the end of area 2 and ends at the beginning of the area 4) can be identified by string 3 position and thus moved back to its original location.

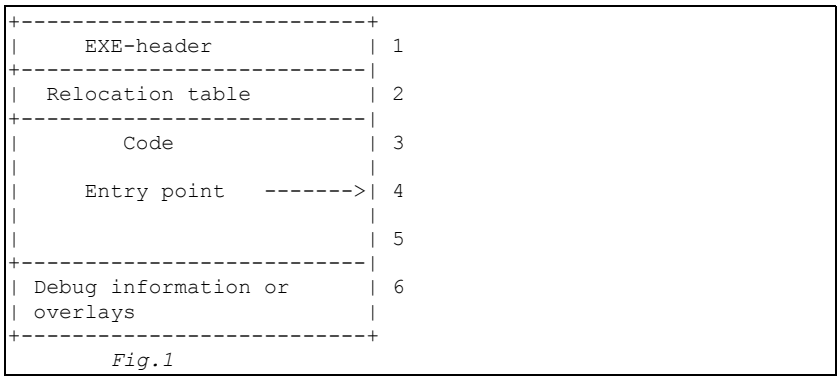

Fig.1

Image of area 6 takes about 3-4 Kb and is essential in recovering a file corrupted by viruses which damage the debug information and overlays in the course of defective infection.

Thus, a file is recovered by reinstating its original status overwriting the image of its structure stored in integrity checker tables on an infected file. Consequently, a knowledge as to which virus infected the file is not mandatory.

Tables containing information necessary for recovering files take about 200-450 Kb for one logical drive. The table size can be cut down to 90 Kb, if a user decides not to save the relocation information and this will not have any perceptible influence on the quality of recovery in most of the cases.

Conclusion

Integrity checkers undoubtedly do not provide a panacea against computer viruses. Unfortunately, there is no such panacea, nor can there be one. But they are quite reliable protection utilities which must be used jointly with other classes of anti-virus tools. The highlights of integrity checkers described above are all implemented in ADinf program, the most popular itegrity checker in Russia. It also is known in Germany where it is distributed on CD-ROM as a component of the DialogueScience Anti-Virus Kit. It checks a disk by reading its sectors one by one directly addressing BIOS, easily traps active stealth viruses by comparing the information obtained through BIOS and DOS. It instantly restores up to 97% of files corrupted by known and unknown viruses.

References

1. Vesselin Bontchev, Possible Virus Attacks Against Integrity Programs And How To Prevent Them, Proc. 2nd Int. Virus Bulletin Conf., September 1992, pp. 131-141.

2. Mostovoy D. Yu., A Method of Detecting and Eradicating Known and Unknown Viruses, IFIP Transactions, A-43, Security&Control of Information Technology i Society, February, 1994, pp. 109-111.