PROGRAMMING AND COMPUTER SOFTWARE

A translation of *Programmirovanie*

March, 1991

Volume 16, Number 3 **May-June, 1990**

CONTENTS

The Russian press date (podpisano k pechati) of this issue was 5/21/1990. Publication therefore did not occur prior to this date, but must be assumed to have taken place reasonably soon thereafter.

CLASSIFICATION OF COMPUTER VIRUSES IN MS DOS

N. N. Bezrukov

UDC 681.3

This paper presents a brief history of the appearance and general operating principles of viruses, along with a proposed scheme for classification; the properties of the most widespread (as of December 1989) viruses infecting MS DOS are tabulated. Use of the proposed taxonomy by those involved in combatting viruses permits considerable simplification of the tasks of gathering and distributing knowledge in the field.

Computer viruses are programs that are capable of secret reproduction in the environment of standard operating systems by including themselves in either directly or indirectly used code (nonresident code and operating system components), the copies (offspring), possibly modified, retaining the ability to reproduce [1].

After infecting a program, viruses may be able to propagate from one program to another. An infected program or copy may be transferred by a diskette or network to other machines. Since exchange and transmission of data is widespread among users of personal computers, there may be a considerable number of infected programs. The use of a single personal computer by several users also aids propagation, especially when the PC has a winchester. From the viewpoint of computer viruses, a particularly strong danger is presented by game players, who usually have little knowledge of system internals and are not fully aware of the consequence of their actions.

In brief, the process whereby a virus infects program files can be described as follows. An infected program contains code that has been modified so that the virus gains control prior to the virus carrier. Once control has been transferred to the virus, it locates a new program and inserts a copy of itself either and the beginning or end of the (usually not yet infected) victim program. If the virus code is appended to the end of the victim, the victim code is modified to transfer control to the virus before the victim gains control. After the virus code has been executed, control is returned to the victim, which then proceeds normally. It is rare that viruses appear in the middle of victim code.

Programs that have been infected by a virus may be treated as examples of Trojan Horse programs [2], which contain secret modules (the body of the virus code) that perform unsanctioned operations (infecting other programs). In addition to causing infection, a virus may also perform other unsanctioned operations whose effects range from totally innocuous to extremely destructive. This makes viruses one of the most dangerous forms of computer vandalism.

The notion of self-replicating programs has quite a history. As early as 1951, John von Neumann proposed a method for creating self-replicating mechanisms. This problem was the subject of a number of publications, among which we should note L. S. Penrose's 1959 article on self-reproducing machines. On the basis of this article, F. G. Stahl, programming in IBM 650 machine language, developed a cybernetic model in which "life" proceeded by writing nonzero words, dying if it was unable to write for a sufficiently long time, and reproducing after eating a sufficient number of words [3]. Mutations could suppress viability to the point of preventing reproduction.

In 1962, V. Vissotsky of Bell Laboratories developed the game "Darwin," in which several assembly language programs (called organisms) reproduced within computer memory. The organisms created by one player had to "kill" representatives of other types of organism and multiply to occupy the "living space" [4]. The winner of the game was the player whose organisms occupied the most memory or had the largest number of points. The game "Animal" should be included among the early viruses. In this game, the program asks questions to determine which animal is being thought of by the player. When the machine provides an incorrect answer, it asks the player to provide a question that would inform it of the characteristics of the animal being thought of. In remembering these questions, the program not only modified itself, but inserted a copy of the updated code

Translated from Programmirovanie, No. 3, pp. 3-22, May-June, 1990. Original article submitted January 22, 1990.

into another catalog. If there was a copy of "Animal" already there, it was replaced. As the user moved from one machine to another, it was transferred in the user's catalog. It did not take long for every catalog in the file system to have a copy of "Animal." Because multiplying copies of "Animal" were occupying considerable amounts of disk space, a "more infectious" version of the program was written to copy itself not once, but twice. Upon reaching a particular line, it offered the user one last chance to play, and then erased itself from the disk [5].

In 1975, John Brunner's best seller "The Shockware Rider" described a program worm that spread throughout a network – it was an idea that anticipated later events, but it was beyond the capabilities of the machines at that time. As early as 1982, however, a worm was created at the Xerox research center that could propagate throughout a network and seize control of machines in the network. During an experiment with Ethernet, the worm was released and uncontrolled propagation was observed, and some of the computers were seen to be looping.

In 1984 a simplified version of "Darwin" was published under the name "Core Wars" [8]. In this game two players execute a single program located in a section of memory (called the arena) with wraparound addressing. The players alternately execute a single instruction, which makes the program a rather simple real time system. The players' programs attack each other and simultaneously attempt to prevent and repair damage. A simple attack is provided by execution of the MOV instruction (write to memory). For example, MOV #0, 1000 might annihilate an opponent if the next command is located immediately after location 1000, it might damage data program code, or it might simply hit garbage. Publication of this game served as a catalyst, and stimulated two Italian readers to develop a rather complete model of a bootstrap virus.

September of that same year saw publications of the first academic investigation of viruses by F. Cohen [1].

A new stage in the evolution of viruses began with the advent of personal computers. The first cases of massive infection appeared in 1987 when the Lehigh virus infected several hundred diskettes in the course of a few days [2]. On December 30, 1987 a virus was discovered at a Jerusalem university (in Israel) [2]; this virus rapidly spread throughout Europe and the United States, and appeared in the USSR at the end of 1988. The so-called "Morris virus" drew a great deal of attention in 1988 [9]. On November 2, 1988, R. Morris, a doctoral candidate at Cornel University used a virus he created to infiltrate a large number of machines (informed estimates were of the order of 6000) attached to the American national network Internet. Although there was no damage done, thousands of hours of machine time were lost to users.

