Detecting Viruses in the NetWare Environment

Articles and Tips: article

MORGAN B. ADAIR
Technical Consultant
Systems Engineering Division

01 Mar 1992

A network can act as a vehicle for virus propagation, but it can also serve as the point where viruses are detected and their spread is prevented. This AppNote describes how different types of PC viruses behave in a NetWare network, and outlines how network managers can use NetWare's security features to limit the spread of viruses. The AppNote also introduces a NetWare Loadable Module that detects symptoms of virus infections.

→ Virus Threat and Virus Hype
→ Categorizing Viruses
→ Boot Sector and Partition Table Viruses
→ Parasitic Program-Infecting Viruses
→ NetWare and Viruses
→ PINCH - Program Integrity Checker NLM

Virus Threat and Virus Hype

Few words strike greater terror into the hearts of network managers everywhere than *virus*. Surveys consistently list viruses as one of the main concerns of network managers. The industry press sometimes raises the fear level by exaggerating the degree of threat, and even when the information reported is accurate, it doesn't always provide enough technical information about how specific viruses work and how they spread.

This AppNote examines the impact of DOS viruses on NetWare networks. First, it discusses how viruses can be classified by their location on disk and by their use of memory. Then, it takes a detailed look at two classes of viruses that present a particular threat to networks: viruses that infect both boot sectors and partition tables, and viruses that infect program files. Two specific viruses are examined in detail: how they spread, the level of damage they can do, and the nature of their threat to networks. Next, the AppNote prescribes actions network managers can take to prevent the spread of viruses on their networks and how to limit the damage viruses can do. Finally, the AppNote describes a NetWare Loadable Module (NLM) that can detect the symptoms of a virus infection on a NetWare file server.

Categorizing Viruses

Viruses are programs that are designed to replicate and spread, both within a computer system and between computer systems. Some viruses simply attach to program files, multiply, and do no damage other than waste disk space. Other viruses are slightly more annoying, creating humorous screen displays, or displaying political messages. Others are designed to completely destroy a computer's file system when a triggering event occurs - a certain date, or the disk reaching a certain percentage of capacity.

Several hundred viruses that infect DOS computers are known to exist. Many of these are variants of a few virus families, and others - because they are easily detected or have bugs that prevent them from spreading effectively - are not widespread. Still, with new viruses appearing at an increasing rate, there are many types of viruses that computers and networks must be protected against.

Viruses can be categorized by the location of the infection on disk, and by how the virus uses memory when it is activated. The categories are not exclusive, meaning (for example) that some viruses that infect boot sectors on floppy disks also infect hard ed as *stealth viruses*, that is, they have features that

Categorizing Viruses by Location of Infection

Viruses can hide in several places on a disk:

- Floppy/hard disk boot sector
- Hard disk partition table
- Executable files
- Unused disk sectors
- Companion files

A virus is not magic. It is a program much like any other. Regardless of where the virus is located on a disk, its code must be loaded into memory and executed before the virus can spread or do damage. The following sections tell how viruses in each of the locations mentioned above get executed.

Boot Sector. The first logical sector (sector 0) of a floppy or hard disk is designated the boot sector. Every DOS-formatted disk has a boot sector *whether the disk is bootable or not*. A non-bootable DOS-formatted disk has a short program in its boot sector that a displays a message something like:

```
Non-System disk or disk error
Replace and press any key when ready
```

the first disk it finds. Boot sector viruses copy the boot sector program to another location on the disk, then write a copy of their own code to the boot sector. When the computer is booted from the infected disk, the BIOS executes the virus code, which then executes the copy of the boot sector it has saved elsewhere.

