David J. Stang, Ph.D.

Computer Virus Survival Guide

National Computer Security Association

Second Edition

September 27, 1991

What's Inside...

Availability: *This book may be purchased only through the National Computer Security Association. For more information, or to order, write NCSA, Suite 309, 4401-A Connecticut Avenue NW, Washington DC 20008. Or call 717-258-1816. Our BBS number is 202-364-1304. Our FAX number is the same as voice, 717-258-1816. This book is priced at $9.95. prepaid. All proceeds benefit the NCSA.*

What is a Computer Virus?

This booklet is designed for managers and users who need to know the key facts about how to prevent, detect, and recover from viruses. In these pages you will find information on how viruses operate and where they come from, recommendations for policy and procedure, advice on how to choose anti-virus products, and guidance on what to do if a computer becomes infected.

Single copies are available for $9.95 prepaid; volume quantity discounts are available. To order, send $9.95 to "Computer Virus Survival Guide", Suite 309, 4401-A Connecticut Avenue NW, Washington DC 20008. This booklet is copyright by the NCSA, and may not be reproduced, in whole or in part, without express written permission of the NCSA.

What is a Virus?

A "**virus**" is software which hides in other programs and requires them as a "host". A virus is not a stand-alone program. All viruses reproduce, making additional copies of themselves. A "**worm**" is software which does not hide in other programs, but rather is self-sufficient, stand-alone code. Worms, like viruses, make additional copies of themselves. A "**Trojan horse**" is a stand-alone program like a worm. Unlike either a virus or a worm, it does not make copies of itself. A "**bug**" is like a virus in that it hides in another program and requires a host. It is like a Trojan in that it does not reproduce.

Defining Virus, Worm, Trojan Horse, Bug, and Bomb

	Requires Host Program?	**Replicates?**
Virus	Yes. Viruses require a host, by definition. By inserting themselves in existing executable programs, they are harder to detect, and ensure that they will be run from time to time.	Yes. All viruses make copies of themselves, infecting other boot sectors, master boot sectors, or programs as the opportunity arises.
Worm	No. A host is not required, because worms are typically a mainframe problem, and do not need to hide from most users.	Yes. A worm makes copies of itself as it finds the opportunity.
Trojan Horse	No. While the term "Trojan horse" sometimes refers to the program containing destructive code, the term is more often used to refer to the entire COM or EXE.	No. Most Trojan horses activate when they are run, and tend to destroy the structure of the current drive (FATs, directory), obliterating themselves in the process.
Bug, Logic Bomb, Time Bomb	Yes. Programmers cannot write a bug without out also writing other code; logic bombs and time bombs are inserted by programmers in otherwise "good" code.	No. This code generally has better things to do than making copies of itself. Logic bombs and time bombs wish to remain hidden, with only their effects visible. Bugs do just about everything except make more bugs.

Purists prefer to call a virus a virus, a worm a worm, and a Trojan a Trojan. The differences between four kinds of "problem software" lie in whether the category requires a host program and whether it makes copies of itself. All four kinds of problem software may cause damage, but this is not part of the definition.

Viruses are more often found in microcomputers running DOS than in UNIX systems, mainframes, or minis. Only in DOS do users spend so much time looking at their files, and only in DOS would a worm be easily caught. Worms, on the other hand, seem to be far more common outside DOS systems than in them. Trojans can be found anywhere, unfortunately, as can bugs. Viruses, worms, Trojans, and bugs can all do damage to your data, although this is not always the case.

Classifying Viruses

Viruses may be classified in many different ways. Some of the common classifications are shown below. To any classification scheme there are exceptions. For instance, there are now a few viruses (eg., the Proud virus) which insert themselves in the middle of a program, neither prepending, appending, or overwriting. A few, such as Invader, infect both the boot sector and files.

Classifying Viruses

Classifying Method	**Major Classification**	**Examples**
Location	Boot	Boot record, floppies only (does not infect hard disk): Original Brain, Den Zuk, Ohio, original Ping Pong, original Stoned.
		Boot record of hard disk and of floppy: Disk Killer, Ping Pong-B.
		Master boot record of hard disk, boot record of floppy: Azusa, Bloody!, Joshi.
	Boot and File (Infects boot sector as well as files)	Invader.
	File - Prepending (All code added to top of file)	Amstrad, Cancer, Piter, Pixel family, V-299, Zero Hunt.
	File - Appending (Nearly all code added to bottom of file)	1253, Anti-Pascal, Datacrime, Evil, Jerk, Keypress, Ontario, Phoenix, Raubkopie, RPVS, Striker #1, SVir, Vienna
	File - Overwriting (code is added to top of file, overwriting previous code)	382 Recovery, 405, 4870 Overwriting, AIDS, Deicide, Green Peace, Kamikazi, Leprosy, Lisbon, Mini-45, The Plague, Yukon Overwriting.
Action	Memory resident	Azusa, Brain, Black Monday, Bloody!, Dark Avenger, Den Zuk, Disk Killer, Invader, Jerusalem-B, Joshi, Keypress, Murphy, Ohio, Ping Pong, Stoned, Sunday, Taiwan 3, Yankee Doodle.
	Direct action	405, 4870 Overwriting, AIDS, Deicide, Friday 13th (COM), Leprosy, Vienna.
Escaping Detection	Encrypted Messages	Bloody!, Fu Manchu, Hybrid, Jeff, Jerk, Leprosy, Little Pieces, Phantom,,
	Variably Encrypted	1226, 1260, Cascade, Fish, Holocaust, June 16, Liberty, Ontario, Phoenix,,
	Stealth	512, 4096, EDV, Fish, Holocaust, Joshi, Murphy, Naughty Hacker, Tequila, V651, V800, ZeroHunt,,

Where Do Viruses Come From?

Most virus authors do not sign their work, and it usually requires some detective work to determine where a virus has come from. The table below shows the place of origin of a number of viruses. Information for this table was derived from the 7th edition of *Computer Viruses*, available from the NCSA. Of the 893 PC viruses described in this book, the place of origin of only 318 (36%) is known. Most viruses are first detected at a university or in a city having a university. Most come from industrialized nations. As percent of all computer users, virus authors are more abundant in countries not sharing western concerns for intellectual property. The high numbers of some locations (Minnesota, Italy, Bulgaria, for instance) are a result of one or two prolific authors.

Place of Origin

Country	*Number*	*Region*
South America		***4***
Argentina	2	
Brazil	2	
Asia		***42***
India	4	
Indonesia	14	
Korea	1	
Pakistan	2	
Malaysia	2	
Phillipines	1	
Taiwan	18	
Europe		***185***
Austria	2	
Bulgaria	76	
Denmark	2	
England	4	
Finland	5	
France	3	
Germany	28	
Greece	1	
Holland	1	
Hungary	6	
Iceland	5	
Israel	14	
Italy	7	
Malta	1	
the Netherlands	13	

Place of Origin

Country	***Number***	***Region***
Poland	8	
Portugal	1	
Scotland	1	
Spain	7	
Switzerland	2	
Yugoslavia	1	
Africa		***4***
South Africa	4	
U.S.S.R.		***36***
North America		***39***
Canada	3	
Mexico	1	
U.S. - all		
U.S. - California	13	
UI.S. - Minnesota	4	
U.S. - New York	1	
U.S. - Pennsylvania	1	
U.S. - Puerto Rico	1	
U.S. - Washington	2	
U.S. - Washington, DC	3	
U.S. - other	10	
Australia & New Zealand		***4***
Australia	2	
New Zealand	2	
TOTAL		***289***

The Virus Threat

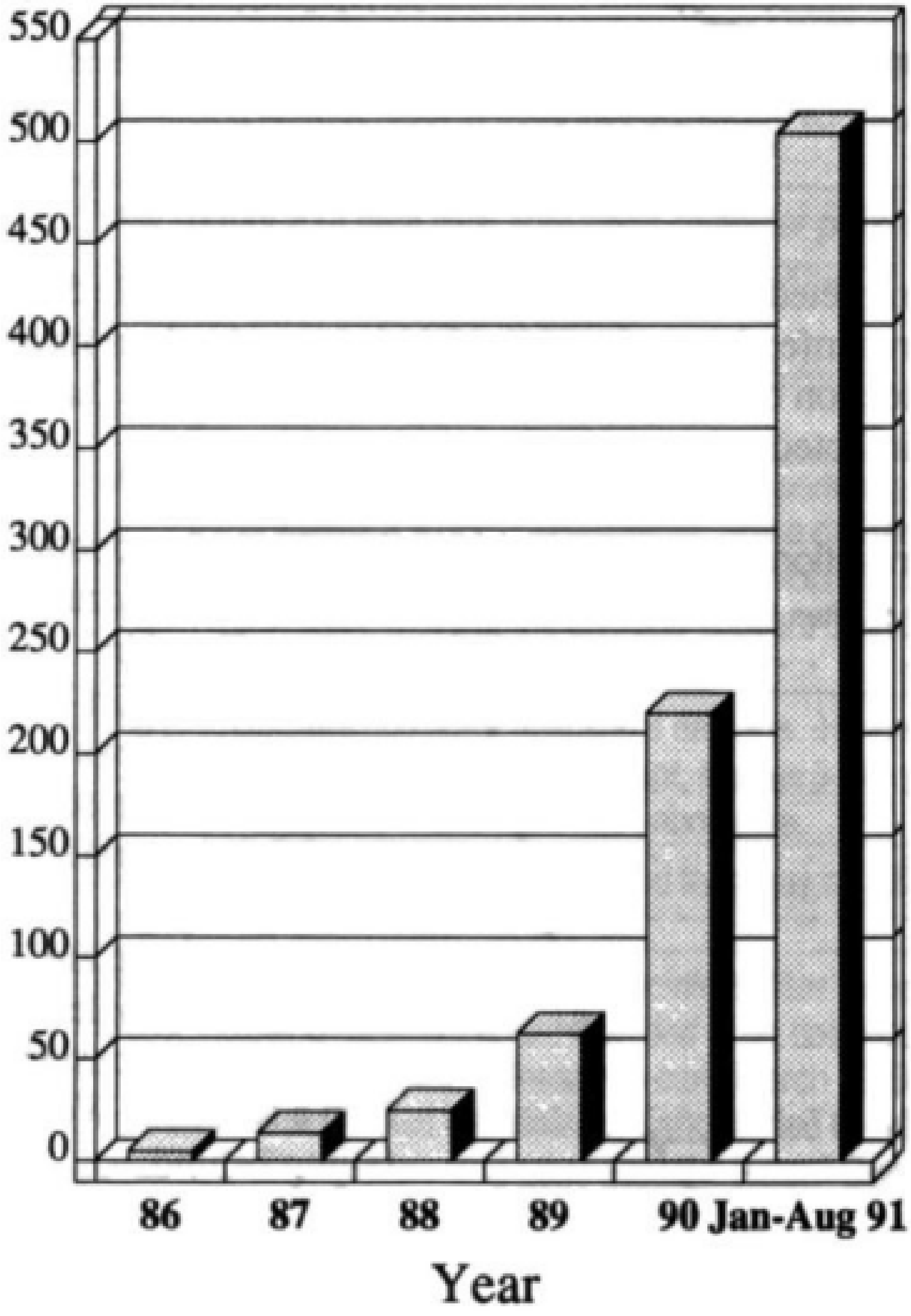
It is our misfortune that we do not have any trustworthy facts about the frequency or costliness of virus incidents. There is no National Center for Computer Virus Epidemiological Studies like the National Institutes of Health. There is no Virus Census Bureau, like the National Centers for Disease Control. There is no world-wide computer health organization, like the many health agencies of the United Nations.

Emergence of New Viruses

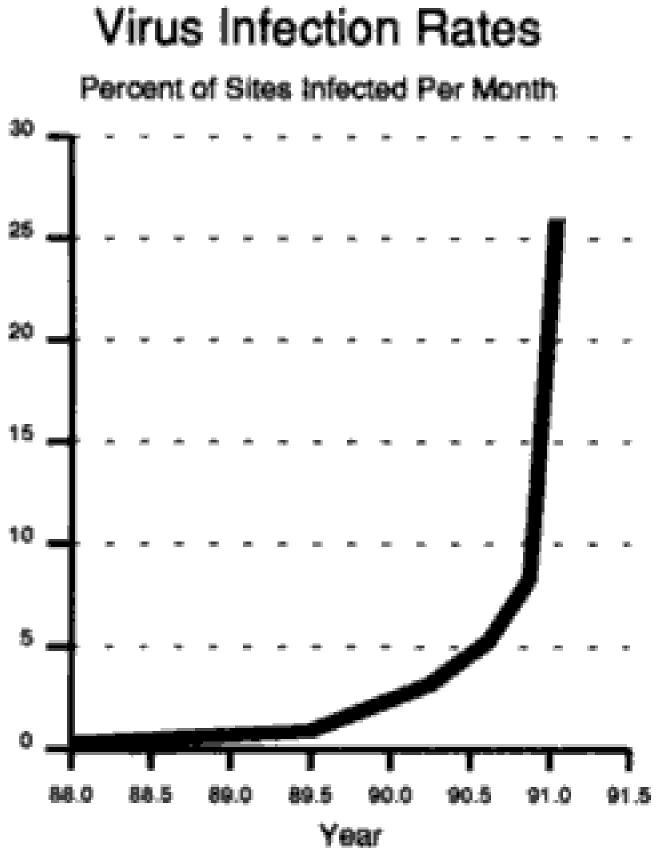
The rate at which new computer viruses are being created continues to increase dramatically. In 1986, there were four viruses, and they were being created at a rate of about one every three months. By 1990, the rate had increased to one every two days. In 1991, we could see about six new viruses emerge on an average day. If new viruses continue to emerge as they have, there will be 38,700 different viruses on the loose by January 1, 1994!

The growth rate for viruses seems to parallel the growth rates for other forms of life, which rise until checked by natural enemies. The computer virus has no natural enemies, other than anti-virus software. The graph below shows the number of viruses written each year. In the first eight months of 1991 alone, over 500 viruses were written.