A major problem presented by the appearance of computer viruses is that of protecting programs and data. The problem is completely solvable, and there is a whole group of effective measures – technical, organizational, and juridical – to protect against damage or destruction of data.

The first action of a computer virus upon gaining control is replication. Following reproduction, most computer viruses then go through a "symptomatic" phase in which, frequently accompanied by visual or auditory effects, the file system is damaged. Symptoms may alternate with replication, after a specific "incubation" time, or they may appear as the result of encountering a set of specific conditions.

It should be noted that a virus may have a specific predetermined latency period, during the course of which no action is taken to replicate or cause symptoms to appear. The latency phase might result from a preprogrammed delay (of months or years), machine configuration (for example, the virus might be activated only upon encountering a specific configuration (e.g., the virus might become active only upon encountering a winchester), or activation might be dependent on special characteristics of equipment (for example, IBM PC clones).

Computer viruses are immortal and can be preserved in various forms of archives, and, after the passage of more or less time, may become reactivated months or even years after the initial infection. Thus, after the first observation and elimination of a virus, repeat infections are to be expected, i.e., special measures must be taken to guard against repeat infection: archived material must have checksums verified, controls must be imposed on incoming programs, and antiviral programs must be installed to make infiltration by viruses difficult (filters) or impossible (vaccines).

The symptoms of viruses may be classified by means of the following basic categories: failure to complete a function (such as prevention of loading programs from write protected disks); performance of operations not intended by the programmer (such as formatting a disk, deleting files, or presenting false, irritating, or misleading messages such as the "falling letters" of the RS-1701 virus, the slowdown caused by the RSE1813 virus, the song played by the RSE-1805 virus, or the moving rhomboids of the Vx1-1S virus, etc.).

The damage done by a virus may be catastrophic (for example, an entire winchester might be wiped out), an action usually associated with a long incubation period, during which the virus only replicates. Or the virus might frequently make small changes that are difficult to observe. Illegal operations performed by viruses may be triggered by the appearance of certain data, completion of a certain number of replications, or the occurrence of certain conditions, such as writing an infected program on a winchester. In this latter case, the combination of conditions may be complex enough to make detection difficult.

The must vulnerable part of the DOS file system is the FAT (the file allocation table). If the FAT is damaged, DOS is unable to locate files, even though the data itself may be undamaged. A virus might also format disk areas containing system data. The most damage is caused not by catastrophic destruction of the FAT, but by small, insignificant changes to file data. For example, one resident virus swaps the first two bytes of each block when DBF files are written. On the disk, the file version of the table is damaged, but it is repaired when it is read. The damage only becomes apparent when the file is moved to another machine or the virus is eliminated.

Although the majority of the dangers presented by viruses pertain to data, it is possible for equipment to be damaged. For example, it is possible to damage the phosphor ("burn a hole") of a monochrome monitor by using the control system. The author does not know of any cases in which this has occurred, but legends of such events have become firmly entrenched in programmers' folklore.

Since the scientific interests of the author in the recent past have directed toward development of effective means of disassembly, decompilation, and reconstruction of program assurance data [10, 11], viruses as programs specially developed to test disassembly methods immediately drew the author's attention as one possible proving ground for the development projects in which he was involved. As part of his research, the author has completely disassembled and reconstructed the initial text of a number of computer viruses [12]. In addition to being disassembled and studied under static conditions, the viruses have had their operation traced and their behavior studied under controlled conditions. The author's research led to the conclusion that it is necessary to develop a standard taxonomy for MS DOS viruses.

At the present time, the basic approach to classification is the use of informal names and nicknames. Examination of the informal names that have been selected indicates that there are three basic approaches to selection. The first is based on the location of detection (Lehigh, Jerusalem, Vienna, Alameda), the second is based on displays or messages for which the virus is responsible (Vacsina, Eddie, Dark Avenger, Disk Killer, sUMsDos), and the third path is naming viruses after their effects (Time Bomb, DOS-62, Cascade, Black Friday, Bouncing Dot). As a result, the same virus might be named differently by different researches involved in antivirus development, and there is no assurance new names always correspond to new viruses.

This historically developed approach leads to a number of undesirable effects. First, every antiviral researcher uses a unique classification, which frequently makes it difficult to determine which virus they are dealing with (for example, the authors of the phage FAG_OM use the name "Omega" for the Viennese virus "Vienna," which, of course, testifies to the inventiveness of the authors, but says nothing to potential users), all the more so because one basic virus usually appears in a variety of versions with similar, but not identical, properties. Second, users tend to estimate the total number of viruses by the number of available antiviral tools, especially phages (i.e., programs that remove the virus from infected programs, thus returning them to close to their initial condition). Such an approximation leads to a substantial overestimate of the total number of existing computer viruses, although people rapidly "rationalize" this fact by separating a single real virus into several "virtual viruses," assigning to each a set of characteristics. Thus, the author has encountered a "home made" classification in which the viruses C-648 and RCE-1813 each had two "faces," and the second face of C-648 had the features of RCE-1813 (an execution slowdown).

As in other fields, the development and use of a standard taxonomy greatly simplifies the acquisition and propagation of knowledge. In particular, it assists in the solution of the important problem of uniquely determining whether a recently detected virus is new or a modification of an old one, and it facilitates selection of defensive tactics (which are very frequently developed ad hoc to meet the needs of the unique problems of a particular organization and are not documented, although such approaches subsequently spread throughout the county and users experimentally determine their applicability in new situations).