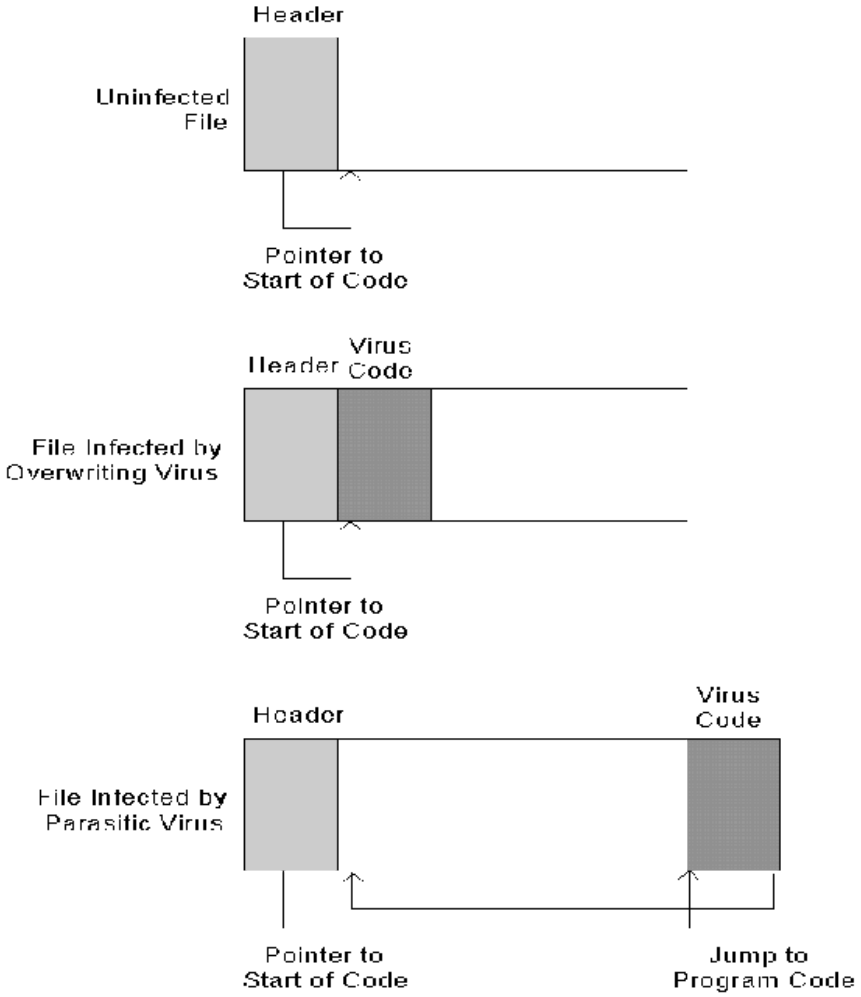
Hard Disk Partition Table. The first physical sector of a hard disk on a PC contains the partition table program. When a computer is booted from a hard disk, the BIOS loads and executes the partition table program. This program first determines which disk partition is the active partition. It then loads the boot sector from the active partition and executes it.

As with viruses that infect boot sectors, partition table infectors copy the partition table program to another location on the disk, then copy their code to the sector normally containing the partition table. The BIOS loads and executes the virus during the boot process, after which the virus executes the copy of the partition table program it has saved elsewhere.

A virus that infects only partition tables could not spread from one computer to another, so partition table infectors must also infect boot sectors or executable files.

Executable Files. Viruses infect executable files by copying the virus code to the program file, either by overwriting part of the existing program code (an overwriting virus), or by appending the virus code to the end of the program file (a parasitic virus). Parasitic program-infecting viruses must modify the program file to assure that the virus code executes first. Figure 1 illustrates how overwriting and parasitic viruses infect program files.

Figure 1: Parasitic and overwriting viruses.



Because overwriting viruses destroy the executable code in infected programs, they are easy to detect and eliminate (by deleting and reinstalling the infected programs). For this reason, only one overwriting virus is commonly seen. The Lehigh virus only infects COMMAND.COM. Since it overwrites COMMAND's stack, copies of COMMAND.COM infected with Lehigh can still execute.

Some program-infecting viruses infect only .COM files, while others infect only .EXE files. Some (like Lehigh) infect only COMMAND.COM. Others infect any .COM file except COMMAND.COM. Some viruses also infect overlay files (.OVL) or the hidden DOS system files (IBMBIO.SYS and IBMDOS.SYS or IO.SYS and MSDOS.SYS).

Companion Files. The virus known as AIDS II takes advantage of a feature of DOS where if a program exists in both .COM and .EXE files in the same directory, the .COM program is executed. A companion file virus does not actually infect program files. Instead, when a user types the name of a program (expecting to execute a program in an .EXE file), the virus in a .COM file of the same name executes, finds another .EXE file, and makes a copy of itself in a companion .COM file for that .EXE file. The virus sets the hidden attribute of the companion file prevent users from seeing the file in a DIR listing.

Unused Disk Sectors. Viruses frequently make use of unused disk sectors to conceal their operation. For example, partition table and boot sector infectors frequently make use of the fact that most of the first cylinder of a hard disk is unused. Viruses can copy the original partition table or boot sector to one of these unused sectors without destroying any data. These viruses are therefore able to survive undetected on the disk longer.