Viruses by Date of Authorship

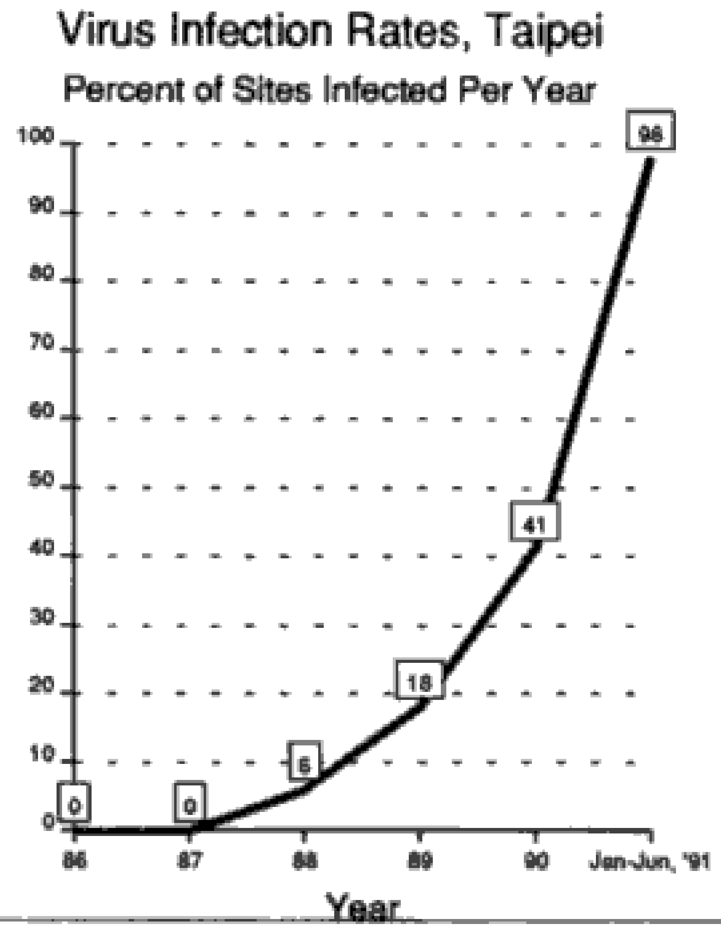


Infection Rates Rising



Computer viruses are becoming an international computer health problem. The rate of infection of corporate microcomputers has been increasing at an alarming rate. A study commissioned by Certus International, sampling from 2,500 large sites having 400 or more micros, found U.S. infection rates rising geometrically.

The study, completed in March, 1991, also found that while 50% have had a virus infection, only 10% of these organizations have any virus protection on most machines and 13% had more than 25 machines infected at once.



A survey of students in the author's virus course offered in Taipei, Taiwan in June, 1991, shows a similar but more substantial growth rate.

In a nutshell, we are looking at a problem that is growing faster than a solution. We are falling behind.

Damage Caused by Common Viruses

Damage Caused by the Most Common Viruses	
1554 (Valert)	Activates on September 1 through December 31 of any year. During this period, it will remove the first ten bytes of any files written during this period, and add 10 bytes of random characters to the end of such files. This will effectively destroy files with headers — graphics, database and spreadsheet files — and cause readily recoverable damage to word processing files. May hang systems with under 640K RAM.
4096	Infects COMMAND.COM, COM, EXE, overlay files as they are opened (run, copied, or xcopied, attribute changed, created.) May hang system. On September 22, may attempt to place a Trojan in boot sectors, intended to display the message "FRODO LIVES".
Aids	Overwrites first 13K of infected programs.
Alameda/ Yale	According to most reports, does little significant damage.
Azusa	May corrupt your data files! May affect computer performance or functioning. Modifies the boot sector. May corrupt or overwrite the File Allocation Table(s).
Black Monday	Every 240 infections or so, the virus will try to format parts of drive C:.
Bloody!	Modifies the boot sector. May affect computer performance or functioning. Makes 3.5" disks unreadable. Some garbage text may appear on the screen. Formerly bootable floppies may fail to boot the system. Much of the damage appears due to bugs in the virus.
Brain	The Brain causes no known intentional damage. However, it can slow diskette access a bit. Resident, it takes 3-7K of RAM. Does not re-infect or make diskettes unbootable.
Cascade	Often produces no obvious effects, and thus may be difficult to detect.
Dark Avenger 1.31	When memory resident, Dark Avenger infects files through any reads, including viewing the file. Intended to be destructive, the author wrote this about its release: "In early March 1989 version 1.31 was called into existence and started to live its own life to all engineers' and other suckers' terror." Overwrites a randomly selected sector with the boot sector. Damage occurs after 16 infections, with the counter stored in the boot record.

Damage Caused by the Most Common Viruses	
dBASE	When an infected application is executed, the virus installs in memory, looking for an open operation on .DBF files. Any writes to this file have two bytes transposed at random. The virus keeps track of which files and bytes have been altered using a file called BUGS.DAT in the same directory as the .DBF files. Reads of data are corrected by the resident portion of the virus, thus data appear correct. When BUGS.DAT is 90 days old, the virus attempts to overwrite/null the root directory and FAT structures of drives D: through Z:. Because of a bug in this virus, the success of this project is somewhat unpredictable. If the virus activates, and your DBF files can be recovered, they will be recovered with non-obvious errors. If the virus is detected and removed before activation, the data you've written to your DBF files in the last three months will be useless because the virus will not be present to "de-garble" it when it is read back. The virus also punishes infrequent dBase users. When you create a new .DBF file, the virus checks the age of "bugs.dat". If it has been more than two months since you last created a DBF file, the virus goes into an endless loop, and computer will hang.
Den Zuk	Modifies boot sector of 360K floppies, removing the Brain virus if found. Unintentionally overwrites the FAT on 1.2M and all 3.5" diskettes. The original causes no intentional damage, but because it places information in track 40, head 0, sectors 1-9, may damage data on diskettes with a valid track 40, such as a 1.2 Mb floppy. Some variants may reformat a floppy disk after a counter reaches a value of 5 to 10 (depending on the version.)
Disk Killer	On floppies, may overwrite one or more portions of files as it sets aside three clusters (5 sectors) for its use, marking them as bad in the FAT. When activated, it will encrypt the data on the hard disk, effectively denying you access to it.
Fellowship	May overwrite the last 20 bytes or more of the original file.
Frere Jacques-A	May crash system during infection, resulting in data loss.

Damage Caused by the Most Common Viruses

Jerusalem	Slows the system 1/2 hour after activation. Corrupts COM and EXE files. If the file length in the EXE header is smaller than the actual length of the file, the virus will overwrite a portion of the file, rather than appending to it. On every Friday the 13th, every program that is loaded is deleted, once an infected program is run. Users will see "Bad command or file name" repeatedly, until they call for help. COM files larger than 63,,723 bytes will be destroyed by overwriting. Three Netware 4.0 functions will be disabled.
Jerusalem-B	This virus is identical to the Jerusalem except that it may not slow the system after infection.
Joshi	Appears to be a non-destructive annoyance. However, Joshi may not permit you to format a diskette when it is active, giving a bad track 0 message (a bug in the virus.)
Korea	On a machine that has only 360K floppy drives, and no hard disk, this virus does little but spread. On all other machines, this virus damages something. On hard disks, the FAT is damaged because the virus stores the original boot sector at track 0, head 1, sector 3.
Liberty	Buggy damage code attempts to convert floppy-writes to formats on rare occasions, but will more likely just hang the machine.
MusicBug	Any file located in cylinder 2, head 1, sector 1 of 360K floppies will be overwritten upon infection. 4K or larger portions of the hard disk may be lost.
Ohio	The virus will freeze the system if a Ctrl-Alt-Del is pressed and a cold boot is then required. When the virus activates, the first copy of the FAT becomes corrupted.
Ping Pong	May damage data on a non-standard disk. Does no "intentional" damage.
Ping Pong B	When a user attempts to format the hard disk, format scans the disk OK and then reports that track 0 is bad. Formatted system floppy disks tend to lock up the PC on boot, and warm reboot doesn't work.
PrintScreen	Slowed drive access.
Slow	The system may hang during the infection of some EXE files.

Damage Caused by the Most Common Viruses	
Stoned	360K floppy disks with more than 96 files in the root and 1.2 M floppies with more than 32 entries may be damaged by overwriting. On hard disks formatted with DOS 2.0 or of a 10Mb capacity, the virus may accidentally overwrite 512 bytes of the FAT, trashing the hard disk.
Sunday (all variants)	Sunday A (with the displayed message) does no apparent damage other than attaching to files. Sunday B (no message displayed) has been reported to delete every file run after the first infected file is run. It has also been reported to damage the FAT on occasion.
Taiwan A (708)	Corrupts COM files. On the 8th day of any month, when an infected program is run, the virus will perform an absolute disk write for 160 sectors starting at logical sector 0 on the C: and D: drives, overwriting the partition table, boot sector, FATs, root directories.
Vienna	One out of eight files infected is destroyed by overwriting the first few bytes with instructions that cause a reboot when the program is run.

Self Defense

In this booklet, we try to present our ideas for the best ways to prevent viruses, the best ways to detect them, and some ways to recover from them.

If you follow the advice in this booklet, your chance of infection — and potentially serious loss — will be minimized. But there are no guarantees. Your organization might do almost everything right and still become infected.

If you do have an infection, feel free to call or write. We'll be glad to try to help.

❖

Preventing Viruses

An ounce of prevention is worth a pound of cure.

Preventing viruses can follow some of the suggested practices for safe sex:

- ❑ Minimize contact with high-risk or unknown software sources;
- ❑ Make sure diskettes placed in your drives are wearing write-protect tabs, and make sure you put write-protect tabs on your diskettes before placing them in the drives of others;
- ❑ Look for signs of "illness" in any software you acquire, and look for signs of infection after using that software.

Unfortunately, preventing viruses isn't sexy. It is boring and often seems to be counter-productive. Much of virus prevention seems to be at cross-purposes to our organization's mission. For instance, puttying over the disk drive and serial ports would stop viruses from getting in or out of the machine. But it wouldn't meet the organization's information sharing needs.

Preventing viruses seems to be disproportionately difficult, when you consider how simple it is to get one. One little computer virus doesn't make two. Given its druthers and a few minutes with an unsuspecting user, it may make a dozen perfect copies of itself. One machine might infect no other in your office for a week or two, or it might infect every computer in the office on the LAN in a minute or two.

Getting them all at once is possible with a scenario like this: Bill gets into work early, and runs a golf game for a few minutes that he has borrowed from a friend. The game contains a virus that becomes active in his computer's memory. Other folks start to arrive, and Bill logs onto the network. The menu program, which he runs, is located in a directory where Bill (and therefore his virus) has write rights. The menu is immediately infected. Other staff log in, run the menu, and are infected. By 9:15, the virus is in every single computer's memory in the entire organization, and now at work finding homes on each user's hard disks and floppies and backup disks.

You won't always be able to follow the advice in this chapter. But do your best, and follow as much as possible, and you will reduce your virus risk.

Guidelines for System Design

In this section we provide some general ideas for how your computers might be interconnected to help prevent the introduction and spread of viruses.

Isolate Systems

When an animal population is widely dispersed, and individuals infrequently come in contact with each other, diseases have difficulty spreading through the species. But when the population is dense, and there is frequent contact, disease can sweep through and devastate. With computer viruses, isolation of systems is one means of reducing your chance of infection.

The idea of isolating systems is anomalous in an age when everyone seems eager to network. But a virus can spread through a network in seconds, infecting every machine on the network. The more nodes you are connected to through a network, the greater the number of sources of infection. If your LAN is connected to another LAN, or to a wide area network, your vulnerability has increased.

Disconnecting from a network may not make sense in your organization. But you should give some thought to isolating critical systems, giving them a network of their own. Perhaps sales and marketing can be on a separate network from accounting, for instance.

One benefit of small, personal networks is that the members of your team care more about the common mission of the team than outsiders do, and may be more careful to check software for viruses before introducing it to the network.

The network is not our only means of connection, however. A virus in the public school's computer can come home with a teenager inadvertently, and find itself a home on the family computer. Mom or dad may take work home with them and use that computer, then bring the virus in to work. Floppy disk exchanges between machines is probably the most common avenue of infection.

This problem can be reduced by giving virus training and tools to all staff with home computers. In a free course for home computer-owning staff, you could cover viruses (along with other topics, perhaps, such as hardware repair, DOS, etc.). You could provide them with copies of the same anti-virus software as they will be using at work, and train them in its use. A virus caught at home is a virus not brought to work. You might even consider extending this to the schools, and send your trainer in for a free class or free scan of their systems.

A hint as to how friends give viruses to friends is provided by tabulating the answers to two questions: "*Have you ever had virus in a computer you were using?*" and "*Do you personally know anyone who has had a virus in a computer they were using?*" When we asked this question in the spring of 1990, 19% of our BBS callers had had a virus, and 39% knew someone who had. In the spring of 1991, the two percentages had increased. But most significant is the interaction here. Of those who had never had a virus, 41% had a friend who had had a virus. Of those who had a virus, 88% had a friend who had a virus.

		Caller has had a virus		
Friend of Caller has had a virus		Yes	No	Total
	Yes	22%	31%	53%
	No	3%	44%	47%
	Total	25%	75%	100%

We can imagine these explanations:

- □ Anyone with a friend who has had a virus is at risk.
- □ Anyone who gets a virus puts their friends at risk.
- □ People who get viruses are likely to have carefree computing policies, and are attracted to people with similar policies.

Because the sharing of diskettes will inevitably happen, here is some advice to help prevent infection:

- □ Don't boot computers from floppies that have been in any other computer. To ensure this is avoided, open the drive A: door immediately after using a floppy, and always check to see that the drive A: door is open before turning the machine on.
- □ If the system is to boot from a floppy, make sure that the boot floppy is labeled to identify the machine it belongs to, and enforce a policy that restricts the use of that floppy to that machine.

Consider creating a virus scanning machine for employee use. The ideal machine contains a menu-driven product that can scan a floppy in A: or B: with the press of a key. All employees could be implored to scan all disks they bring in from home before taking them to their office, and to scan any disks given them by coworkers or received in the mail before using them.

Limit the Function of Systems

Some office computers are specialists, designated for word processing or graphics or spreadsheet work. Even in the organization as a whole, there may only be 20 or so programs that are "authorized", including certain utilities. If you do not permit users to add programs to their system, you will reduce the chance of infection by a file virus.

There are a number of approaches available to limiting the introduction of new software, including software products like F-Prot and Certus, and secure micros such as Xtek's Secura. Before purchasing such products and establishing a policy against introducing new software, you will need to assess how the idea of restricting software will be received. Users who have been unrestrained in the past will likely resent anyone who takes their freedom away. And they might be able to justifiably argue that software they might introduce would make them more productive or capable.

Be Wary of the Help Desk, Support Team as Virus Propagators

Any software lending library, shared computer, or training room is a potential virus transmission site. At even greater risk is the support team, who may visit a user experiencing difficulties, get a virus on their diagnostic disk, then move on to another user, infecting that user's machine in the process. We have heard many stories which implicate hard disk technicians, service personnel, and support staff in the unwitting propagation of viruses.

Production Systems

Any system used to create diskettes for distribution is one which must be given special attention, or you may find your organization the distributor of a virus, as numerous software companies and government agencies have in the past year. Make sure you have the best safeguards on systems designed for software duplication, and be sure to scan very carefully for viruses before shipping.

Guidelines for All Users

Here are some guidelines for your computer users.

Personal Software

Many companies have a policy prohibiting users from executing personal software. The policy probably reduces the amount of software that users bring in from home, but it does not stop it.

Assuming that users will bring in their own software from home, regardless of policy to the contrary, do what you can to keep that personal software personal. Giving each user a different color for their diskettes or a differently colored label for their diskettes may help them sift through what is not theirs, and therefore must be more carefully checked.

Users should assume that any disk they receive from anyone is infected. Such an assumption makes them personally responsible for detection, rather than somehow shifting the responsibility to others. Before using any new disk, they should run their scanner to see if it contains a virus. Assuming they are using a good, up-to-date scanner properly, a failure to find a virus on the diskette is probably sufficient.