The fundamental requirement presented to a taxonomy is objectivity, i.e., a classification system must be based on a fixed selection of relatively simple and incontrovertible tests that do not require profound analysis of infected programs and elements of the operation system. The author has developed an approach based on three fundamental elements:

a code (somewhat reminiscent of the scheme for classifying transistors)

a descriptor (formalized list of basic characteristics)

a signature (line for concrete search for a given virus in an infected program)

Classification Code. In the proposed scheme each virus is assigned a code consisting of an alphabetic prefix, a numeric root (characteristic) and an optional alphabetic suffix. For example, in the code RC-1701f the RC is the prefix, 1701 is the root (characteristic), and f is the suffix.

The principal requirement presented to the code is that it make it possible to identify the majority of the virus' properties on a computer that is not infected by a resident virus. Execution of various operations to investigate a virus on an infected machine is the most common and profound error that can be committed by inexperienced users. It must be emphasized that any operation by a machine infected by an unidentified virus is associated with the risk of activating a Trojan component. In addition, a resident virus may intercept inquiries and present distorted replies in order to hide itself (there are two known viruses with this ability).

The *prefix* characterizes the means of propagation and consists of an alphanumeric string beginning with a capital letter. Five basic propagation paths can be distinguished:

file (virus infecting the COM, EXE, and OVL files — prefixes C, E, and CE);

boot (virus infecting the bootstrap sector of the MBR block — prefixes B, D, or M);

driver (virus infecting the driver facility or inserting itself into the CONFIG.SYS code — prefix S);

packet (virus either written in the job control language of the operating system infected or using this language to propagate itself — prefix J);

compiled (virus inserts code in the input text of some compiled program — prefix T);

A special class of viruses is comprised of the so-called *network viruses* (more accurately, *replicators*), which include logic to assure propagation to all or part of the subscribers to a network. This class includes both the packet viruses (for example, the virus Christmas Tree was written in the rather well known (in the Soviet Union) control language REXX for VM 370, which is widely used for the 360/370 series machines), and compiled viruses (Morris [7] has a compiled component). Only two of the enumerated types of viruses have been known to infect MS DOS — file viruses and boot viruses.

The *characteristic* of a virus represents a quantitatively measurable property that can easily be determined and is distinct for the majority of virus types. For example, for file viruses we can use the change in the length of a file upon infection.

A *suffix* is used when two different viruses or two different variants of the same virus have the same prefix and characteristic. In this case, in order to obtain a unique code we use an alphabetic character. For example, in the code RC-1704f the letter "f" indicates "variant f." The letter "G" is reserved for groups of viruses (see below).

Virus Descriptor. The descriptor is a systematization of the basic properties of the virus in coded form. The code consists of a group of symbols beginning with a capital letter and continuing with an alphanumeric string.Here the capital letter defines the form of the property being described, and the lower case letter or digits indicate the value of the property for the particular virus in question. For example, in the descriptor "XabYcZdmt" there are three properties: X with value ab, Y with value c, and Z with value dmt.

Virus Signature. Since all currently known viruses can be detected by means of a context search, one of the important problems of classification is that of providing text for concrete searches (the signature). Knowledge of the signature makes it possible to screen incoming programs and provide enhanced security. Standardization of signatures is particularly important when a virus has many variants, since the formal scheme provided by the above code and descriptor has the weakness that certain variants may not be distinguished by a given set of tests. At the same time, it is relatively easy to assure uniqueness of the signature, at least for the known virus types, although theoretically it is possible to create a virus without a signature, i.e., that cannot be found by means of a textual search.

Although the following material only considers signatures using text, it is also possible to use regular expressions. These are much more resistant to certain mutations and, therefore, with a shorter search ensure higher quality identification (a smaller number of false identifications). This makes them preferable to lines of text. Some time in the future, the author hopes to publish a version of the appendix to this article with signatures using regular expressions.

It is obvious that a signature corresponding to executable code is preferred to a program section containing data, such as readable text (this section can be modified). It is therefore desirable to choose a signature on the basis of analysis of disassembled code. The length of the signature should not be too great, since it is impossible to remember a long signature, and difficult to input. Similarly, too short a signature or selection of the wrong section of code for the signature may result in many false identifications, which is very undesirable. Proper selection of a signature requires that it not appear in any of the most frequently used MS DOS programs, including, of course, MS DOS components themselves. Thus, choosing a good signature requires a number of experiments, and the signatures themselves are worthy of analysis.

At the present time there are numerous programs for detecting viruses by examining files for corresponding code, and it is natural to use the signatures developed for these programs as a "starting point." The author used code found in two foreign detectors: SCAN, from McAfee Associates (USA), and VIRSCAN, from IBM. For definiteness, we call the text used by SCAN the M-signature, and the text used by VIRSCAN the I-signature. We should note that these programs do not have signatures for a number of viruses known in the Soviet Union (C-534, C-623, C-529, etc.), and the signatures for the TP viruses fail. Where signatures that have been selected by the author are given, they are called B-signatures. Also, the body of certain viruses contains readable characteristic text. We call such text T-signatures and use it for confirmation.

It should be noted that a context search may be used not only for examining programs infected by a virus, but also for finding files that have been destroyed or damaged by a virus. For example, the virus C-648, upon detecting certain timer values, clobbers programs, instead of infecting them, by replacing the first five bytes with code to transfer to the bootstrap routine of BIOS. Virus-clobbered programs can be located by searching for the line "EAF0FF00F0." Similarly, the virus RCE-1800 destroys sectors in a winchester by overwriting data with the message "Eddie lives...somewhere in time." This message can be used by Norton Utilities of PC Tools to find all damaged sectors and the files to which they belong.