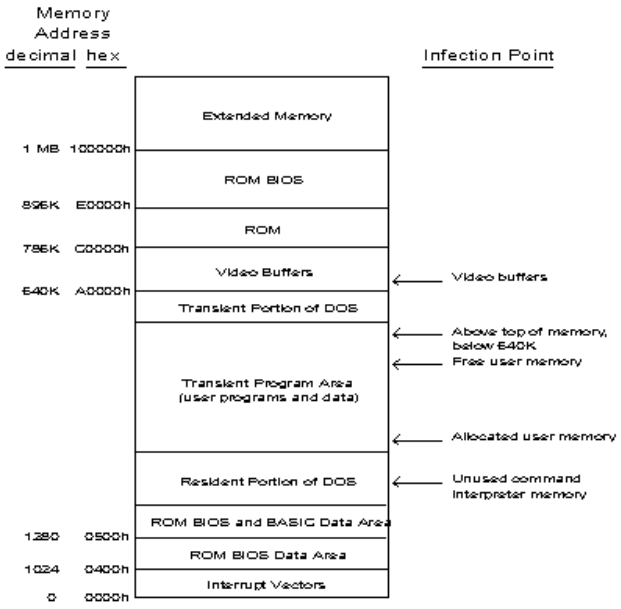
The Dir-2 virus uses unused disk sectors in a unique way that places the virus in a class by itself. Dir-2 writes itself to an unused area on the disk, then changes the directory pointers on the disk so that all program file entries point to the disk sector containing the virus code. The virus saves a copy of the original directory pointers, so it can execute the program the user originally tried to run. The result is that the disk's directory structure is thoroughly scrambled, but this is not apparent until the computer is booted from a non-infected disk and the user attempts to access files on the infected disk.

Once its code is loaded into memory and executed, a virus can take two general approaches: (1) it can look for other disks or files to infect, then terminate; or (2) it can hook an interrupt and remain resident in memory.

Viruses that do not remain resident in memory must locate and infect new disks or files immediately upon execution. This means that the virus has limited time to locate a target for replication, or the user will recognize that something is wrong.

Memory-resident viruses can place themselves into a number of different locations in memory. Figure 2 shows places where memory-resident viruses can hide in memory.

Figure 2: Infection points for memory-resident viruses.



A brief description of each approach is given below.

Resident in Command Interpreter Memory. A few viruses place their memory-resident code into memory allocated to COMMAND.COM, either in its stack space or in the COMMAND data area in low system memory. Because these viruses tamper with COMMAND.COM, they frequently cause system crashes.

Resident in Allocated User Memory. Many viruses simply allocate memory through a DOS call and assume that users will not notice the loss of a few kilobytes of memory. Using this approach assures that the virus code will not be overwritten in memory.

Resident in Free User Memory. A few viruses simply place their code in unallocated memory. Doing so does not decrease available memory, which makes detection by the user less likely. It also means that the virus code is subject to being overwritten when another program allocates that memory. Some viruses deal with this problem by intercepting memory allocation calls to interrupt 21h, and preventing DOS from allocating the memory block occupied by the virus. Other viruses simply do not deal with the problem - if a program overwrites the virus code, the system crashes when one of the interrupts that was hooked by the virus is generated.

Resident Above Top of Memory, but Below 640K. A large number of viruses place themselves resident at the top of memory, just below the 640K boundary. They then redirect BIOS interrupt 12h, which reports the total amount of conventional memory available in the system. This reduces the apparent amount of total memory, and the virus is protected from being overwritten by DOS, because DOS uses the BIOS interrupt to determine the amount of conventional memory available for it to allocate.

Resident in Video Buffers. The Doom II and Doom II-B viruses place their resident code on the video card buffers between 640 and 768K (A0000h - C0000h). This approach does not change the amount of total or free memory, but it results in frequent system crashes when programs attempt to write to the part of the video buffers used by the virus, or when programs change the display mode.