Data Disks and Boot Sector Viruses

If you who thought that "DOS Boot" was the name of a German war movie, you might be surprised to learn that data diskettes are the most common means by which boot sector viruses are transferred. About half of office infections are infections of the **Boot Record** (on floppies) or **Master Boot Record** (on hard disks). You may have thought that data diskettes didn't even contain a boot record if they were not bootable. It turns out that the DOS FORMAT program creates a boot record whenever it is executed. A boot record that is created with the command:

```
    FORMAT A: /S
```

creates a boot record, two FATs, a root directory, two hidden system files, and COMMAND.COM on A:. But when a disk is formatted with:

```
    FORMAT A:
```

you create a boot record, two FATs, and a root directory.

A boot record is all a boot sector virus needs to call a diskette home. And *every* disk has one.

How does a boot sector virus get from a non-bootable floppy to a hard disk? It's simple. You put an infected disk in drive A: and copy a file or two from it or to it. Perhaps you edit a file on A:. Five o'clock rolls around and you shut your machine off. Tomorrow you come in to work, turn your machine on, and get a cup of coffee. You come in to see the message "Non system disk or disk error. Replace and strike any key when ready."

You open the drive door and press Ctrl-Alt-Del. But it is already too late. The virus has already been run as your machine tried to boot from A:. For many common boot sector viruses, it has infected your hard disk. When you press Ctrl-Alt-Del, you boot from your hard disk, first putting the virus back in memory, and then putting DOS in memory. From now on, every time you access a floppy in A: or B:, you'll infect it.

Some older boot sector viruses are not able to infect the hard disk, so must take another approach. When you boot (or attempt to boot) from an infected floppy, they go into memory and look for another floppy in a drive. If they find one, they may infect it immediately. If they can't find one, but your boot was successful, they wait in memory until you insert a floppy into a disk drive, and then infect it the first time you access it (perhaps with the DIR command.)

The moral of the story is this:

- ❑ We must now all learn to remove all diskettes from our disk drives immediately after using them, so as to not boot from them.
- ❑ We must never turn the machine on or press Ctrl-Alt-Del without first ensuring that the door to drive A: is open.
- ❑ If we normally boot from a floppy, it should be our own, personal floppy, and it should have a write-protect tab.
- ❑ If we have two drives, we should always try to use B: for reading and writing floppies, and insofar as possible not use A:. Microcomputers never try to boot from B:.
- ❑ With machines that normally boot from the hard disk, and have only one floppy, that floppy should be made B:, not A:, to prevent accidentally booting from it. This can be done by changing the cable connection on the back of the drive, or by moving a jumper on the drive.

Unless the user needs to write to a floppy, the disk should have a write-protect tab on it. This will prevent the infection of programs on the disk and prevent infection of the boot sector. While a write-protect tab on a floppy won't protect a hard disk from a virus on the floppy, it will protect the floppy from a virus on the hard disk.

Backup

You don't need to be told that backups are important. You must believe it by now, because you have heard it so many times. But you may not appreciate your backups today as much as you will the day you need them, because a virus has destroyed everything in your computer.

Backups cannot prevent viruses, but they can prevent the unmeasurably immense damage that a virus can do. With a good set of backups, a virus is nearly no problem at all. Without a good set of backups, a virus can mean the end of Accounts Receivable, the end of many organizations.

- □ Backup your hard disks often. The more important the information stored on the hard disk, the more important it is to back it up often.
- □ Backup selectively, if this will increase backup speed. Consider backing up everything weekly, and backing up essential, live files daily.
- □ Keep multiple sets of backups. If you happen to backup files containing a virus, your previous set of backups may be virus-free.
- □ If you are backing up to tape, backup by file, rather than image, so that if a virus is later detected and removed, you can omit from your restoration any infected files on the backup tape.
- □ Be sure to check for viruses after restoring from your backups. As many as 9 out of 10 who eradicate a virus find it back in a month. Infected backup disks can be the culprit.

Access to Bulletin Boards

Many people think that the bulletin board (BBS) is a common source of viral infection. Some organizations have gone to the trouble of taking modems away from users and not permitting calls to bulletin boards. While some viruses have been deliberately or accidentally uploaded to bulletin boards, then downloaded by subsequent callers, the BBS has been unfairly accused of being a threat. For instance, we know that about half of all infections are boot sector infections, and that boot sectors are never downloaded from bulletin boards.

Trojans, on the other hand, are more often distributed via bulletin boards than via any other mechanism, so the BBS is never fully trustworthy. Here are some general precautions concerning bulletin board use:

- □ Do not download files from a BBS directory that the SYSOP has labeled "new, untested". Let someone else test your files first!

- □ Download shareware from the shareware author's BBS. Most professional shareware authors provide support BBSs for their products. You are guaranteed an uncorrupted version of software when you download it directly from the source. You are also assured of getting the latest program version.
- □ Minimize contact with possible carriers of a virus. Bulletin boards full of games, unprotection programs, and discussions of hacking are probably more risky sources of shareware than other boards.
- □ Always scan software downloaded from a BBS before running it. Make sure your scanner is up-to-date.
- □ Because scanners normally do not detect Trojans, test downloads on a test computer, or do so just after backing up the system you are about to test it on.

Train Users

You should not consider training all users to the Ph.D.-level in viruses. But some training would be a good idea. You should cover such topics as those covered in this document. Users should know the risk that viruses pose to the organization. They should understand common ways that viruses are introduced to an organization, how to prevent them, and how to detect them. Users should know the warning signs of the most common viruses. Each user should have good anti-virus software installed, and know how to use it. If users can't be counted on to use it properly, then a memory-resident program or scanner called by AUTOEXEC.BAT should be installed that will check for viruses automatically. All users should know what to do if they detect a virus (who to call for help, when to call, etc.)

Guidelines for Management

Software Selection & Testing

your business is software testing, evaluation, and selection, en you have much to do to help ensure a virus-free workplace:

❑ Test new software on a test computer. Then, if a problem arises, working computers are not damaged. If no problems arise, the program can probably be copied.

❑ Choose software with known, short distribution histories. If you buy software still in the shrink-wrap, you are safer than if you use a copy of a copy of a copy... (And copying much software is illegal and/or a violation of license agreements.) Remember that any local dealer who accepts software returns may own a shrink-wrap machine, and inadvertently sell you a shrink-wrapped virus. So be careful even with shrink-wrapped software.

❑ Set up a test machine for users that will permit them to safely scan software they bring in from home. This machine, located in a public, accessible location, should have both 5.25" and 3.5" drives and should be running the latest version of your favorite scanner. With a keypress, the user should be able to establish that there is no virus on their disk.

❑ Because most Trojans do their damage the minute they are run, and are not normally searched for by anti-virus software, when testing shareware obtained from bulletin boards, you might want to temporarily install a hard disk write-protection utility.

❑ Apply common sense before testing software. Small programs that claim to do wonderful things (such as a 10K program that some BBS describes as a "great word processor") might be suspect. Programs without manuals should be suspect. The latest version of popular programs should be suspect if not downloaded from the author's BBS. All downloaded games should be suspect.

❑ Favor Trusted Software. Shareware software purchased from the author is probably safer than the same software obtained from a bulletin board or friend. Packaged software still in the shrink-wrap is probably safer than the same software that is borrowed from a friend. Software you have seen running for some time on a friend's machine may be safer than software you acquire from a bulletin board and try for the first time.

Machine Assignment

When possible, assign machines to users, and work to ensure that each user takes personal responsibility for their computer. Make each user a bit of a virus detective, and reward, rather than punish, a user who detects a virus.

Virus Response Teams

Even if your organization is not yet ready for a full-time virus expert, you cannot afford to delay in assembling an informal support group that will begin now to prepare for disaster. You should create an informal virus response team at each site, and if you have two or more sites, a team at central headquarters or regional headquarters that will back up the local teams.

Your team should be knowledgeable about hardware malfunction, especially how to troubleshoot hard disk problems and how to recover data. After all, most virus alarms turn out to be simple hardware or software problems having nothing to do with viruses. You already have a few on staff who have some knowledge of programs such as the Norton Utilities, and can unerase files. Folks like these might volunteer for a virus busting mission. They can learn more about how viruses work from books on the subject, so that when your first virus comes along, they can look it up and decide what to do. And they should be skilled in the use of your organization's anti-virus software, so they can use it to best advantage. The version of the software that they have should be the very latest.

One mission of your virus response team is to ensure that safe, anti-virus preventive measures are being followed. They can do this by helping write policy, by training users, by working with users one-on-one, by installing anti-virus software on personal computers, and by answering questions as they arise. They might also work to ensure that systems are backed up more frequently. The existence of this group — and their office phone numbers — should be known by all users, who should call them at the first sign of trouble.

Another mission of your virus response team is to get ready for the day that a virus slips through your first line of defense. Their response should be immediate, thorough, and well-guided, remembering that more damage is normally done by "experts" after an infection than by the virus itself. To prepare, they should become as well-connected to the world of virus busting as they can, perhaps through reading the books and journals suggested at the end of this document. And they would be well-advised to have a removal and recovery plan for the dozen most common viruses. They might want to make copies of the Master Boot Records and Boot Records of your variously partitioned hard disks, and store them in a safe location. They might want to be sure that systems are backed up, that files are not fragmented, and that a minimum number of files are in the root (CONFIG.SYS, AUTOEXEC.BAT, COMMAND.COM) in case recovery from a FAT-scrambling virus becomes necessary.

Network Virus Prevention Guidelines

Networks are common victims of viruses. They suffer from these problems:

- ❑ They cannot prevent a user from executing programs from a local hard disk or floppy. An infected program run there can access the network through the user's network access permissions.
- ❑ While the LAN can block a user's access to programs residing on the network drive, it does not naturally prevent an infected program on the server from infecting users.
- ❑ The LAN can limit user write access to directories containing programs. But such limitations may interfere with user functioning (especially programmers!) and makes the network unmanageable. Anyone with write permissions in a directory containing programs is a potential source of viruses.

LANs can provide a terrific environment for virus propagation. Any program on your LAN's server, such as a menu system, is a potential virus victim and potential virus distribution hub. If a user runs an infected program at their workstation, a virus may become memory-resident in their machine. If they then access the network, the resident virus will be able to infect any program they run which resides in a directory in which they have write permission.

- ❑ Network managers must ensure that no programs on the network reside in any directory to which any user has write permission. Once this is established (and constantly monitored), then it is only the network manager who puts the network at risk. The network manager, having write permission everywhere, makes all directories vulnerable when he or she logs in from an infected workstation.

- ❑ Do not connect to any network whose security standards are not as good as your own. Work with other network administrators to ensure that no network in a system weakens the overall system.

- ❑ If possible, consider eliminating any E-Mail capabilities for transferring programs. If you must electronically transfer programs, then use a virus detecting transfer approach (such as that offered by Arcen Data), a memory-resident program that will monitor for the inadvertent copying of a virus-infected file (such as that offered by Norton Anti-Virus), or an anti-virus copy program, such as McAfee's VCopy. You can also have all electronic transfers directed to a directory on the network that is constantly scanned for viruses, and that automatically purges any virus found.

Here are some additional steps some organizations have taken to secure LANs against viruses:

- ❑ All public microcomputers that are connected to LANs are either diskless or have a hard disk but no floppy drives.

- ❑ All boot disks are notchless and contain nothing other than the operating system boot files and the software needed for the LAN.

- ❑ All hard disks on the file servers are read-only, with the exception of a scratch area where users can place their temporary files. The scratch areas get erased periodically by staff. Users logging into the LAN are not automatically placed in the scratch directory.

What Won't Work

By this late date, you have doubtless heard plenty of reasonable-sounding advice on viruses from other authorities, and read your share of marketing literature. There is a certain amount of misinformation out there. Here we will make a few comments about some claims that concern us:

Immunization

"Immunization" refers to running a special program that inserts code that will convince Mr. Virus that it has already infected this program, and thus must leave it alone. This is a silly idea. While most immunizers will do no damage to your software, and may do a bit of good, there are a number of reasons why you may not want to waste your time with the approach:

- ❑ Many viruses don't look for a signature in a file, and simply infect whatever they see (example: Devil's Dance);
- ❑ Many viruses that look for a signature are unsuccessful, and re-infect files over and over, as Jerusalem-B virus does to your EXE files.
- ❑ Many viruses use the same place to write a signature. If one signature is placed in that location, you will not have inoculated against any other virus that writes a different signature to that location.

This is not to say that immunization is completely without merit. It will slow the spread of certain viruses. For instance, Dr. Solomon's Toolkit contains a program to "inoculate" memory for three viruses; Fridrik Skulason's F-Prot includes an immunizer that modifies floppies to convince a pair of common boot viruses that they have already infected the floppy. This technique will never detect the virus, but will protect your diskettes from becoming infected with either of these viruses.

Read-Only Attributes for Files

We've seen some folks advising to mark all files as read-only, as if a virus didn't know how to change an attribute, infect the file, and change it back. Most viruses are untroubled by a read-only attribute, and you will only slow down your users and favorite software by doing this.

Other Techniques

There are many other techniques that may occur to power users in their efforts to defeat viruses. It should be remembered that a virus is in potentially complete control of all PC system resources at the moment of infection. Anti-virus programs renamed and stored in hidden subdirectories can theoretically be found by a virus. Write-protected hard disks can theoretically be write-enabled by the virus. Hidden and read-only files can have their attributes modified, be infected, then have their attributes reset. Everything in the system is subject to the scrutiny of viruses as they examine their hosts. Anti-virus systems stored on non-removable media, relying on support files stored on non-removable media or residing in memory are themselves subject to infection.

Because we cannot guarantee that any policies and procedures will be 100% effective in preventing viruses, we must look at how we can detect them. That is the subject of our next chapter.

❖

Detecting Viruses

Most of the procedures outlined in the previous chapter for preventing virus infections are good but imperfect. When several are combined, our risk of infection is reduced. But completely eliminating the risk of infection is probably not cost-effective. The "perfect fix" to the virus problem might be putty. Just buy a can of putty and pack it in your disk drive doors, packing a bit onto the connectors on the back of the machine, you will have prevented the introduction of a virus. Now, if there was no virus in your machine when you applied the putty, you are safe. But software sharing is a necessity of life. We must share software to share information. Putty won't do.

This chapter assumes you won't buy putty, and will continue to share software. It assumes that you will follow the advice in the chapter on preventing viruses, to the best of your ability.

Timing

The most important aspect of detection is not *what* we detect, or *how* we detect, but *when* we detect. If you turn your machine on in the morning, and check for viruses, it is not likely that you will find any that have been installed by the night cleaning crew.

What you'll find, if anything, is a virus that has been roaming through your system since sometime after the last time you checked for viruses. Viruses do not pop up in your machine through spontaneous generation. They get there when you run new programs or boot from a new disk. They are introduced through the floppy drive or (less commonly) the modem. The only intelligent time to check for viruses, therefore, is when you introduce new software. A disk from your office mate should be checked for viruses just before you run any program on it. Any file downloaded from a bulletin board should be checked before running.