When signatures are available, infection of files by a virus of a given type can be checked not only by special programs (of which, in the author's opinion, the best is VL (Virus Locator), which was written by A. Shekhovtsov and permits examination in a directory or its subdirectories), but Norton Utilities or PC Tools, which are always available (all files can be searched by using the global search mode).

CLASSIFICATION OF FILE VIRUSES

Most file viruses have strains — variants that are only slightly different from the base version. It is therefore useful to speak in terms of groups of file viruses, and, accordingly, group descriptors and group signatures. At the present time, the following groups of file viruses have been detected:

The *Vienna virus* (the first representative of this group is the C-648 virus, which appeared in 1987 in Vienna. Its disassembled text was published and distributed on diskettes with antivirus programs, so there have been numerous attempts at modifying it);

the *CASCADE group* (the first representative of this group is RC-1701, which appeared in the middle of 1988);

the *Jerusalem group* (the first representative of this group is RCE-1813, detected at the end of 1987 in Jerusalem);

the *TP group* (these viruses, supposedly created in Bulgaria, have a characteristic termination in which after a common two-byte code, there is a specific 16-bit code distinguishing different versions. In turn, this group splits into three subgroups — the VACSINA (TP-4 through TP-6), the musical overlay subgroup (TP-24, TP-25) and the Samoyed group (TP-34 and above)).

File Virus Code. First, the file viruses must be classified as resident or nonresident, since their residence, to a large extent, determines their behavior (resident viruses are substantially more virulent than nonresident). Thus, we begin the codes for resident viruses with the prefix R, as in RC-1701.

The structures of resident and nonresident viruses are quite different. Resident viruses have an installation phase, during which they are loaded into main memory and, simultaneously, hidden in an attempt to make it difficult to find them in the midst of resident code. This installation phase requires the existence of a special virus section —the installer — that is not required for nonresident viruses.

In addition, the methods used by resident and nonresident viruses to locate "victims" differ substantially. A nonresident virus seizes control when an infected program is loaded, and then uses PATH, COMSPEC, or other information to find and infect files. Control is then returned to the infected program. Once loaded, resident viruses masquerade as interrupt handlers, intercepting interrupts and inserting their own code into the interrupt handling sequence. Thus, when a program is being written to disk, a resident virus may seize the output interrupt (21,4B) and infect the program being written. The RCE-1800 virus also subverts the read interrupt and, in particular, infects both files referred to by the COPY command if either has the extension EXE or COM. At the same time, some resident viruses, at installation time, use COMSPEC to find and infect COMMAND.COM. Thus, residence does not completely determine the method of finding "victims."

File Virus Characteristics. For file viruses, the most obvious easily observed objective property is the increase in length of infected files. This increase is due to the addition of the virus code and can be used to identify the type of virus responsible for the infection. However, there are two possible problems in this regard. First, the extension may vary, depending on the length of the infected file (many viruses, upon injecting themselves into a file, extend it to the nearest address that is a multiple of 16).

Second, the size of the extensions may differ between COM and EXE file. Thus, for the characteristic we use a normed extension:

1. For viruses of types C and CE the characteristic is equal to the extension of a COM file, which is a multiple of 16 (we must eliminate the influence of extensions to paragraph boundaries (addresses that are multiples of 16)). This is appropriate for the viruses that insert their code at the end of COM files.

2. For viruses of type E, the characteristic is the minimum extension of infected EXE files with length taken modulo 16.

Determination of the indicated characteristic requires no experimentation if the length of the infected file is a multiple of 16. We need only compare the extensions of two or more infected COM files. Most often, file viruses infect the command processor (COMMAND.COM) of MS DOS and programs named in AUTOEXEC.BAT. During analysis of several infected files, two basic cases can be distinguished. If the sizes of the extensions are the same, while the lengths of the initial files modulo 16 differ, then the virus does not extend itself to the end of a paragraph and the extension length that has been found is the characteristic. If the extensions are different, then the virus extends itself to a paragraph boundary and the characteristic LX of the virus must be obtained from the formula

$$\mathrm{LX}=\mathrm{DELTA}-(16-\mathrm{mod}(\mathrm{LEN},16)),$$

i.e., we subtract from the extension obtained (DELTA) the complement (relative to 16) of the length of the initial file modulo 16. For example, COMMAND.COM (this file is usually among the first attacked) in MS DOS 3.3, which is the most commonly available version at the present time, usually has a length of 25,307. In this case, 25,307 modulo 16 is 11. The 16's complement of 11 is 5 (i.e., to bring the code to a paragraph boundary requires 5 bytes). Thus, the characteristic will be 5 bytes less than the length added to the infected command processor. An additional advantage of this approach is that with rare exceptions such as RCE-1813, the characteristic obtained this way is the length of the virus.