Stealth Viruses

If a virus is to succeed in spreading, it must first escape detection. Many viruses have features intended to hide their presence in memory or on disk. Viruses that have these features are called *stealth viruses*. Some examples of stealth features are:

- When infecting a program, parasitic program-infecting viruses restore the original time stamp on the infected file.
- Memory-resident, program-infecting viruses intercept disk reads of executable files, check whether the file being read is infected,and if so, prevent the program reading the file from reading the part of the file containing the virus code.
- Memory-resident, program-infecting viruses intercept attempts to read file sizes of infected files and return the size of the original, uninfected file.
- Memory-resident, boot sector- or partition table-infecting viruses intercept disk reads of the boot sector or partition table, and return the original contents of the sector being read.

Boot Sector and Partition Table Viruses

Viruses that infect both boot sectors of floppy disks and partition tables of hard disks pose a unique threat to network file servers. A typical boot sector/partition table virus propagates in the following manner:

1. The computer is booted from an infected floppy disk (the virus is in the disk's boot sector).
2. The virus executes the normal boot code from a copy of boot sector it has saved.
3. The virus hooks an interrupt, then terminates and stays resident (TSRs), so it can infect other disks as they are accessed.

Anti-Tel	On the 400th system boot, displays the message "VIRUS ANTITELEFONICA(BARCELONA)" and overwrites the first twohard disks with random data.
Azusa	Every 32 times an infected computer is booted, disablesthe LPT1 and COM1 ports until the computer is rebooted.
Bloody!	On the 128th boot,displays the message "Bloody! Jun. 4, 1989"(date of the Tiananmen Square Massacre).
EDV	After infecting sixother disks while it is active in memory,disables the keyboard and overwrites the firstthree tracks of every disk on the system, thendisplays the message "That rings a bell,no? From Cursy."
Evil Empire	Overwrites rootdirectory entries in sector 10 of floppydisks (if any).
Joshi	On January5th,locks up the system and displays themessage "type HappyBirthday Joshi." If theuser types "Happy Birthday Joshi," the systemagain becomes usable.
Michelangelo	On March 6th,overwrites the hard disk with random data.
Music Bug	Plays music ormakes clicking sound from the system speakerduring boot or disk access.
NoInt	Overwritesroot directory entries on 1.2 MB floppy disks.
Stoned	During boot,might display a message similar to "Your PCis now stoned." Overwrites root directoryentries on 1.2 MB floppy disks.

Detailed Example: NoInt

One of the more common boot sector/partition table viruses is NoInt, or Stoned III. It works much the same as the Stoned family of viruses, except that it does not display a message at boot. NoInt propagates in much the same way as most boot sector/partition table infectors:

1. The computer is powered on with a NoInt-infected disk in the floppy disk drive (NoIntis in the boot sector of the infected disk).
2. The BIOS boot program loads NoInt into memory and executes it.
3. NoInt installs itself at the top of memory and decreases the apparent amount of system memory by subtracting 2,048 bytes from the value at address 40:13h in low BIOS memory.
4. The virus hooks interrupt13h, the BIOS disk I/O services, so that the virus can infect other disks as they are accessed.
5. NoInt executes the normal boot code from a copy of the boot sector it has saved on the infected disk.
6. The virus infects the first hard disk in the system (if any) by copying the disk's partition table to side 0, cylinder 0, sector 7, then copying the virus code to the partition table sector (side 0, cylinder 0, sector 1).

Network Threat

A boot sector/partition table virus cannot spread from a workstation to a NetWare file server, for two reasons:

- NetWare volumes do not have boot sectors or partition tables.
- Programs running on a workstation cannot call the low-level functions that read and write sectors on a file server's hard disks.

However, if the computer being booted from the infected floppy disk is a NetWare file server, the server's partition table might be damaged such that when the file server is brought up, NetWare will be unable to locate its partition on the file server's hard disk.

File Server Prescriptions

A NetWare file server is vulnerable to attack by boot sector/partition table viruses at boot time, and when DOS is running before SERVER.EXE is loaded. Here are some steps you can take to minimize the threat:

- Use a virus scanning program on all floppy disks before inserting them into the server's floppy disk drive.
- Place a sticker over the file server's floppy disk drive with a message on it reminding users to scan all floppy disks before inserting them in the disk drive.
- Always boot from the same disk - whether floppy or hard disk.
- If you do not need a DOS partition on the hard drive - do not have one. (This reduces the temptation to run DOS on the file server machine "just to copy a few files" before bringing up the server.) Boot from a write-protected floppy disk that stays in the floppy drive (copy SERVER.EXE, STARTUP.NCF, and the disk driver to the boot disk).