Timing is important, because running just one infected file can mean that every file in the directory is infected. Change directories, and perhaps every file in the new directory is infected. With boot sector viruses, just one boot from an infected disk can infect memory, and every disk placed in the machine today (and tomorrow and the day after...) will become infected. One virus doesn't just become two. One can become dozens and dozens. A virus can spread through an office like wildfire. Sites that find one infected file or disk are likely to find hundreds and hundreds more, with only a few days between first introduction of the virus and first detection.

Purposes of Detection

One purpose of detection is to simply provide an alarm. If we can know that a virus has been introduced to our system the moment this occurs, we can shut the machine off and go for help. But the help team must know what virus is in the machine, where it is, and ultimately, how to remove it. Good virus detection software can tell us exactly what the virus is and where it is. We can then rely on books and other resources to learn what to do to get it out.

What Must be Searched for Viruses

Any new software, regardless of its source, can potentially contain a virus. There are documented, verified reports of viruses:

- ❑ downloaded from CompuServe and from bulletin boards;
- ❑ shipped in huge quantities with new computers purchased mail order;
- ❑ installed on computers or hard disks purchased from computer stores;
- ❑ packaged in shrink-wrapped software that was shipped from reputable, well-known software houses;
- ❑ repackaged in software that local dealers accepted as returns from customers, then re-shrink wrapped and resold;
- ❑ installed in hard disks sent in to specialty houses for repair;
- ❑ from diagnostic diskettes used by in-house repair and support staff;
- ❑ packaged with shareware anti-virus product that couldn't detect the new "bundled" virus, and placed on bulletin boards, to be used by unsuspecting users.

If the software has been in your machine for months, and you haven't run it, it may contain a virus. After all, a virus only comes "to life" when electricity flows through its body. Like all software, it must be run to do anything.

While old trusted software might contain a virus, we will be more likely to detect viruses if we focus our attention on new software that has not been running in our machines.

Program Change as a Detection Method

The fundamental fact of a virus is that if it is to add its code to another program, it must somehow change that program. You cannot eat a cheeseburger without changing your weight. If you get on a sensitive scale after lunch, you will weigh a bit more than you did before lunch. This is a pleasant fact, because unlike your weight, we do not expect the "weight" of a program to ever fluctuate naturally.

There is one exception to this, unfortunately. Some programs, such as those written by Borland International, permit you to reconfigure them for different colors and other options. When you make these changes, they can be recorded by the program in itself, thus causing some change. But with this exception, programs are stable creatures, and it is only your documents and spreadsheets and databases that should ever change. Since viruses do not get into these files, we can look toward a good method of detecting change in a program as a means of suspecting the introduction of a virus to our system. That is, the only files that should ever change are files a virus does not infect; the only files that should never change are the files a virus might infect, and must change during the process.

Of the hundreds of files on your hard disk, viruses only infect those files that end with the extensions COM and EXE (and sometimes BIN, SYS, OVL, OVR, etc.) Most viruses will change the files they infect in some way, adding code at the top or bottom (or rarely, the middle, too). For most viruses, this will affect the file size, as reported by DOS.

Why Detect Changes?

There are several good reasons.

- ❑ Viruses have great difficulty infecting your machine without making some change in it. To detect a change is to begin the process of detecting a virus. Although some are concerned that a change-detecting program cannot *prove* there isn't already a virus in your computer, the fact is that you needn't worry about this. If you infect your computer with a dozen viruses, then measure its state, one of these viruses will change that state in the next hour or so; a remeasurement establishes that something is afoot.

- ❑ Occasionally things go wrong with computer hardware and software. You run CHKDSK and discover a number of lost clusters in a number of lost chains. You scrap these clusters, but wonder what files you've lost. A proper change-detection program will give you a list of files deleted since your last run. You can then restore them from your backups.
- ❑ In many organizations, we only want to permit the use of authorized software. Using a proper change-detection program, you can establish what software was added to the machine since your last run. Any extra software will quickly come to your attention.

Changes in a Program's Size

One method of detecting a change in our programs is to record their size in bytes, and from time to time to repeat this process, comparing our results with previously recorded sizes. Any growth in file sizes is a sure clue that something is amiss.

A change in the size of a COM file can also provide a useful clue as to just what virus we have. For instance, if your file grows by 1210 bytes, you might have the 1210 virus. Often a virus is named with its size. The 163, 405, 512, 541, 637, 800, 867, 1008, 1024, 1210, 1226, 1253, 1381, 1392, 1536, 1554, 1704, 2608, 3066, 3551, and 4096 viruses are all named for the number of bytes they increase COM files by.

To notice a size increase in a file requires the memory of an elephant, or the storage skills of a disk file. If you record all file sizes now, and then become infected, you should be able to compare the files pre- and post-infection to see which has changed. You could then delete these files and replace them. If done manually, the job is unfortunately too laborious for mortals, and too in-effective to be trusted. Even when we automate this detection of change in file size, we must deal with several significant problems:

- ❑ We cannot use this approach to catch a virus before it has begun to spread, unless we are studying new software on an isolated test machine. If we are given new software to try on our machine, all we can do with this approach is to compare our other programs before and after trying the software. If there is a change, we may have a virus in every file we find has changed.
- ❑ Only about one half of all virus infections are infections of files. In the other half of the cases, the virus infects a file called the boot record (on floppies) or the master boot record (on hard disks). We do not see these files when we type "DIR", and that accounts for how easily these viruses spread.
- ❑ There are some viruses, called "stealth" viruses, that deliberately hide size changes from our view. If we run a 12,345 byte program and it becomes infected with a stealth virus such as 4096, then do type "DIR" to see how large it is, we will see the size unchanged. Actually, the size has changed, but the stealth hides this fact from us by watching from memory and subtracting its actual size from every file it knows it has infected. When the stealth virus is not in memory, you might see that the actual size of the virus has grown to 16,441 bytes.
- ❑ Not every virus infects immediately. For instance, if you run a program containing version 3 of the Icelandic virus, the virus goes into memory and will not try to infect COM, BIN, SYS, OVL, or most of your other files. It only infects EXEs. Watching for changes in COM files won't be very useful. To make matters worse, this virus doesn't infect immediately, but rather every tenth EXE you run while it is in memory. The virus may infect slowly, but it is also very hard to detect by looking for changes in file size unless you have relentless patience.
- ❑ Some viruses overwrite part of your program with themselves. This replacement is not likely to result in any change in the infected program's size in bytes.This does

not mean it is not worth looking for changes in programs. It means that finding no change in the apparent **size** of a program does not guarantee you have no virus.

Changes in a Program's Checksum or CRC

Checksum or CRC comparison programs can be used to detect unwanted change in other programs, possibly the result of a virus at work. There are some small differences between checksum algorithms and CRC (cyclic redundancy check) algorithms. The latter usually uses a table, and is usually a bit slower than the former. Despite the differences, many authors seem to use the words interchangeably, and we will continue the sloppy practice here. Throughout this document, we will use the word "checksum," regardless of whether we mean checksum or CRC.

Each file has a unique fingerprint in the form of a checksum. Changes in any character within the file likely changes the checksum. If a file's original checksum is known — perhaps recorded in a file elsewhere — and its current checksum is known, the two values can be compared.

Any difference indicates that the file has been changed, and offers reason to investigate further. If a program's size is changed, it must be concluded that some modification has occurred to the file. If the size has not changed, some modification is still possible. A file that contains the simple message "Hi Mom!" could be modified so that it contained the message "Hi Dad!", and it would not show any change in size.

A much tougher test of whether a file has been modified is to compute the checksum. At this writing, there are no viruses able to modify a file without modifying the file's checksum. Thus any checksum checker will work just fine in catching viruses, providing that you use it to establish checksums before a virus has modified your files.

How is the checksum computed? Simply adding the values of all the characters in the file is not enough, as a file containing just AE would produce the same result as a file with just EA. Rather, the first byte of a file is read, and an algorithm applied to it. This algorithm does something to the value of the byte, such as rotating the bits a certain number of times, and logically ANDing or ORing the bits to something else. The result of that algorithm is then applied to the next byte of the file. The process is repeated until the final byte is reached, and the remainder is recorded. During this process, different algorithms might be used for different portions of the code being processed. With most procedures, a small file produces a checksum value of the same size as a large file.

Is there such as thing as *the* checksum value? No. The algorithm used defines the result. Consider COMMAND.COM for DOS 3.3 dated 2/2/88 and taking 25,308 bytes. Here are some of the checksums produced for this file by various programs.

- □ BSearch, 16-bit CRC - 13369 (3439 h)
- □ BSearch, CRCTT - 10994 (2AC0 h)
- □ CHKSUM - 20011 (4E2B h)
- □ CRCDOS - 59676 (E91C h)
- □ Delouse, method 1 - 1073916 (1062FC h)
- □ Delouse, method 2 - 1067428 (1049A4 h)
- □ Delouse, method 3 - 1048666 (10005A h)
- □ The Detective, CRC 1 - 26939 (693B h)
- □ The Detective, CRC 2 - 54914 (D682 h)
- □ Module Integrity Check - 24922 (615A)
- □ SSCRC - 52167 (CBC7 h)
- □ Validate, method 1 - 52167 (CBC7 h)
- □ Validate, method 2 - 4024 (0FB8 h)
- □ VCheck - 2141344 (0020ACA0 h)

Kinds of Programs that Detect Changes

File Comparison Programs

File comparison programs work by comparing two copies of the same program to determine if they differ. If the program detects a difference, then there may be reason to believe that one of the copies has been modified — perhaps by a virus.

The approach is not clever, not fast, not efficient. It is actually fairly brainless. It requires that the following assumptions be made:

- ❑ At the time you create the copy, your original program cannot be infected with a virus. If it is, you have defeated the purpose of the approach. Unfortunately, the approach has no way to know whether your original, which you are about to copy, is clean or not.
- ❑ If a virus can infect your original copy of a program, there is a good chance it can find the copy and infect it, too. It is likely that two identical files, infected with an identical virus, will appear identical to a file comparison program. Thus the copies must be stored off-line, in a way that a virus cannot access them. This interferes with any convenience or speed we might wish when we undertake file comparison. In fact, it will normally prove impractical to do such comparisons before running a program.
- ❑ The approach doubles your costs of storage, because it requires that you store two copies of a program for every copy you plan to use.

We do not recommend this approach, except in certain special circumstances. Keeping good backups, of course, we do recommend.

Adding Self-Checking to Programs

The most efficient approach to checking files is not to check only critical files, or all files, but rather to check files as they are run. This checking can be done with either code which is added to each file, or with a memory-resident driver, that monitors file access.

To add self-checksumming code to a program, the program must be modified so that when it is run, control is first passed to the added code, which then calculates the checksum of the file with the checksum that was stored in that file earlier. A failed comparison can result in an alert to the user.

There are a few drawbacks to the approach:

- ❑ it slows processing a small amount;
- ❑ it enlarges each file a small amount;
- ❑ it may not work on COM files that are nearly 64K in size, since 64K is the largest size supported by the COM format;
- ❑ it cannot work with BIN, SYS, and OVL files;
- ❑ it cannot work on archives (such as *.ZIP or *.ARC or self-extracting EXE files);
- ❑ Some products — particularly those from Borland — modify themselves when a user changes parameters or exits. A vaccine applied to such a product would generate spurious false alarms with such programs. Since very few programs are currently self-modifying, this does not seem to be anything but an occasional nuisance with the approach.

Despite these shortcomings, we think that the idea of adding self-checking to your files is a good one. One day, perhaps, all of the software we purchase will include this feature. Until then, you can add self-checking to your trusted software with program's such as Skulason's F-XLock, part of his F-Prot package of anti-virus software.

Some anti-virus programs are able to detect that they have become infected, display a warning message, and exit. Examples include HTScan, Norton AntiVirus, and Skulason's F-Prot. A few others, such as McAfee's Pro-Scan, perform a self-check, and if it finds a virus, can remove it from memory and itself before proceeding.

Checking Groups of Files

Theory of Operation

Checksum programs usually create a single master database file listing programs, their directories, and one or more initial values. When they are run a second time, they can compare their new results against the old list, and report. However, some create a small file of information for every program they check. For instance, Norton Anti-Virus does this, a feature that has both fans and detractors. The benefit of this feature is that a memory resident program can very rapidly check for changes in the file you are about to run, before you run it. This means that you do not need to check the entire hard disk every morning before getting down to work. You can simply run programs as you normally would, and if any has somehow become infected, you will know before it loads.

Not every approach to storing the checksum information is equally elegant. One program creates a file with the extension XUP for every file it finds. Thus GO.BAT and GO.COM get a total of one XUP file, named GO.XUP. This program, therefore, will provide a false alarm that one of your two GO files has been modified.

You must know that a stealth virus might be able to defeat a checksum program if it loads into memory before the checksum program runs. The stealth virus can then detect the checksum program as it attempts to read each program on the disk, and — before letting the checksum program see the file it is trying to read — extracting the virus from it. After the checksum program is satisfied that there is no virus in the file, the virus in memory can re-insert it into the file just checked. Such a problem can be easily avoided: simply boot the system from an uninfected floppy, then run your checksum program from it.

What to Look For in Checksum Software

Ease of Use Some programs are substantially easier to use than others. The ideal program does not even require a manual to be run correctly out of the box, yet is accompanied by a clear manual that provides cogent instructions for basic and advanced use.

Number of Techniques The program should compute checksums using two different approaches, or compute both file size and checksum, to ensure that a virus doesn't modify a file in such a way that the checksum isn't changed. Gilmore Systems has a program called PROVECRC that creates a modified version of a file that is different, but that has the same CRC as the original. The program proves that a single CRC is not fool-proof for virus detection, for it is possible to write a virus which can add code to your programs without changing the CRC. When two algorithms are used, PROVECRC creates changes undetected by one, but detected by the other.

Scanning of Critical System Files On an MS-DOS hard disk, there are five critical system files that are read during the boot process: the Master Boot Record (containing the partition table), the boot record, two hidden system files, and COMMAND.COM. Because many viruses take up residence in the Master Boot Record, boot record, or COMMAND.COM, it may be desirable to check these files on each boot. Not all checksum programs, however, can check all of these files. While viruses rarely touch the two hidden system files, and many do not touch COMMAND.COM, quite a few get into the Master Boot Record of the hard disk or boot record of floppies.

Complexity of Checking Algorithm A 32-bit checksum is potentially harder for a virus to beat than a 16-bit checksum.

Speed when Checking Files From time to time, it may be desirable to check all files on the hard disk for changes. However, if this process takes a long time, users will not do it as often *as they* should. What is the speed of checking files?

The most elegant solution to the speed problem is to never require a user to check all files, but rather to always permit the user to perform an automatic check for change whenever he or she runs a program. This doesn't waste the user's time checking for changes in programs that will not be run today, and it doesn't allow any fatal delays between checking and running. Norton AntiVirus and Skulason's F-Prot both permit run-time checksumming.

For most CRC checkers, it is not the number of files, but the number of bytes, that determines the overall speed of operation. Thus the speed of your checker, per file, will be affected by file size.