File Virus Descriptors. The following characteristics of viruses are used:

A — *READ ONLY files attacked* (y — the attribute is removed and then restored, n — READ ONLY files are not infected, r — the attribute is removed and not restored);

B — *MS DOS warnings are suppressed* upon attempt to infect write protected files (y — yes, n — no);

C — *COMMAND.COM is a target* (y — COMMAND.COM can be infected, n — command processor is not attacked, o — only the command processor is a target, s — victims found by using COMSPEC);

D — *creation date and time changed for infected files* (y — yes, n — no, s — changes only seconds in the creation time);

E — *method of determining file type* (e — from the extension, in which case EXE files with extensions of COM are infected, or files are destroyed, s — from the first two bytes, since EXE files usually begin with the prefix "MZ");

F — *minimum length of infected files* ("integer" — infects only files with length greater than the given integer);

I — *increment in the file length upon infection* (c — constant, p — extended to paragraph boundary, n — absent);

J — *affected by first instruction in COM files* (n — does not inspect first instruction in COM files, j — infects only COM files in which the first instruction is JMP, s — does not infect files beginning with JMP, x — unknown);

K — *multiplicity of infection* (⟨digit⟩ — number of files infected in a single attack, m — multiple. For type CE viruses, the first digit is the multiplicity of COM file infection, and the second digit is the multiplicity of EXE file infections);

L — *size of virus in bytes* (⟨integer⟩ — the length of the virus, i — equal to the normed increase in file length upon infection);

M — *concealment of resident viruses* (i — evades detection by filters monitoring state vectors for damage, m — evades detection by memory map scans, t — evades detection by trace programs, n — none);

P — *physical location of infection* (h — beginning of file, t — end of file, m — middle of file. For CE type viruses, the first letter indicates the position in COM files, while the second indicates location in EXE files);

S — *invasion strategy* (e — attacks existing files not in use, r — attacks during file reading, o — attacks during file write, c — attacks files in catalog, p — uses PATH);

U — *upper limits on length of infected files* (n — infects COM files for which length after infection exceeds 64K (thus destroying them), y — does not infect oversize files, e — does not attack EXE files with length greater than 64K (uses a strategy based on "conversion" of EXE files to COM files));

X — *infects "extended" EXE files*, in which the length indicated in the header is greater than the length in the trailer. At load time, only the header value is used, and the remainder is used as buffer space (y — infects, attacking middle of file, thereby destroying or damaging program code, t — infects by insertion at end of code, n — does not attack).

File Virus Signatures. As noted above, for a signature we use a hexadecimal code characterizing the program. The signatures given in Appendix 1 were subjected to the following rule: if the M-signature is induced in the I-signature, it is given after the I-signature. As we noted above, T-signatures do not exist for all file viruses.

For group viruses it is useful to think in terms of group signatures, which are available for all known strains and for which it is assumed there is a good chance of preservation in still unknown strains. These signatures are identified by the suffix G.

CLASSIFICATION OF BOOTSTRAP VIRUSES

Bootstrap viruses differ inherently from file viruses, so the rules for selecting codes, descriptors, and signatures are somewhat different. There are fewer bootstrap viruses than file viruses, and they are less infectious (there are obviously fewer diskettes than files). The bootstrap viruses are classified in Appendix 2. Like the file viruses, most of the bootstrap viruses have different strains, which can be combined into groups. The following groups are known at the present time:

the *Italian group* (the first detected virus in this group was Bx1-1C, which appeared toward the end of 1987);

the *Pakistani group* (which includes Brain, Ashar, and, possibly, Disk Killer; the first detected representative of this group was Dx3-E9, which appeared in 1986 in Lahore).

Bootstrap Virus Code. Since bootstrap viruses must be resident, it is useless to use the prefix R for them. In comparison with resident file viruses, the most important characteristic of bootstrap viruses is whether or not they remain in memory after a warm reboot via Ctrl/Alt/Del. We indicate this property by means of the letter W (for warm reboot). All bootstrap viruses attack floppy disks, but some attack winchesters and others don't. We use the prefix D for viruses that attack only diskettes. Now, when the boot sector is infected, there are two possible cases: if the boot sector of the C partition of a winchester is subject to attack we use the prefix B, while if the Master Boot Record (MBR) is vulnerable, we use the prefix M. Also, if, as part of the infection process of either a winchester or a diskette, the virus reserves additional space by marking clusters unusable, we add the suffix "x," followed by the number of clusters reserved (e.g., Bx1).

Bootstrap Virus Characteristics. The choice of a characteristic for bootstrap viruses presents certain problems. Although the role played by the increase in the length of files infected by file viruses is similar to the reduction in space available to DOS when it is infected by a bootstrap virus, we noted above that the virus property selected for the characteristic must be testable on an infected machine. The amount of memory used by a bootstrap virus does not meet this criterion, so we must reject it. Nonetheless, analysis of the amount of memory used by DOS is a useful diagnostic tool.

Thus, we selected a different "observable" characteristic of bootstrap viruses — the infected boot sector. For the characteristic we selected the value of the second byte in the bootsector, the contents of which differ for the types of bootstrap viruses known to the author.

It must be emphasized that the contents of a suspect boot sector must be examined only on write-protected disks, since the very act of examining the sector may cause the virus to restore the boot sector to its unmodified "pure" form (the virus Brain uses this trick) or, worse, may trigger some other unsanctioned operation. Nonetheless, this approach is flexible and makes it possible, in case of ambiguity, to proceed to use the third byte as well. Note that it is desirable to use a "cold start" (by means of a RESET key if available, or by powering down and up if not), rather than a "warm start" (use of CTRL/ALT/DEL). As we have already noted, a number of bootstrap viruses intercept keyboard interrupts and can preserve themselves in memory in the case of a "warm" start.

Bootstrap Virus Descriptors. For bootstrap viruses we use the following systematization of properties:

L — *the size of the virus code in bytes;*

I — *the amount of memory "stolen" by the virus from DOS upon loading;*

N — *the location of the first byte that differs from the contents of a normal boot sector;*

S — *the location of the "tail" of the virus and the original boot sector* (xN in sector N of a cluster or group of clusters marked unavailable; 1 — on the last track of a diskette or winchester; NNN the absolute address (cylinder/head/sector) on the diskette or winchester (if they coincide, we give one address));

M — *concealment of infected boot sector when the virus is resident* (n — visible, y —not visible (the virus restores the original boot sector));

W — *possibility of overwriting an infected boot sector on an infected machine* (y — yes, n — not possible);

Boot Virus Signatures. In addition to the signatures used for file viruses (B, I, and M), for bootstrap viruses it is useful to take for a signature the first three bytes in an infected boot sector (the J-signature). The contents of these three bytes in a normal boot sector are EB3490, a JMP instruction to branch around the parameter table.