Parasitic Program-Infecting Viruses

A network can be an effective vehicle for spreading program-infecting viruses. The following is a typical scenario for the infection of a network by a program-infecting virus:

2. The user logs in and runs the infected program.
3. The virus code is loaded into memory with the infected program and executes first. It places itself resident in memory, hooks interrupt 21h, then allows the utility program to run.
4. The user executes MAP and WordPerfect, but the virus is unable to infect the program files, because the user does not have Write rights to the directories where they reside.
5. The user takes a break and plays a game another user has copied into a directory to which all rights have been granted to the group EVERYONE. The virus infects the game program.
6. Another user plays the game on her lunch hour. The virus, now resident in the memory of her machine, infects several utility programs on her local hard disk and the scheduling program used by everyone in the department (she was granted all rights to the directory containing the program, so she could install and configure it).
7. By 2:00 p.m., the virus is resident in the memory of every computer in the department (except the Macintoshes), and has infected COMMAND.COM on every boot disk. At 3:00 p.m., the system administrator starts getting phone calls. Two users' computers lock up every time they try to run the scheduling program. One user had an ominous message appear on her screen. Another user gets an "insufficient memory" message every time he tries to run WordPerfect, although it ran just fine this morning. The system administrator logs in as SUPERVISOR and tries to run the scheduling program. It runs just fine for him (meanwhile, the virus has gone resident in his computer).
8. The system administrator runs WordPerfect. It also runs okay (although it may not next time, now that he has infected it).
9. Remembering the ominous message, the system administrator begins to suspect a virus, so he maps a search drive to a directory containing a virus scanning program. He runs the program, telling it to scan the entire file server directory structure (note that he has infected both the MAP utility and the virus scanner in the process). Fortunately, the virus scanner recognizes that its own program file has been infected, and that there is a virus resident in memory. The scanner tells the system administrator to reboot with a write-protected DOS disk, then run a separate program to remove the virus from all infected files.
10. The system administrator spends the next two hours scanning and removing the virus from infected program files on the file server. One program file is corrupted and must be restored from a backup tape.
11. Two days later, the system administrator repeats the process after a user executes an infected program from his local hard disk. This time, the system administrator spends two hours disinfecting the file server,and three hours disinfecting hard disks in workstations.

The section titled "Prescriptions," prescribes some measures you can take to prevent a nightmare like this from happening to you.

Some examples of common parasitic program-infecting viruses are:

1575	The "green caterpillar"virus, infects a new file with every DIRor COPY command. Might display a green caterpillarthat eats characters on the screen.
Cascade	When thevirus is memory-resident on a machine witha CGA or VGA monitor, and the system clockis set to a date in the "Fall" (September-December) of1980 or 1988, all characters fall to the bottom of the screen.
Dark Avenger	Each time the virus infects 16 files, it writes garbageto a random disk sector on the workstation.
Jerusalem B	The most common PCvirus, with dozens of variants. Behaviorcommon to many variants is display of a blackbox in the upper left part of the screen, systemslowdown, and deletion of all programs executedon a trigger date (Friday the 13th, Tuesdaythe 13th, October 12th, January 25th, or anyFriday after the 15th of the month).
Keypress	After beingresident in memory for 30 minutes, the virusrepeats keystrokes (for example, the virusconverts one "a" key pressed to "aaaaaa").
Sunday	When activated onany Sunday, displays the message "Today isSunday! Why do you work so hard? All workand no play make you a dull boy! Come on!Let's go out and have some fun!"
Taiwan 4	Gradually slows workstation.
Telecom	Writes a variant of the Anti-Tel virus into the partition table of the workstation's hard disk (see the descriptionof the Anti-Tel virus in the list of bootsector/partition table infectors above).
Vienna	Infects only .COM files. The virus corrupts one .COM file in six, rather than infecting it. Files corruptedby the virus cause a warm reboot when executed.

Detailed Example: 4096

The 4096 virus is one of the more complex program-infecting viruses, and it is a prime example of a stealth virus. The virus behaves as described below:

1. The user executes a 4096-infected program.
2. The virus installs itself at the top of memory, then reduces the apparent amount of conventional memory in the system by about 6 KB.
3. The virus hooks interrupt 21h, then turns control over to the infected program.
4. The virus monitors most DOS functions that deal with files. It infects files when DOS functions 3Ch (create file), 3Dh (open file), or 4Bh (exec) are called for EXE or COM files. When 4096 infects a file, it saves the file's original size, time

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.