Efficiency in Checking All Files Does the program permit checking of all files with some option such as /ALL, or is it necessary to feed the program a list of all the files you wish checked? The latter approach can be grueling for any user with a large hard disk! Is there an upper limit to the number of files that can be checked? Is the program smart enough to check other logical drives, such as D:?

User Control of Files to be Checked Because checking all files can take some time, users may wish to provide the program with a list of files to be checked. Can this be done? Can the user use their text editor or other convenient tool to build the file list? If the program checksums at run-time (as with Norton and Skulason's programs) this option is not needed.

Optional System Lockup on Detection of Modification Many things can modify a program: a virus, a hacker, an error in using a sector editor. If a program has been modified, do you want to try to run it? The smart money says no, let's stop right now and see what has happened here. Running any program that contains a virus is certain to spread the virus. It might be desirable if the system is able to prevent any modified program from running.

Self-Protection of Checksum Program If a checksum program becomes infected, it then puts the virus into memory before it begins to run. A stealth virus in memory is able to remove itself from any file as the file is checksummed, preventing the checker from finding the virus. Thus we need some notification that the checksum program has been infected. Ideally, the checker reports that it has been infected and quits running.

Checks of Hidden Files This is important if the program is to protect your operating system and any programs marked with these attributes.

Working from a Floppy You might want this if you wish to use the program to monitor another user's machine, for instance, to see if they are clandestinely running a golf game that is not on the approved corporate software list. It is also valuable for guarding against stealth viruses and for protecting the program, through write-protection and removal from the machine, from attack from a virus.

Updates to the Database Do you have control over whether the program updates its baseline database? If the program updates this every time it is run, you will lose your history file.

Searching for Specific Patterns: Scanning

About Scanners

Virus Scanners are products designed to help identify viruses within files, boot sectors, partition tables, memory, and other hiding places; to name them; and potentially to help remove them. Scanners cannot be your only approach to maintaining a virus-free environment, for they are destined to failure, and can be time-consuming to run.

Nevertheless, the scanning job is an important one, and it is reasonable to wonder whether the scanner you are using is adequate for the job. It is reasonable because of the astonishing claims made by some vendors. The NCSA has reviewed many anti-virus products, and has often been disappointed. For instance, one vendor bragged that their product "is now on its third upgrade and well into its second year on the market. The product has had a chance to mature and grow in order to satisfy end-user's needs. We are also pround (sic) to announce that [truly big and famous company] has recently signed a corporate license agreement with us to use [product name] internally." Unfortunately for this truly big and famous company and other shoppers, the product was only able to detect viruses in four of 95 test files we fed it, each of which was infected with a different virus.

We believe that ability to detect a virus, even if the name provided is not quite in keeping with local custom, is probably the most important feature of a scanner. Assuming a choice between two scanners which detect every virus ever developed, we would then choose the most precise or fastest, providing it made a clamor or displayed a message on the screen when it found something.

Why a Scanner: Pros and Cons

Many users swear by scanners as a means of helping to detect viruses. Others seem to swear at them. Here's a quick look at their pros and cons.

Pros

A scanner is critical for providing a precise identification of a virus in your system. Without a precise identification, we cannot know the best course of action for removing it, and we cannot be sure we have removed it completely.

Unlike the "change checkers" described previously, or the TSRs to be described in the next section, the scanner is the only kind of anti-virus program that can spot a virus before it has run. Thus if you use it to check new software before running it, you can prevent even one of your files from becoming infected.

Cons

Not all vendors tell the truth. I was once shown a scanner at a trade show that "defeated all known viruses." I asked if it defeated X, Y, or Z, pulling a few names from my hat that I thought they might not have heard of. No, no, no were the replies. "What does it catch", I asked. "DataCrime" was the reply. "Which version?", I asked. The programmer wasn't sure. But I was assured that I could add scan codes, to catch other viruses. "Where do I get those scan codes?", I inquired. The programmer was not sure. The moral of the story is that a scanner might not catch quite as many viruses as the marketing folks would have you believe. Be sure that the scanner you are about to purchase catches the viruses you have some reasonable probability of getting this year.

Scanners are always out of date. Most scanners are usually somewhat incomplete when they are released. No virus researcher has access to every virus on earth. But even if your scanner was absolutely current one month ago, and caught every virus that existed on earth the day it was released, there are now 60 or more viruses that it cannot catch. While your odds of getting a brand new virus are very, very small, you must recognize that scanners will not catch everything.

Some scanners may not clean from memory the scan codes they use in searching for viruses. If you run two scanners in an uninfected machine, you have some chance of a false alarm when running the second scanner.

What a Scanner Might Look For

text strings

Some viruses contain specific unencrypted text strings. You can look for some strings (perhaps with TextSearch from the Norton Utilities) and get reasonably positive identification of some viruses, especially if you know where in the file the string is supposed to be found. For instance, the message "@AIDS" can be found in the first five bytes of any program infected with the Lisbon virus. If you don't find this string in the file in this position, your program does not have this virus. Looking for strings has a couple shortcomings:

- ❑ It can only be used for some viruses. For many, there is no text message at all, and for others, the string is encrypted so that it cannot be read by a text searching program.
- ❑ Knowing the message contained in a virus does not always help us identify it when it is running. There are viruses that contain unencrypted messages, yet never display them. For instance, the Sunday virus contains the message *"Today is SunDay! Why do you work so hard? All work and no play make you a dull boy! Come on! Let's go out and have some fun!"* However, this message is set to be displayed on the 7th day of the week in a counting system in which days are numbered 0 to 6. Thus you'll never see this message on your screen if this virus is running. Other messages may display on the screen, yet are encrypted so that we cannot see them when we look directly at the virus code. For instance, the Devil's Dance virus will display the message *"Did you ever dance with the devil in the weak moonlight? Pray for your disks!! The Joker."* Yet you will not find this in the virus itself, using a program such as Norton's Text Search (part of the Norton Utilities).
- ❑ For all viruses, the message is less important than the rest of the program. Changing the message does not constitute much of a change in the virus; changing the real virus code may result in a significant difference in how the virus works. Thus two samples with different messages might behave identically, and two samples with identical messages might behave very differently.

Nevertheless, the messages in a virus are sometimes interesting, and sometimes helpful in identifying just what you have. For instance, finding the sequence "Eddie lives." suggests you may have the Eddie-2 virus, although you might have Dark Avenger instead; In the Ashar virus, you will find this message hiding in your floppy's boot record: "Welcome to the Dungeon (c) 1986 Brain.& Amjads (pvt) Ltd VIRUS_SHOE RECORD, v9.0 Dedicated to the dynamic memories of millions of virus who are no longer with us today - Thanks GOODNESS!! BEWARE OF THE er..VIRUS : \this program is catching. program follows after these messeges" You can learn a bit about the authors by a look at the message content, syntax and spelling.

hex (binary) strings

It is sometimes said that a program is stored in "binary". As with most assertions, this is not quite true. When you try using the TYPE command to view one of your COM or EXE files, what you are looking at is tokens, or symbols which can be translated into instructions your machine can understand. A total of 256 different symbols can be used, sometimes called "the extended character set", ASCII, or ANSI. When these characters are stored, they are not stored as you see them, of course, but rather as magnetic patterns of positive and negative pulses, according to rules followed by your hard disk or floppy controller.

By using a base of 16 rather than 10, it is possible to refer to each of these 256 values using just two characters. The 256 values can be assigned numbers, which can range from 0 to 255 in decimal (base 10) or from 00 to FF in hexadecimal.

A program that is looking for a virus can look for some unique sequence of code within each of your programs. In the case of the Jerusalem virus, we are likely to be able to find the 4 byte hexadecimal pattern 8E D0 BC 00 07 50 B8 C5 in many of our different strains.

Using a short scan code can result in some confusion, as a scanner can call each of several viruses by the same generic name. Using longer scan codes poses a different problem, however. While it ensures that we have precisely identified the virus, a virus that is slightly different might not be detected at all! So a clever scanner might look for a simple common string, and then if found, look for additional information from its database of known strings. If one is found, it can identify the virus precisely. If only the simple common string is found, then the scanner might confess as follows: "Jerusalem-related virus detected in file."

Many viruses encrypt parts of themselves when they infect files, in order to defeat the scanners that are looking for them. Each encrypted copy of the virus looks a bit different to a scanner, and only the decryption algorithm provides a solid clue to the scanner as to what virus is here. Because decryption algorithms can be legitimate code, this becomes a difficult problem. The Fish-6 virus is encrypted, but you can still find it by looking for 7 or 8 byte patterns.

Using a Scanner

If you plan to use a scanner, you must follow these instructions if you hope to achieve success in detecting viruses:

- ❑ Cold-boot the machine to be scanned from an uninfected write-protected floppy disk. Because a virus cannot get past the write protection mechanism of a floppy drive, you are thus assured that no virus has moved into memory, from whence it can defeat your scanner.
- ❑ Run the scanner from the disk you just booted from, and do so before running any other programs — especially programs that are on a drive other than the one you just booted from. Any device driver called by your hard disk's CONFIG.SYS can contain a virus.
- ❑ Scan the disk you just booted from, to ensure that it is not infected.
- ❑ Watch your screen carefully during the scanning process, if your scanner does not pause when it detects a virus.
- ❑ Follow the instructions for your scanner carefully. Failure to issue the correct command line may result in a partial scan, and you might inadvertently skip the scanning of a directory which contains a virus.

What You Should Look for in a Scanner

Virus Recognition

How well do the scanners recognize a virus in your files? Differences between the products can be significant. With so many viruses on the loose today, we do not recommend any scanner that detects fewer than 100 viruses. On the other hand, carefully avoid choosing a scanner simply because it claims to be able to recognize a zillion viruses. You aren't going to find a zillion viruses in your machine. If it correctly identifies each instance of the two dozen most common viruses, you can sleep well at night.

Virus Identification

Identification Accuracy No scanner catches everything, and when two scanners do detect the same virus, it is quite likely that they will report it by different names. One scanner might find "4096", another "Frodo", and another might find "4K". You don't have three viruses, though. You have one that travels under different names in different parts of the world.

Some scanners report false alarms. For instance, one scanner we tested somehow managed to find the Sunday virus (1,813 bytes of virus) in a 3 byte test file and managed to find boot sector infectors in 512 byte files containing a copy of this sector. Another hung up the system repeatedly when scanning a file that had been compressed, then infected.

How successful a program is depends in large measure on the scan codes used. When short codes are used, many strains may be detected as being the same; when overlapping codes are used (code 1 a subset of code 2), a scanner might detect two viruses in one file. This is the case with a number of scanners.

You will want your scanner to identify viruses accurately, and to provide as precise names as possible. With accurate, precise identification, you can take appropriate recovery steps.

Boot Sector Virus Detection Some viruses infect the boot sector of floppies, and the boot sector or master boot sector of hard disks. Such viruses can cause as much damage as their file-based counterparts, and must be detected by our scanners if we are to use scanning as a defense against them. Some scanners cannot check boot sectors or master boot sectors. Make sure the scanner you choose can!

We tested one scanner that reported boot sector infectors " may be non-working, damaged, or dead viruses." In our testing, this was not true, and we think it is misleading and therefore a potentially dangerous message. Make sure the scanner you purchase identifies living viruses as living viruses!

One scanner we tested from the command line took a bit over 8 minutes to scan a single floppy with a boot sector virus. Running it from the menu, and disabling checks of memory and files was substantially faster, but no more effective for detecting these boot sector viruses. For a subsequent version of this scanner, detection of a boot sector infection was nearly instantaneous, and quite thorough. From this experience, we must conclude that we cannot talk accurately about anti-virus products without also referencing the version we are talking about.

Detecting Common Viruses If you are a scanner, you have only one mission in life: to detect every virus on a drive and in memory. Doing it fast, and making it fun might also be desirable. Since scanners cannot detect every virus (a new one will be introduced in the time it takes you to read this booklet), then they must, at least, meet another goal: to be able to detect every common virus.

Common viruses include 4096, Ashar, Cascade-A, Cascade-B, Dark Avenger, Invader, Jerusalem B, Joshi, Ping Pong B, Stoned 2, Sunday, Taiwan 4, and Yankee Doodle. You can check with your vendor to see whether the product you are considering catches these viruses.

Detecting Stealth The stealth virus is one which is able to hide the evidence of its presence from curious onlookers. Some can infect a file, but when memory resident, remove themselves from the file when the DIR command is given, so that the infected file looks normal in size. Such a virus can also remove itself from the file whenever a scanner takes a look at the file. A common stealth virus is the 4096 — also known as the Century Virus, IDF Virus, Stealth Virus, 100 Years Virus, Frodo Virus, and 4K virus. Can your scanner detect a stealth virus when it is memory-resident?

Virus Reporting Functions

Messages Does the program deliver useful messages? Does the program pause after displaying a screen full of messages?

Program Output to File or Printer Can you send program output to a file, listing the infected files and the nature of the infection? This is valuable if you have a large number of infected files, or wish to use a hard copy report in a memo or other document.

Efficiency

Speed How long does it take for the program to scan? Rapid scanning is desirable if it is to be done frequently, assuming that accuracy is not compromised. Because a typical 33 Mb hard disk may have 1500 files or so, time required to scan a few files becomes very important. While the difference between 2 minutes and 2.3 minutes won't matter to most users, the difference between 2 minutes and 10 certainly will.

Running in Batch Mode Can you run the scanner from a batch file such as AUTOEXEC.BAT? To do so successfully means that the scanner needs to have a mode in which it can be driven by a command line, and need not have any keyboard input during operation. It should be designed to provide warnings or useful error codes if a virus is detected, or pass control smoothly to the next file in the batch file.

Efficiency in Scanning Removable Media After an infection of a hard disk is detected, it is prudent to scan everything in sight, which likely includes a hoard of floppy disks. Is there a built-in option to scan removable media efficiently? Efficiency means that memory would only need to be scanned once, the program loaded only once, and the command to scan another would be only a keystroke or two.

Scanner Self-Check

If the scanner becomes infected, does it report this to the user, and refuse to proceed?

Coping with New Viruses

Can you add virus signatures to a list read by the scanner, to keep it up to date? If so, you won't need to buy an update every few days. The ability to add signatures is desirable if you have a good source of trustworthy, tested signatures. For instance, if your vendor will provide these signatures at no charge, via fax, phone, or BBS, then you can keep your scanner "fresh."

In adding signatures, we would like to be able to:

- ❑ provide very long signatures if possible (to be more confident that a detection is real).
- ❑ provide signatures in which one or more bytes may take any value (wildcards), to be able to efficiently detect self-encrypting viruses or minor variants.
- ❑ add comments to your file.
- ❑ have our signature file, itself, checked for unapproved changes.
- ❑ specify byte offset to search (to improve scanning speed).
- ❑ indicate the type of file that this string might be found in (again, to improve speed).
- ❑ indicate whether this signature will be in memory, in a disk file, in the boot record, in the partition table, etc.

Price

We don't deserve viruses, we shouldn't have to pay an arm and a leg for scanners. What are the terms and conditions of these products? Is volume pricing available? Site licensing?