In conclusion, we should note that although MS DOS, which is distinguished by the almost total absence of protection against illegal operations, makes it easy to construct computer viruses, the viruses are not programs that take advantage of errors or defficiencies in the operating system. Their operation requires only rather ordinary use of available facilities used by the majority of "normal" programs. Thus, it is impossible to provide blanket protection against viruses. Nonetheless, the spread of viruses can be substantially impeded by use of special methods in both the operating system or other means of protection. The author intends to address this problem in a separate article.

LITERATURE CITED

1. F. Cohen, "Computer viruses: theory and experiments," Proc. 2nd IFIP Int. Conf. on Computer Security, **1984**, 143-158.
2. A. Solomon, "A Trojan War," Pers. Comput. World, **11**, No. 8, 166-170 (1988).
3. L. S. Penrose, "Self-reproducing machines," Scientific American, **200**, No. 6, 105-114 (1959).
4. "Darwin," Soft.: Pract. & Exper., **2**, No. 1, 93-96 (1972).
5. A. K. Dewdney, "A Core War bestiary of viruses, worms, and the other threats to computer memories," Scientific American, **252**, No. 3, 14-19 (1985).
6. J. Brunner, "The shockware Rider," Harper & Row, N.Y. (1975).
7. J. F. Shoch and J. Hupp, "The 'worm' programs — early experience with a distributed computation," Commun. ACM, **25**, No. 3, 172-180 (1982).
8. A. K. Dewdney, "In the game called Core War hostile programs engage in a battle of bits," Am. Scien., **250**, No. 5, 15-19 (1984).
9. P. J. Denning, "The Internet worm," Am. Scien., **77**, No. 2, 126-128 (1989).
10. N. N. Bezrukov, "Heuristic methods for improving disassembly quality," Programmirovanie, No. 4, 81-93 (1988).
11. N. N. Bezrukov, "Decompilation of control structures by the method of decomposition of program graphs into simple nets," in: Theoretical and Applied Problems in the Use of Computers [in Russian], KNIGA, Kiev (1988), pp. 3-22.
12. N. N. Bezrukov, "Classification of computer viruses and defense against infection," in: Software Quality Assurance of Real Time Mini- and Microcomputer Systems [in Russian], KNIGA, Kiev (1989), pp. 3-21.

APPENDIX 1

Informal name	Virus classification code	form	Descriptor and signature — value	Notes
1	2	3	4	5
Lehigh —	RC-0, RC-20	D M I +	AnBxCoDyInJnK1L346K1PmSe B72102C3 505380FC4B740880FC4E7403E977018 BDA807F013A75058A07EB07	Destroys FAT of diskettes and winchesters. Information on length of infected files is contradictory (0 or 20 bytes).
405 virus	?C-405 or ?CE-405	M I +	19CD2126A24902B447040150 B8000026A2490226A24B0226A28B025 0B419CD2126A24902B4470401	Effects unknown.
	RC 529	D B	AxBnCnDnEeLxIeJnK1LxMnPhSeLn B815CA8B361B01BF00018B0E1D018B1E19	Variant of RCE-1813.
Vienna-M Monster, 13 Months, Micro-88	C-534	D B T T T	AxByCxDxEeF256IeJnK1LiPtSeUx D681C60000FCB90300BF0001F3A48BFAB4 «????????COM». «Microsoftyright». «Microsoft 1988»	Variant of C-648. Upon infection, month in creation date is changed to 13 (indication of infection).

Vienna-Y	C-623	D B T	AnBnCyDsEeF1OIcJnK1LiPtSpUy B42FCD21891C8C4402B82435CD21899C8F00 «*. COM», «PATH-», «????????COM»	Variant of C-648. Distinguished by abnormal termination. In "weakened" modules the reload address is replaced by C000,0000.
Vienna-A , DOS-62 , Time Bomb	C-648a	D M M I + T	AyByCyDsEeF1OIcJnK1LiPtSpUv 8BFE81C71F008BDE81C61F00 strain A 8BFE83C71F908BDE83C61F90 strain B FC8BF281C60A00BF0001B90300F3A4 8BF2B430CD213C007503E9C701 «*. COM», «PATH=», «????????COM»	Attacks individual files, since MS DOS is reloaded when they are written. When COMMAND.COM or files in AUTOEXEC. BAT are attacked, reload loops. Has a number of different strains.
1168, Datacrime	?C-1168	M I +	EB00B40ECD21B4 8B36010183EE038BC63D000075 03E9FE00	Erases file data. Has three variants
1280, Datacrime-2	?CE-xxxx	I +	8A9403018DBC29018D8CEA065E81EE030183 FE00742A2E8A9403018DBC2901	Strain of Datacrime.

Informal name	Virus classification code	form	Descriptor and signature value	Notes
1	2	3	4	5
TP-04, Vacsina-4	RCE-1212	D M T B	AyBnCyDyEsFxIppJjK12LiMyPttSeUye 1726C5B5000183C7 «VACSINA» F47A04	Destroys file creation data.
TP-05, Vacsina-5	RCE-1206	D M T B	AyBnCyDyEsFxIppJjK12LiMyPttSeUye 1726C5B5000183C7 «VACSINA» F47A05	EXE files are attacked in two stages. Initially length increases by 132 bytes, and converted to COM files for the system (EXE extension is unchanged), and then attacked as COM files. Attacks files up to 63K long. Creates secret file on disk.
TP-16, Vacsina-16	RCE-1339	D M T B	AyBnCyDnEsFxIppJjK11LiMyPttSeUye 1726C5B5000183C7 «VACSINA» F47A10	Differs from TP-5 by infecting in one step. EXE files converted to COM.
1536, Zero Bug	???-1536	M	EB2B905A45CD602E	Symptoms unknown.
Cascade Falling letters	RC-1701	D M I + M	AyByCyEsDnF10IcJxK1LiMvPtSeUy 31343124464C77F8 (strain C) FA8BECE800005B81EB31012EF6872A01017 40F8DB74D01BC820631343124464C75F8 strain B) 31343124464C	"Falling" letters displace material, accompanied by "rustling." Interferes with operation by locking out keyboard. May restart upon reboot.

Cascade-B	RC-1704	D I +	AyByCyEsDnF1OIcJxK11LiMyPtSeUy FA8BECE800005B81EB31012E1 6872A0101 740F8DB74D01BC850631343124464C75F8	Variant of RC-1701, replicates on IBM PC clones only.
1704-C 1704-Format	RC- 1704f	I +	F6872A0101740F8DB74D01BC8506313 43124464C77F8	Effects unknown.
TP-24, Musical restart-24	RCE- 1760	D M T B	AyByCyDnEsF256IppJxK11LiMyPtlSeUn 1726C5B5000183C7 «VACSINA» F47A18	20-sec musical melody played upon attempt at warm reboot; ; normal restart after end of melody.

Informal name	Virus classification code	Descriptor and signature		Notes
		form	value	
1	2	3	4	5
Dark Avenger Eddie	RCE- 1800	D M T	AyByCyDnEsF1774IcvJnK11LiMnPtlSer 4F0789072EA151 «Eddie lives... somewhere in time», «This program was... Dark Avenger»	Destroys disk sector after each 16 installations. Occasional destruction of COM files (length extended beyond 64k).
TP-25, Musical reboot-25	RCE- 1805	D M B B	AyByCyDnEsF256IppJxK11LiMyPtlSeUn 35CD218BF38CC7 9CFA2EFF1E1400C3 F47A22	See TP-24
Jerusalem Black Friday Israeli	RCE- 1813	D M M M M M I + T	AyBnCnDnEeF8IcpJnK1mL1808MvPhtmSeUyXm 2EFF0E1F00EB122EC7061F strain A 122EC7061F000100505156 strain B E992000000000000000000 strain B-2 73555249560032 — strain D 73555249560033 — strain E 8ED0BC000750B8C50050CBFC062E8C0631 002E8C0639002E8C063D002E8C0641008CC0 «sUMsDos», «COMMAND. COM»	Reduction in response of infected DOS, with message "stack overflow," display of black square in left corner, destroys programs being written out on Friday the thirteenth. SCAN distinguishes a number of variants.
TP-34	RCE- 2568	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A22 «0505050505050505»h ((8)05h)	See TP-33
TP-33	RCE- 2680	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A21 «0505050505050505»h ((8)05h)	Symptoms disapear upon attempts to trace operation on infected machines. The melody Yankee Doodle is played at 17:00.
2730	???- 2730	M	9177917AA4B7570056000000	Symptoms unknown.
TP-38	RCE- 2756	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A26 «0505050505050505»h ((8)05h)	Does not appear in memory map.
TP-39	RCE- 2772	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A27 «0505050505050505»h ((8)05h)	See TP-38
2086, Fu Manchu	RCE- 2086	M I	72454D484F72 8ED0BC200950B8230250CBFC062E8C062C0 02E8C0634002E8C0638002E8C063C008CC0	

Informal name	Virus classification code	Descriptor and signature		Notes
		form	value	
1	2	3	4	5
TP-44, Samoed-44, Five O'Clock, Yankee Doodle	RCE-2885	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A2C «0505050505050505»h ((8)05h)	See TP-41. Melody Yankee Doodle played at 17:00 with probability 1/8 instead of every day.
TP-45	RCE-2901	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A2C «0505050505050505»h ((8)05h)	See TP-44.
TP-41	RCE-2932	D B T	AyBnCyDnF10EsIppJnK11LiMmtPtSpe F47A2C «0505050505050505»h ((8)05h)	See TP-44.
2930	?CE-2930	M I +	148B4D168BC18ACD E82906E8E005B419CD218884E30 0E8CE048A95E2000E1F7509	Effects unknown.
3066/2930, Traceback	?CE-3066	M I +	148B4D168BC18ACD E87106E82806B419CD2189B4510 18184510184088C8C5301	—»—
3551, Syslock	?CE-3551	I + M	D1E98AE18AC1 33061400310446 46E2F25E5958C3 330614003104 4646E2F2	—»—
April First COM-virus	RC-xxxx	M I + +	73555249560031 89263401B419CD2104412EA265032EA2B 103BF6703578BF2807C013A750D8A042E A265032EA2B103	May issue message "APRIL 1ST HA-HA-HA YOU HAVE A VIRUS." If so, DOS loops.
April First EXE-virus	RE-xxxx	I +	2EA31700BB17000E1FB4DECD21B42ACD2 181FA0104742281F9BCO77506E8C504	The same as preceding, but for EXE files
Cookie Cookie Monster	RCE-xxxx			Displays message I WANT COOKIE. Only input of work COOKIE from keyboard permits operation to continue. Some strains of this virus destroy files in the case of incorrect input.