Memory-Resident Scanners

Most virus scanners are stand-alone programs ending with an EXE extension. To scan a file or group of files, you manually run them. But scanners can be memory resident, and monitor your computer's activity. Such a program can load when CONFIG.SYS or AUTOEXEC.BAT run, and might monitor for viruses during COPY or when a program runs.

The approach has great potential benefit for users who will not always elect to manually scan new software for viruses. However, the program may conflict with other software, or occupy so much memory that the user's other software does not work.

Watching for Specific Behaviors: The TSR Approach

A TSR is a program that loads into memory, then becomes inactive, allowing some other program to run. Certain keypresses will activate or de-activate a given TSR, and some permit a keypress to remove them from memory. Popular TSRs include SideKick, spoolers, cache memory, and various pop-up menu systems, help systems, appointment books, calculators, and other desktop utilities.

An anti-virus tool can be written as a TSR. As such, it can monitor the behavior of your other software, and activate whenever the other software attempts to do certain things. For instance, if a virus in program X attempted to spread, a file write could be detected by an anti-virus TSR, and the user warned.

While the idea is a good one, and TSR programming shows a fairly high degree of cleverness, the actual implementation problems leave the approach inferior in many situations. TSR anti-virus software is burdened with all the problems of TSRs in general, and with a few additional problems. Shortcomings include:

- Many microcomputer configurations are intolerant of TSRs, and crash, lock-up, or misbehave. The factors which interfere are numerous, but usually involve a conflict with other TSR's the user loads.
- All TSRs use RAM, and most use conventional memory — addresses below 640K — something which is in very short supply in today's micros. The addition of another TSR to your system may interfere with the operation of some other large program, such as Ventura Publisher.
- Disk writes are common, normal, and normally desirable. The virus attempting to write to disk may constitute a very small fraction of legitimate disk writes. Any disk write monitor which provides the user with many false alarms will be seen as a nuisance, and likely be deactivated long before a real virus tries a nasty disk write. The TSR disk-write monitor provides the software equivalent of the boy who cried wolf.
- Any anti-virus TSR can, in theory, be easily detected by viruses, and disabled or removed. Most viruses, of course, do no such thing, but viruses of the future are likely to attack TSRs if this means of virus detection proves popular. The result will be users with a false sense of security.
- A disk write monitor slows the system, as each call is intercepted and processed.
- A TSR may not be able to prevent a boot sector infection if the infection occurs before the TSR is even loaded, as in an accidental boot from a data disk in A:.
- It is simple to write a virus that evades the interrupt vectoring capabilities of a TSR in a DOS environment.

Because of these limitations, we are not particularly enthusiastic about this approach.

Disk Mappers

Disk-mappers (also known as picture takers, fingerprinters and signature checkers) maintain centralized data files that consist of coded disk images. These programs notify users when changes are discovered between target disks and their coded images. There are cases where this is a brilliant approach to virus detection. For instance, the approach is sound when verifying that disks which are about to be shipped from a software house are exactly as they were meant to be. However, in everyday life, the approach has serious drawbacks, including:

- ❑ The output files from this approach can occupy vast amounts of disk space, and increase in direct proportion to the number of files being tracked.
- ❑ If the target disks are in a state of flux — the normal condition in our computers — time consuming maintenance of disk-mapper data files is generally required. As target disks change, entries must be sorted, updated, deleted or purged.
- ❑ Other problems: sometimes complex and difficult to operate, sometimes they slow the boot process, some use memory resident code to intercept DOS calls, and may conflict with other TSRs.

We generally don't recommend the use of such products, and there are few on the market.

Using a Test Computer

If you are going to take software evaluation very seriously, you might create a software testing machine. On this machine you can run software and look for viruses, as well as looking for all the other things we must look for when evaluating new software: does it work as we need it to? is it easy to learn? easy to use? etc.

It is a good idea, but usually impractical, to install new software on a separate test computer, to run it there, and make sure it is virus-free before installing it on your big hard disk or network server.

If you were to try the quarantine approach, you would back the system up, run some checksum program (such as Norton Anti-Virus, Advanced option), then run the new software and some of your old, trusted software, then look for changes in the size or checksum of your trusted programs. Looking for such changes is best done if you boot from an uninfected floppy, as some viruses can mask such changes when they are in memory.

The approach is reasonable when you are testing software for an organization. But when you are about to use software on your home computer, few home owners want to invest in a two computers just to test software for one.

If you will have a special test computer, it can either be temporary or permanent.

Temporary Test Machines

There are two ways to create a temporary test machine:

- ❏ Swap hard disks, placing a test hard disk on the test machine and disabling whatever hard disks may be on the machine;
- ❏ Run a checksum evaluation of every file on the hard disk. Completely backup the machine to tape. Then test the software, now run the checksum comparison again and scan for viruses. If any are found, you can turn the machine off at the switch, boot from a write-protected floppy, delete the infected files, and restore what you need from tape.

Swapping Hard Disks

Testing software on your favorite computer, with its 6,000 zigabyte hard disk is perfectly safe. If you have an AT, you will want to learn your drive type. Do this by running SETUP and reading.

Record the drive type number — something in the range of 1-47. You will find SETUP on the DOS disks which came with the computer. If you can't find this program, don't proceed, because you'll need to run it for your AT to install your temporary, test hard disk, and you'll need to run it to get your original hard disk working again.

Now simply park the heads, turn it off, remove the cover, unplug the power connector to the drive and the two cables from the controller to the drive. (If your drive is an IDE drive, you'll only have one flat cable going to it.) You can probably do this without removing the drive from the computer. Your next step is to add a hard disk that you will use for testing. I'd recommend something like a Seagate ST-225, a 20 mb hard disk that you can buy for about $225, and which you may have available from an old XT. This drive will work in an XT, 286, or 386-based machine. On an AT, you'll need to run setup. The ST-225 is probably Type 2. After you have installed this hard disk, you are ready to roll.

The safest approach to testing new software would follow these steps:

- ❑ Select a test machine that is similar to the one on which the product will be run, if it is not the actual machine. If you have a machine shortage, you can probably acquire a small hard disk that you can use for testing only, and simply swap this drive for the one connected to the real machine. More on this below.
- ❑ Scan the machine for viruses in the boot sector and all files.
- ❑ Backup all files on that machine, making sure you can restore from the backup.
- ❑ Install the new software, and test it in the normal way.
- ❑ Scan again for viruses. If you use a quality scanner and can find no viruses, it is safe to assume there are none in the copy you installed.

- ❑ If you will be installing multiple copies of this software, use the disks you just installed from. Assuming you purchased multiple copies of the software, place those other copies on the shelf. They may contain a virus, even though the one you tested did not.
- ❑ Be sure to place write-protect tabs on the install disks, to prevent a virus from a user's machine getting onto the disks. Be sure to boot their machine from their own hard disk, rather than your install disk, to prevent an undetected boot sector virus from your disks infecting the user's hard disk.

Permanent Test Machines

Much more glamorous than a temporary test machine is a permanent machine, dedicated to the glamorous mission of serving as the test unit for all software that enters an organization. Here are the credentials of the dream test machine:

- ❑ This machine ideally has a small hard disk that can be reformatted at will, as this is often the easiest way of fully purging viruses that are detected.
- ❑ The machine is capable of running any software that will be tested. This will require that it be fully compatible with other office machines.
- ❑ All diskettes placed in this machine are a different color than those used in the office, and are labeled with a warning that will help keep them out of the drives of other machines.
- ❑ The machine is not connected via modem or cable to any other machine anywhere.
- ❑ The machine has a simple but versatile printer connected, for doing screen captures and recording notes.

Confirming Your Anti-Virus Software's Hunches

Not every suspicious activity turns out to be a virus. You can do much to help ensure that there are a minimal number of false alarms, so that you have more energy to respond when there is a real problem.

False Alarms

Here are some common false alarms:

- ❑ *My machine won't boot.* Sorry. This is quite likely a hardware failure. Viruses rarely do this much damage. If this is Monday morning in the winter, and your hard disk is 2 years old, its bearings may be worn. Let it warm up by leaving the machine on for an hour or so, then turn it off and back on. You may be back in business.
- ❑ *I saw a "1701" on the screen when I turned it on. Isn't this the "Cascade virus"?* Nope. 1701 is the number of the Starship Enterprise, and the error code you'll see when your hard disk subsystem is not feeling up to par, during the power-on self-test. Any number of hardware problems might be the cause, but this is not a virus, because we have not succeeded in reading one byte from the hard disk yet.
- ❑ *I tried booting and it told me that I was using the wrong version of Command.com. A virus?* No. A virus won't modify COMMAND.COM so that any error message is produced. And most modern viruses are smart enough to not infect command.com at all (to help escape your watchful eye.) You might have copied some diskette to your root directory during your last session, and copied over COMMAND.COM. Replace it with the correct version, and your problem will likely go away. To avoid the problem in the future, you set its attributes to Read only, which will prevent it from being accidentally copied to.
- ❑ *I just ran CHKDSK, and I was told I had lost clusters and file allocation errors. I also had a cross-link. Surely this is a virus!* No, probably not. If you haven't run

CHKDSK in some time, or haven't corrected such problems with Norton Disk Doctor or CHKDSK /F, then the problem is normal if unfortunate. Most undoctored machines have some problems like this, which can be caused by dBASE, Windows, or just about any program crash. A few viruses cause progressive cross-linking, but you are not likely to have one of them. Keep your eyes on the machine, run CHKDSK or Norton Disk Doctor more often, and see what you can learn.

□ *I just discovered some bad sectors when I ran CHKDSK. Is this a virus?* Probably not. Some boot sector viruses will place programs in certain locations, and mark those sectors bad in the FAT. For instance (Stoned...) However, nearly every hard disk will have 1-5% of its surface bad. It is probably nothing to worry about.

□ *I tried copying a file, but got a message "Error reading drive" or "Sector not found." A virus?* Not at all. Chances are your drive is due for a low-level format. You can run Norton Disk Doctor, Norton Disk Test, SpinRite, HDTest, Calibrate, or any other non-destructive low-level format program and correct the situation.

□ *I just ran my favorite old scanner on my new 200 Megabyte partition, and found a virus in the Master Boot Record! So now tell me I don't have a virus.* Ok, you don't have a virus. What you might have is anti-virus software that doesn't recognize the strange look of a large partition. You can prove it by trying to infect a standard floppy: Just put an disk with no write-protect tab in drive a:, do a DIR A:, and it should be infected. Now run the same scanner against A:. Do you have a virus? If not, its time to upgrade your scanner to one of a more recent vintage, that knows about large partitions. A newer scanner will also happen to know about newer viruses.

□ *I was using my favorite program when I noticed that the letters were dripping down the screen. A virus?* Maybe. That one might be Cascade, or might be a little joke program called Drippy that your office mate is trying out on you. We'll need to run a scanner to be sure (or perhaps look in your autoexec.bat for something like drippy.com.

- ❑ *I just got the message "Your PC is now stoned! Legalise Marijuana" when I turned it on. Is that a virus?* Yes. That's the Stoned/Marijuana/New Zealand virus. And if you had a ping-pong ball bouncing on your screen, you probably had Italian/bouncing ball. And if you found a black rectangular hole over on the screen, you may have Jerusalem. And funny colors to the letters you type in DOS, might be Devil's Dance. None of these things seem to be the result of some component no longer working (hardware failure), but of some strange additional controlling force. If its not you, and not the programs you have grown to love, it may be some new software that's making things happen.

If you got stumped on some of the previous questions, then you may want to study some more about hardware problems and recovery from them. Some books are suggested at the end for your reading.

Real Symptoms

You can also sometimes spot one or more of these symptoms, which are not likely to be caused by hardware failure:

- ❑ Changes in the length of programs. Any program that is larger (by a few hundred bytes or more) than another copy of it has possibly been infected by a virus. Viruses never make programs smaller.
- ❑ Changes in the time stamps. Most viruses won't affect the date or time stamp, but a few do. If you see a date stamp on a COM or EXE that is more recent than when you installed it, you can be suspicious.
- ❑ Longer loading times. If a program seems to take a bit longer to load than normal, it may be that a virus has infected it. Check to ensure that nothing has changed to your computer's configuration that might account for this.
- ❑ Slower operation. Some viruses will slow a program after it has loaded. You can look for this as a possible clue.

□ Unexplained drive activity. When a virus tries to infect files on your floppy, you will see the floppy drive light come on. If this light comes on for no apparent reason (the program you are running isn't doing this), a virus may be trying to infect it. If you get an error message because the floppy disk is write-protected, be extra suspicious. While most viruses are able to trap such error messages, some will permit them.

□ Unexplained reduction in available memory. Running CHKDSK will show you how much memory is available. This value should be the same whenever you boot. If you find less, it may be due to a memory-resident program you have run or possibly a virus.

□ Bad sectors on your floppy. If CHKDSK tells you that you have bad sectors on a floppy today, and you did not have bad sectors yesterday on that floppy, a boot sector virus may have written some code to those parts of the diskette, and marked them bad. You might have a boot sector virus.

□ Programs disappearing. If you run a program that was in this directory last time you looked, but get "Bad command or file name" when you run it now, a virus might have deleted it. Run some other program that you can see in that directory, and see if this error message repeats. If so, and both have been erased, you may have a virus.

□ Unexpected reboots. If your micro unexpectedly reboots after running your favorite program, or when you are doing nothing, a virus might be playing tricks.

□ Odd screen action. If you see characters dripping from the screen, balls bouncing about, ambulances driving across the horizon, screen colors changing, a black square on the screen, an unexpected message, or anything else unusual, you may have a virus.

In general, a virus may be responsible for a wide variety of odd computer behavior. One strong symptom (such as a message) or two or more weak symptoms (such as a reduction in memory and an unexpected drive light on) should get you busy investigating.

Symptoms of Common Viruses

Virus Symptoms	
1451	Adds 1,451 bytes to COM files, 1,411 bytes to EXE files.
1554 (Valert)	COM and EXE files grow larger. FAT corruption will result in errors such as lost clusters. The system may hang.
4096	A "Stealth virus"
Aids	When activated, displays "Your computer now has AIDS." The word "AIDS" covers about half the screen. Following display of this message, the system halts and must be rebooted.
AirCop	Your computer may stop running. You'll find a decrease in available and system memory. On most systems you'll see "Red State, Germ Offensive. AIRCOP." from time to time. On less compatible systems, the system may crash with a stack overflow error.
Ashar	You can find this text in the boot sector: "Welcome to the Dungeon (c) 1986 Brain.& Amjads (pvt) Ltd VIRUS_SHOE RECORD, v9.0 Dedicated to the dynamic memories of millions of virus who are no longer with us today - Thanks GOODNESS!! BEWARE OF THE er..VIRUS : \this program is catching program follows after these messeges". This message is never displayed.
Bloody!	Virus displays message the first time between the first and 128th system boot, then every six boots after that. Due to bugs in the decryption of the message, it may display as garbage.
Brain	Booting may take longer than normal from floppy. Use DIR or VOL to find a volume label on an infected floppy: (c) Brain.