Informal name	Virus classification code	Descriptor and signature		Notes
		form	values	
1	2	3	4	5
DBASE virus	RCE xxxx	M	80FC6C74EA80FC5B74E5	Swaps first two bytes of each block during output operations. Attacks DBF files, so disk record is damaged. Reading of file on uninfected machine causes errors to become apparent.

Friday the 13th COM-virus	RC-xxxx	I +	1E8BECC746100001E80000582DD700B 104D3E88CCB03C32D100050	Performs some action on Friday the 13th. See RCE-1813.
Ghost Version of DOS-62	C-xxxx	M	5E81C65A04B80102	Form of C-648.
Sunday	RC-xxxx	M	C8F7E1EEE70001	Issues message on Sundays.
Typo COM virus	RC-xxxx	M	99FE26A15A002E89	Intercepts the time interrupt, searches the screen, beginning at a random location, for a series of digits. Then swaps two adjacent numbers.
Saratoga/ Icelandic	?CE-xxxx	I + M	8CDB4B8EDBB04DA20000A103002D8000 A3030003D8438EC333F633FF0E1FB9D007 A3030003D8438EC333F633FF	Symptoms unknown.
MIX1/Ice-	???-xxxx	M	43813F455875F1B80043	—»—
Icelandic-B	???-xxxx	M	2E8E1E6D02B90030BE0000	—»—
Iceland II	?CE-xxxx	I +	26C6067F03FFB452CD212E8C066D022 68B47FE8EC026030603004040	Variant, may be the same as one of the preceding.
AIDS	???-xxxx	M	42E8EFE3BFCA031E	Symptoms unknown.
Pentagon	???-xxxx	M	EB349048414C2020	—»—
Virus-B	???-xxxx	M	B44FCD2173F758	—»—
Ohio	???-xxxx	M	EB2990493412000100000000	—»—
Do Nothing	???-xxxx	M	720450EBO790B44C	—»—
Lisbon Virus	???-xxxx	M	8B44793D0A0072DE	—»—
Ghost Virus	???-xxxx	M	90EA59EC00F09090	—»—

APPENDIX 2

Informal name	Virus classification code	Descriptor and signature		Notes
		form	value	
1	2	3	4	5
Stoned Marijuana	M-05	D J M I + T T	L512 I2 N1 S0. 1. 3—0. 0. 7 Mn Wn EA0500 00535152065657 1E5080FC02721780FC0173120AD2750 E33CO8ED8A03F04A8017503E80700 Your PC is now Stoned LEGALISE MARIJUANA!	Upon booting DOS, the message "Your PC is Now Stoned" appears with probability 1/8. Interferes with drivers (for example, 800.COM), causing damage to information on diskettes in 720k format.

Italian Bouncing Ping-Pong Bouncing Ball	Bx1-1C	D J M M I +	L.1024 I2 N2 Sx1 Mn Wn EB1C90 595B58071FEA (strain, A) A1F581A3F57D8B36F981 (strain, B) 8ED8A113042D0200A31304B106D3 E02DC0078EC0BE007C8BFEB90001	Rhomboid moves about screen, bouncing off screen edges and pseudographic symbols. Intensity of symbols or fields on screen may change. Strain B has 2 moving symbols.
Brain Pakistani virus	Dx3-E9	D J I + M T T T T	L.3072 I7 N1 Sx3 My Wn FAE94A 8CC88ED88ED0BC00F0FBA006 7CA2097C8B0E077C890E0A7CE85700 8D88ED0BC00F0FBA006 Welcome to the Dungeon (c) 1986 Basit & Amjad (pvt) Ltd. BRAIN COMPUTER SERVICES LAHORE—PAKISTAN	Damages the FAT of affected disks. Relabels disks "c. Brain." Restores original boot sector when infected disks are examined. Thus, it is possible to see the contents of the infected boot sector only when load is performed off write-protected disks.
Ashar (Ашар)	Dx3-E9	I + M M T T T T T	8CC88ED88ED0BC00F0FB A0067CA2097C8B0E077C890E0A7CE85900 A0067CA2097C8B0E077C 208CC88ED88ED0 (C) 1988 Basit & Amjad (pvt) Ltd. Lahore, Pakistan. Ver (Singapore) Beware of this «virus». It will transfer to million of Disk	Variant of BRAIN, apparently developed in Singapore in 1988.
Disk Killer OGRE	Bx3-EB	D J M T T	L3072 I8 N1 Sx3 Mn Wn FAEB4F C310E2F2C606F301FF90EB55 Disk Killer — Version 1.00 by COMPUTER OGRE 04/01/1989	Damages FAT and individual sectors.
Alameda	WB-xx	D M I +	L512 I1 S1 B400CD13720DB801 BB40008EDBA11300F7E32DE0078EC00 E1F81FF56347504FF0EF87D	Upon distribution of MS DOS, the cluster with the original boot sector causes errors during bootstrapping.
Falling Letters Boot, Israeli Boot, Swap	?-xx	I + M	31C0CD13B80202B90627BA0001 BB00208EC3BB0001CD139A00010020 CD13B80202B90627BA0001	May exhibit visual symptoms similar to CR-1701.
Den Zuk	B-xx	M I	8EC0BEC67CBF007E FA8CC88ED88ED0BC00F0FBB8787C50C3	Symptoms unknown.
Typo	B-xx	M	241355AA	May change the location of symbols on the screen after a specific time interval.
Boot Killer	Bxx-xx			On a winchester, creates a chain of lost clusters about 2M long. Formats the first track of the disk (8 sectors numbered 2 through 9).