Virus Symptoms	
Cascade	In the original version, if the system month is between October 1 and December 31, the system year is either 1980 or 1988, and the monitor is either CGA or VGA, a cascade display will be activated at random intervals. In subsequent modifications, any month and year will do. The cascade display consists of characters falling from the screen, landing and remaining on the bottom line. The virus spreads well because these symptoms are not normally displayed.
dBASE	None.
Den Zuk	In the most common version, a pretty purple DEN ZUK graphic slides across your screen after a CTRL-ALT-DEL is performed if the system has a CGA, EGA, or VGA monitor and an infected floppy in drive A:. The virus is not removed from memory with Ctrl-Alt-Del; a reboot is simulated when these keys are pressed. There are many variants which display a different message, in part because of the availability, in Indonesia, of a program to modify the message.
E.D.V.	System may crash sporadically.
Fellowship	In September, when an infected file is run, you will see the message: "This message is dedicated to all fellow PC users on Earth Toward A Better Tomorrow And a better Place To Live In 03/03/90 KV KL MAL."
Frere Jacques-A	Plays the tune "Frere Jacques" on Fridays. COM and EXE files become larger. System may crash. Available memory may decrease.
Icelandic-3	The message "Gledileg jol" ("Merry Christmas" in Icelandic) may appear on December 24. If this virus is resident on that date, you may find no program will run.
Jerusalem	One half hour after activation, a black 2 by 12 rectangle will appear in the lower left of the screen. The system will slow at this time, too.
Joshi	Joshi activates on January 5 of any year, displaying the message: type Happy Birthday Joshi on a green background. If you do so, Joshi will let you proceed. System may hang. CHKDSK will report 6K less memory than before Joshi became resident.

Virus Symptoms	
Keypress	Every 10 minutes, the virus examines the keyboard for 2 seconds, looking for a keystroke. If a keystroke is pressed, the key is repeated, with the number of repetitions driven by the duration of the press. Thus the virus simulates a sticking key. The date/time stamp of files infected is set to the current date.
Microbes	Machine may not boot from an infected disk. Once in memory, Microbes may spread to other disks which are inserted. After some number of boots/infections, may display some "credits" during the boot process. A stealth virus.
Murphy-1	System may crash during attempts to infect EXE files. If the virus is loaded between 10 and 11 AM, the speaker is turned on and reset on every DOS function call, making a shuffling or clicking noise. COM and EXE files become larger. Not particularly harmful.
MusicBug	Some infected systems play a few notes with every DOS command issued. On others there are no notes but there is a lot of I/O of write protected disks.
Payday	EXE and COM files become larger. The system may slow down. Files may be deleted on any Friday but Friday the 13th. A black square may appear on the screen.
Perfume	You will be asked for a password after 80 uses of an infected program. If you do not enter the correct password ("4711") your program will not run.
Ping Pong	A bouncing ball or dot may appear on the screen upon activation. This display is stopped when the system is rebooted. You can activate it by entering TIME 0, then pressing any key and then the Enter key. CHKDSK will report one cluster bad (usually 1K on floppies.)
Print-Screen	Due to bugs, this early version of this virus was unable to do anything related to the printer. Subsequent versions may occasionally perform the PrintScreen operation, triggered by disk reads.
Saddam	Will display the message "HEY SADAM LEAVE QUEIT BEFORE I COME." The message is displayed on every 8th infection.
Slow	According to some reports, may slow the system.
Stoned	The screen will sometimes display "Your PC is now stoned!"

Virus Symptoms	
Sunday	In version A, you will see this message displayed 30 minutes after the first infected program is run on any Sunday: "Today is SunDay, why do you work so hard?". Version B does not display the message.
Suriv 1.01	"APRIL 1ST HA HA HA YOU HAVE A VIRUS" displayed on April 1st of any year. On April 1, the machine locks after display of this message. After April 1, 1988, it displays "YOU HAVE A VIRUS!!!"
Yankee Doodle (2885)	Plays "Yankee Doodle" on the system speaker at 5PM.

What Users Can Do to Help

Learn to use your anti-virus software correctly.

You don't want a system in which some anti-virus SWAT team has to roam the halls searching for viruses with their geiger counters. You want users to know what anti-virus to use, when to use it, and especially how to use it. For example, with some programs, if you issue the wrong command line, you don't scan all of your hard disk, like you might think. You scan just the files in the root directory!

You'll improve your odds if your users are motivated and competent. You can either choose complex anti-virus software and provide training, or choose easy-to-learn, easy-to-use anti-virus software, and skip much of the "how-to" training.

Be careful when using low-level tools.

More information is lost by the "data recovery experts" in some companies than by the users themselves. Users put information at risk by deleting it. The experts can completely destroy it by bumbling during the recovery process. Before using any power tools to work with a hard disk, be sure things are backed up, you are using quality tools, and that you have read the manuals and fully understand what you are doing. Potentially lethal power tools include FAT editors, directory editors, directory sorters, unerasers, disk optimizers, format/recover systems, as well as those old DOS villains: FORMAT, CHKDSK, FDISK, RECOVER, and BACKUP/RESTORE.

Be aware of the virus-like effects of Joke Programs

Joke programs are programs that play tricks on users, making the letters tumble from the screen, move about, or disappear as tiny bugs run around the screen. They include such things as BUG, BUGS, DRAIN, DRIPPYand MELT. You can get such programs from the BBS of the National Computer Security Association, and study them, to help you separate their effects from those of real viruses.

Use the Right Tool for the Job

Anti-Virus Tool Comparison

Product Type	Advantages	Disadvantages	Examples
Checksum or CRC Comparison Programs. (Detect any changes to a program.)	Can be used to detect unwanted change in other programs, possibly the result of a virus at work.. There are no viruses able to modify a file without modifying the file's checksum.	Software must be run on an uninfected machine. Once a virus is memory-resident, the checksum program might fail.	Most popular products include checksum options. Dedicated products include Alert, The Antibody Test, BSearch, CHKSUM, CRCDOS, Delouse, The Detecti F-Prot, FICHECK, SSCRC, VCheck, and VirusGuard.

Anti-Virus Tool Comparison

Product Type	Advantages	Disadvantages	Examples
Adding Self-Checking to Programs. (Your program is modified so that when it is run, control is first passed to added code. The added code calculates the current checksum and compares with a stored value. A failed comparison can result in an alert to the user.)	The most efficient approach to checking files is not to check only critical files, or all files, but rather to check files as they are run. The approach is efficient with the user's time, and very effective.	Slows processing a small amount; it enlarges each file a small amount; it may not work on COM files that are nearly 64K in size, since 64K is the largest size supported by the COM format; it cannot work with BIN, SYS, and OVL files; it cannot work on archives (such as *.ZIP or *.ARC or self-extracting EXE files); Some products — particularly those from Borland — modify themselves when a user changes parameters or exits. A vaccine applied to such a product would generate a spurious false alarms with such programs. Since very few programs are currently self-modifying, this does not seem to be anything but an occasional nuisance with the approach.	Skulason's F-XLock (part of F-Prot) may be the only program that let's you do this. Some anti-virus programs detect when they have become infected, display a warning message, and exit. Examples include HTScan, Norton AntiVirus, McAfee's ProScan, and Skulason's F-Prot .
Scanners. (A scanner looks for specific patterns or strings that indicate the presence of a virus, and reports what virus is found, where is is located. Removal capability is provided by some scanners.)	A scanner is critical for providing a precise identification of a virus in your system. Without a precise identification, we cannot know the best course of action for removing it, and we cannot be sure we have removed it completely. A scanner is the only kind of anti-virus program that can spot a virus before it has run.	Scanners are always out of date. Most scanners are usually somewhat incomplete when they are released. No virus researcher has access to every virus on earth. But even if your scanner was absolutely current one month ago, and caught every virus that existed on earth the day it was released, there are now 60 or more viruses that it cannot catch. While your odds of getting a brand new virus are very, very small, you must recognize that scanners will not catch everything.	All popular anti-virus products include a scanner.

Anti-Virus Tool Comparison

Product Type	Advantages	Disadvantages	Examples
Memory Resident Software. (Once run, a memory-resident anti-virus program can monitor all subsequent activities, looking for viruses in files being copied or run, and watching for "suspicious" behavior, such as writes to the Master Boot Record.)	Requires no user intervention once installed. Can focus on programs as they are run, thus more efficiently timing its checking than with manually-activated scanners or checksum programs. Can monitor behavior, not merely search for known strings, thus potentially detecting new viruses.	Many microcomputer configurations are intolerant of TSRs, and crash, lock-up, or misbehave. All TSRs use RAM — something which is in very short supply in today's micros. Any disk write monitor which provides the user with many false alarms will be seen as a nuisance, and likely be de-activated long before a real virus tries a nasty disk write. Any anti-virus TSR can, in theory, be easily detected by viruses, and disabled or removed. A disk write monitor slows the system, as each call is intercepted and processed. A TSR may not be able to prevent a boot sector infection if the infection occurs before the TSR is even loaded, as in an accidental boot from a data disk in A:.	Included with most popular anti-virus products.

Summary

Because a computer is a machine designed to run software, and viruses are software, a computer is a machine that is fundamentally designed to run viruses. And viruses are being designed today to run in your machines.

Even with perfect policy, we must be concerned about perfect implementation. Without the procedures, policy fails. And even with perfect procedures, a lapse in our vigilance opens a window of potential vulnerability. As a result, the micros in our offices are now forever at risk.

The NCSA recommends a combination strategy to detection. We recommend use of an effective scanner before introducing new software to a system. Because a scanner cannot identify viruses that were unknown at the time the scanner was produced, we recommend regular use of a checksum-approach to looking for virus-induced changes in programs. And because users will not always use the above tools at the correct moment (if at all), we also recommend installation of some memory-resident program to monitor for virus activity and/or check for viruses during the copy process, and/or examine files for viruses before they are run.

❖

Recovering From Viruses

Even with inflation, an ounce of prevention is worth a pound of cure.

If you find a virus in your system, and can establish precisely what it is (from the previous chapter on "Detection"), it is now time to remove it from your system.

Virus Removal Tools

There are many programs that will effectively remove certain viruses from your system. Some of these tools are tools you already own, such as SYS, FORMAT, and FDISK. Others are special tools you will purchase. Some virus removal tools are specialists in Master Boot Record or Boot Record repair. Others specialize in removing viruses from files. And a few can do some of both.

File Viruses

For infections of your programs, you can delete all infected files, replacing them with programs from the original distribution diskettes.

Virus Removal

Many anti-virus products offer virus removal capabilities. Some of these functions are built-in to the scanner, giving you the option to remove a virus from a file the moment it is detected in the file. Examples include Norton Anti-Virus, Dr. Solomon's ToolKit, and Skulason's F-Prot. Other products work in pairs. For example, you would use McAfee's Scan to identify the viruses you have, record the secret code that identifies those viruses, then run McAfee's Cleanup with that secret code as a command line parameter, to remove the specified virus(es) from the files.

Removal of some viruses is fairly straightforward for anti-virus products. Removal of others is more difficult. Many anti-virus products can remove some viruses, but cannot remove very many. In choosing a product, ask whether the product can remove all common removable viruses.

Some viruses cannot be removed, no matter how clever the author of the anti-virus software. This is because the virus has overwritten some of your original program, effectively erasing it. Without a knowledge of what was supposed to be where the virus sat down, all an anti-virus product can do is offer to delete the entire program.

Deletion and Overwriting

If you cannot remove a virus from a program in such a way that the program is both clean and willing to run the way it used to, you may want to delete the program. Simply erasing the program does not kill the virus. It removes all reference to the program from the directory and the FATs. This is always enough for file viruses, because a program that cannot be run cannot bring the virus back to life. Over the course of the next day or two, a user will add to the drive or edit files, and the virus will be overwritten.

There always is the outside possibility that some user will unerase an erased file containing a virus. If this happens, whether intentionally or by accident, the virus can be brought back to life. If you want to truly erase a virus, you can:

- ❑ Use an anti-virus product that overwrites a file containing a virus with 0's or 1's or random patterns, to fully exterminate it.
- ❑ Simply optimize or compress your hard disk, causing substantial rearrangement of files and full loss of everything on the outer cylinders that had ever been erased. Norton's SpeedDisk (part of the Norton Utilities) can do this, as can many other optimizers.
- ❑ Use any shareware, public domain, or commercial program to wipe the infected files through overwriting.

Repairing Boot Records

For boot sector infections: Power down the system. Power up and boot from an uninfected, write-protected floppy. Execute the DOS SYS command to attempt an overwrite of the boot sector. This works in many cases. If this does not work, backup all data files and run FDISK.

FORMAT

Format is a DOS utility which prepares the drive for data. On both hard disk and floppy, it scans the surface for bad sectors, creates a boot record, creates two FATs which record the position of any bad sectors found, and creates a root directory. If run with the /V option, it will place a volume label entry in this root. If run with the /S option, it will also write copies of two hidden system DOS files and COMMAND.COM.

Format behaves somewhat differently on floppies than on hard disks. On the floppy, it overwrites every square inch of surface with some character or other, completely obliterating anything that was previously there. On the hard disk, it does not do this.

Formatting can be completely successful at removing all viruses from the floppy, providing you have first shut the machine off then turned it back on, rebooting from an uninfected disk. Unfortunately, FORMAT also removes all data and programs from the floppy, too. So while you can absolutely re-use a previously infected diskette by merely reformatting it, you will lose all your work with this approach.

You must be aware that if a virus is in memory at the time you reformat a floppy, a file virus may infect your copy of FORMAT (unless it is run from a write-protected disk), and a boot virus will add itself to the floppy's boot record after formatting has been completed. So turning the machine off and then booting from a clean disk is important, as is running an uninfected copy of FORMAT.

Some people think that formatting a hard disk is just as useful for removing boot viruses as is formatting a floppy. This is not generally true. If the virus has infected the Master Boot Record (an area created or modified by FDISK but not FORMAT), you may either not remove the virus, or you might lose your ability to boot from the drive if the virus has displaced some of the Master Boot Record. FDISK, described below, is the another household tool you can use to remove a virus.

SYS

The SYS program does some of the functions of FORMAT, with less potential damage to your data. To remove a virus from a floppy's boot record, simply boot from an uninfected disk, run a scanner to make sure there is no virus in the disk you booted from, then run the SYS program, as in SYS A:. The boot record of the floppy will be replaced, along with the hidden system files. If you get the message "No room for system on destination", it is likely because that diskette was not previously bootable. However, the SYS command will have already overwritten the boot record of the floppy, and the virus will likely be gone.

Unfortunately, SYS does not touch the Master Boot Record of a hard disk. Only FDISK does this. So if you have a virus that hides in the Master Boot Record, you cannot overwrite it on your hard disk with SYS.

Repairing Master Boot Records

FDISK

FDISK is a program you use when to create a "master boot record", including a "partition table". This information, stored in one 512 byte sector on the outside cylinder, first sector, first side of your hard disk, tells the hard disk controller basic critical information on your drive. The information includes instructions on where the heads of the hard disk should move to read the boot record of C:, D:, E:, etc., and from which of these partitions it should continue to read during the boot process.

FDISK can absolutely destroy any virus on a hard disk that occupies the master boot sector, providing that the virus is not in memory when FDISK is run. When you have finished running FDISK, you will need to run FORMAT to finish writing the booting instructions.

Now for the bad news: FDISK guarantees that you will permanently lose all the files on your hard disk (unless you are a real data recovery expert), because most versions overwrite some critical information.

Norton Disk Doctor

If you have a copy of Norton Disk Doctor (part of the Norton Utilities), you can follow this simple procedure: Boot from an uninfected floppy. Run Norton Disk Doctor. Ask it to repair the Master Boot Record. It will gently place a new, intelligent file on top of your old virus-infected Master Boot Record. Your problem may be over. Simply scan again for viruses, to see if anything turns up.

Strategies

Having procedures in place to detect viral infection is very important. By itself, however, it is of little use. The individual who makes the first detection must have a procedure to follow to verify the problem and to make sure that appropriate action occurs. If the information supplied in the detection phase is allowed to fall between the cracks, even for a relatively short time, the benefit of detection can easily be lost.

Catch it Early!

Computer viruses spread exponentially — one becomes two, two become four, four become eight... become hundreds. Often a virus is detected when only one machine is infected. But sometimes the infection is not caught for several days, and there are dozens or hundreds of infected machines. Many users have lost months worth of work because they did not prevent the virus from getting in, did not detect it early enough, did not know enough about recovery, and did not have good backups. You can't play cards in the computer world without something in your hand. If you won't prevent, won't detect early, did not know how to recover, and did not have backups, then you lose.

Be a Sociologist

When planning your recovery strategy, consider the sociometry of your organization. The person to spot the virus first is possibly the one who has been infected the longest. Those they share disks with are likely also infected, if disk sharing has happened in the last few days. Users in your organization with home computers are likely sources of infection. Users who try the most software are likely to be the most vulnerable. And users with the greatest access to files and machines — the help desk, network manager, and other support staff — are the most common carriers of viruses once the virus has gotten in. Look at each of these players in your organization, and make sure that they are not infected. After you have recovered, you can spend some more time working with these key players to minimize their contribution to your next problem, and maximize their contribution to your virus solution.

Devise an Action Plan

It might be a good idea for you to have some rough action plan ready in the event that your organization has an infection. Some of the details to be covered by such a plan are described below. If you are a little late for a plan, and have an infection right now, you can still consider these steps as you work through your jam.

- ❑ Create a crisis team to take on the assignment. Relieve them of their other duties until you are sure that there is not one copy of this virus still running loose in the organization.

- ❑ Take notes on where the virus has surfaced, and try to draw some intelligent conclusions of how it might have gotten there, and where it might have gone from there. If you can track down the most likely path, rather than simply sweeping from one end of the building to the other, you are likely to be able to stop it efficiently.

- ❑ Isolate all systems from each other during the disinfection process, insofar as you can. Each system with a virus should be closed to the outside world, and each system that is virus-free should temporarily be closed to the outside world, until that world can be trusted. You may need to bring the network down for a few days.

- ❑ Brief all users and staff about the situation, and give them useful, concrete tasks to perform to help protect themselves, search for additional copies of the virus, and remove any copies found. Don't try to keep this a secret from users. There is nothing embarrassing about a virus. But there should be embarrassment in not asking others for their help.

- ❑ Arm yourself with the very best anti-virus software. You want to have sure-fire rapid and accurate detection and an easy-to-use virus removal procedure. If this means running out to the store and doing some hasty shopping for an anti-virus product, do so.
- ❑ Arm yourself with thorough information about the virus you have discovered. The more you know about your enemy, the better you will be able to deal with it. Call your anti-virus vendor's hotline, call your computer dealer, call NCSA, call anyone you think will know about this virus. Read what you can about this virus in your virus books.

Don't run Disinfectors Needlessly

If you run FORMAT, FDISK, SYS, MD, or just about any serious virus removal tool, and you try to remove a virus that is not there, you can certainly make a mess of things. Don't disinfect until you have clearly established that you have an infection.

Remove Every Single Copy of the Virus

The plot in recovery is to remove every single copy of the virus from the premises, except perhaps one or two trophies for your wall. This removal includes every single copy on every single hard disk, and every single floppy.

Removal will either require physical extraction of the virus code from every infected program, restoring them to a form that is functionally identical to that prior to infection. It isn't necessary that these disinfected programs be identical, byte-for-byte, only that they function identically. It isn't even necessary to physically remove every byte of the virus from the file, as long as you remove the critical byte(s) that the virus requires to execute.

If your virus cannot be removed from files, then the files containing the virus must be deleted and replaced from backups. Remember that the virus might be on your backups, and so you must scan for the virus after you think you have successfully restored everything from them.

Use Backups Wisely

If you can learn some key details about your virus, you can safely restore files from infected backups. For instance, if you know that your virus only infects COM files, you can restore all other file types from infected backups. You might want to restore these other file types if the virus has destroyed your hard disk copy of them (through FAT manipulation, for instance.)

If you learn that your virus is a boot sector virus, then none of the files in your backups will be contaminated. Once the boot sector virus is removed from memory, and you have booted from an uninfected disk, you will be able to restore all files from backups infected with a boot sector virus. Just remember to not accidentally boot with one of these floppies!

Watch for Re-Infection

If you have just removed a virus from a system, and cleaned every single diskette in site, you should not yet breath easily. You are very likely to get another infection in the next few days, from some unnoticed copy of the virus or from the same outside source from which you got the first infection.

Don't Panic

You must remember that the majority of virus alarms turn out to be false alarms, and are the result of some user, software, or hardware error. Proceed with your detection assuming that while it might be a virus, it is just as likely to be something else, so note all symptoms carefully, and see what the combination points to.

If it turns out to be a virus, it is still very important to not panic. You will need a cool head to minimize further damage and to remove it as cleanly and efficiently as possible. Don't just get out your copy of FORMAT and begin your counterattack!

Terminate Network Connections

Until the virus is removed from all systems, your network provides a terrific means of spreading it. If you clean 99 of 100 workstations, and the remaining workstation logs into the network, you may have 100 more workstations to clean. Unless you enjoy fighting viruses, are paid by the hour, and have nothing else to do with your time, the network must be shut down until the entire organization is clean.

Isolate all Infected and Potentially Infected Micros

Any system that is infected must be isolated from all others. This means no diskette transfer into or out of that office, no file transfer in or out. Until you can be sure which other systems are absolutely clean, they should be completely isolated as well. Without such isolation, users can spread the virus faster than you can crush it.

If You Can't Immediately Identify the Virus, Record Clues

Record your last activities. Write down as many details as possible, including the names of all programs you ran since booting the system, any odd behaviors of the system, etc. Note any times when the system seemed unusually slow. Try to recollect whether your recent uses of the system were abnormal in any way. Capture the current screen, if it could possibly be of use in detective work, using a Shift-PrintScrn.

Study the nature and extent of the damage. Even if you plan to FDISK and FORMAT the hard disk and restore from backups, you should be aware of exactly what has happened to your system: which files, in which directories, were affected? Has the damage stopped, or does it continue with continued use?

Notify Other Users

Once you are certain that you have an infection (or have been hit by a Trojan), notify other appropriate users, including anyone you think might have been the source of the virus or Trojan), and anyone who might have received a copy from you.

How Serious is the Virus Threat?

Viruses were once a very minor security problem. Your probability of infection was too low to justify the devotion of too many resources to the problem. But today the probability of infection has risen to an astonishing level. Given the certainty that your organization will experience an infection in the next six months, and given the likelihood that systems lacking virus-free backups will lose important programs and perhaps data files, some intelligent action for prevention and detection is now required.

While the best defense against viruses is prevention, the cost of complete prevention is probably excessive. Consequently, your best use of your anti-virus budget would be to save some funds for recovery.

Your exact probability of being infected is a function of several factors: the number of viruses appearing at your door, and the measures you have in place for preventing their entry through that door. The number "at the door" is determined by the frequency with which new software is brought into your organization (by disk or modem) and the increasing probability that this software is virus-infected. The measures in place are established by policy and procedure.

The magnitude of the threat is difficult to estimate, given the extent of under-reporting. Victims of crimes, like the rest of us, wish to believe in a "just world", in which good things happen to good people, bad things happen to bad people. Because a virus infection is definitely a bad thing, victims often some shame and guilt. No one feels pride when they experience an infection, no one seems to want to talk about it. But it is increasingly clear that computer viruses are posing a new, grave threat to safe computing. You can do something to help make computing safe again. We're all counting on you.

❖

For Further Information

We provide here a few key references for those who need to learn more about computer viruses.

Books and Reports

All publications listed here are available from National Computer Security Association, Suite 309, 4401-A Connecticut Av NW, Washington DC 20008.

Computer Viruses

The definitive reference and survival guide for combating computer viruses. Intended for every organization with microcomputers, this encyclopedia provides the latest information on viruses and describes how to prevent, detect, remove, and recover from them. It includes suggestions for virus policy, information on hundreds of worms, Trojan horses, and hacks, and numerous graphs and tables.

Now in its seventh edition, the book describes 893 viruses that infect the PC, as well as viruses affecting the Mac, Atari, and Amiga. For each virus, we have included information such as synonyms, behavior, scan codes, date and place of origin, programs which will detect it, scan codes, an NCSA sample number (which references it to our collection of over 5,000 samples), and how to deal with it. Other sections of this 379-page, 8 1/2" x 11" book describe how viruses operate and how to prevent them. There are nearly 1,100 index entries. Various tables show synonyms, virus evolution, and scan codes for hundreds of viruses, and numerous graphs show trends. Available from the NCSA for $44 for NCSA members, $75 for non-members.

Defend Your Data: A Guide to Data Recovery

NCSA's *Defend Your Data!* is now in its fifth edition. This book provides a thorough explanation of how disk drives and DOS manage information, and step-by-step instructions for coping with dozens of data disasters. Over 100 experiments are described in detail, to allow you to practice your skills before you need to use them. The information in this book is useful for those charged with recovering lost data, with repairing computers, and with recovering from viruses. Like other NCSA books, this one contains illustrations, a directory of products, and a complete index, and is accompanied by a disk crammed with relevant programs. The index includes every DOS error message you are likely to see. Available from NCSA for $44 for members, $75 to non-members.

Product Comparison: Programs to Detect Changes in Programs

Checksum or CRC comparison programs are used to detect unwanted change in other programs, possibly the result of a virus at work. Change detection is not just good for detecting a virus at work, but it can be helpful in recovery from hardware problems. Some organizations use such software to help ensure that only authorized software is in use.

In this research report, we make an effort to concisely describe the merits of this class of products, and then to help you in selecting a product from their ranks. Our ratings of 19 different products may surprise you. We scored some of the most popular products lowest! Available for $44 to NCSA members, $75 to nonmembers.

Virus Scanners: An Evaluation

The most popular means of detecting viruses is through the use of a scanner, which examines your programs to see if they contain a specific string of code that is unique to a given virus or family of viruses. Many viruses are now self-encrypting, and do not create such easily recognized strings when they have infected a file. Stealth viruses, which remove themselves from files as they are scanned, pose a further problem for scanners, which must check for patterns in memory as well.

Virus Scanners: An Evaluation is a best-selling research report from NCSA. The report compares nearly two dozen popular scanners in tests of their abilities to detect hundreds of viruses in over 900 infected files. Comparison tables show:

- □ What virus(es), if any, each identified in each of the 900+ infected files and boot sectors;
- □ Which common viruses each failed to detect, if any;
- □ Ability to detect a stealth virus, once it was memory-resident;
- □ The program's reporting functions, including adequacy of messages from the scanner and the ability to send output to a file;
- □ The scanner's speed, ability to run in batch mode, and efficiency in scanning removable media;
- □ The scanner's ability to detect that it had become infected, and what it does when (if) it finds this has happened;
- □ The ability to add signatures to a file that will be read by this scanner, so you can scan for virus signatures that did not exist when you acquired the scanner;
- □ The ability of a user to scan boot record, partition table during scans, the ability to list files to be scanned, and the ability to control the types of files to be scanned;

- ❑ Your options in removing viruses or deleting infected files, when an infection is discovered.

You may be surprised at the findings in this report. The report is available from NCSA now, at a cost of $44 to members and $75 to the general public.

Virus Removal Tools: An Evaluation

This new NCSA report evaluates the virus removal abilities of the dozen best programs available. Aspects evaluated include:

- ❑ What happens if the removal tool becomes infected with a virus?
- ❑ Could the product detect four common and three less common boot virus?
- ❑ Could the product remove these viruses from the infected boot disk?
- ❑ Could the product detect 18 common and 18 exotic file viruses?
- ❑ Could the product remove these viruses from the file on the infected disk?
- ❑ Could the product delete the infected files?
- ❑ Did the product remove all copies of a virus that infected a file multiple times? Would the file then run correctly?

Overall product performance ranged from 54 to 129 out of a possible 142 points. Available for $44 to NCSA members, $75 to non-members.

Materials for Virus Trainers

Today every computer trainer and help desk staffer is finding a need for virus training. NCSA's materials for trainers will help you. This package includes just about everything but a microphone:

- ❑ *Reference Material.* You get the latest copy of Computer Viruses and copies of each of our anti-virus product evaluations.

- ❑ *Student Handouts.* Your kit will include as many copies of this booklet as you might need to hand out to students. It also includes a disk-based general tutorial on viruses and a disk-based tutorial on boot viruses.
- ❑ *Training Aids.* Your kit includes a set of figures and tables you can photocopy onto acetate and use in your training. And it includes a menu-driven program that runs simulations of a number of interesting viruses.

Kit pricing will vary with the number of copies of this booklet you require. Call for more information.

Organizations

National Computer Security Association (NCSA)

The National Computer Security Association (NCSA) is an organization dedicated to improving our understanding of computer security and the threats to it such as computer viruses, and to devising practical ways to improve it. To achieve these goals, NCSA conducts research on computer viruses and other security problems, evaluates anti-virus and other computer security products, and presents its findings to the public through seminars, books, research reports, and telephone support.

The NCSA is located at Suite 309, 4401-A Connecticut Avenue NW, Washington DC 20008. 717-258-1816 (both voice and fax).

National Computer Security Center (NCSC)

This is the government agency charged with helping military and defense agencies in computer security. They may be able to provide such organizations with some virus consulting. You can call them at 301-859-4371.

National Institute of Standards and Technology (NIST)

The Federal Government's organization to help with computer security for non-defense agencies and the private sector. Their public information office number is 301-975-2762. You can write them at NIST Public Affairs Office, A903 Administration Building, Gaithersburg, MD 20899. Their fax number is 301-926-1630.

Bulletin Boards

Each of the bulletin boards listed below promises virus-free software for download, including the latest information on computer viruses. They are also good sources for other security and data recovery software and information.

- ❑ Washington, DC: 202-364-1304.
- ❑ Denver, Colorado: 303-962-9536
- ❑ Hayward, Califomia: 415-786-0471
- ❑ Hudson, Ohio: 216-656-1046
- ❑ Exeter, England: +44 392-433566; 221730

❖

ISBN 0-941375-36-6

9 780941 375368