A seminar

On

ANTIVIRUS

Submitted by

T. ARUNA KUMARI
MCA III SEM
ROLL NO: 12

Submitted to

Computer Science Department
Mahatma Gandhi College (P.G courses)
Edulapalem, Guntur.

CONTENTS

1. INTRODUCTION:

A virus reproduces, usually without your permission or knowledge. In general terms they have an infection phase where they reproduce widely and an attack phase where they do whatever damage they are programmed to do (if any). There are a large number of virus types.

Viruses are a cause of much confusion and a target of considerable misinformation even from some virus "experts." Let's define what we mean by virus:

A virus is a program that reproduces its own code by attaching itself to other executable files in such a way that the virus code is executed when the infected executable file is executed.

You could probably also say that the virus must do this without the permission or knowledge of the user, but that's not a vital distinction for purposes of our discussion here. **We are using a broad definition of "executable file" and "attach" here.**

An obvious example of an executable file would be a program (COM or EXE file) or an overlay or library file used by an EXE file. Less obvious, but just as critical, would be the macro portion of what you might generally consider to be a data file (e.g., a Microsoft Word document). It's important to also realize that the system sectors on either a hard or floppy disk contain executable code that can be infected--even those on a data disk. More recently, scripts written for internet web sites and/or included in E-mail can also be executed and infected.

To attach might mean physically adding to the end of a file, inserting into the middle of a file, or simply placing a pointer to a different location on the disk somewhere where the virus can find it.

Most viruses do their "job" by placing self-replicating code in other programs, so that when those other programs are executed, even more programs are "infected" with the self-replicating code. This self-replicating code, when triggered by some event, may do a potentially harmful act to your computer.

Another way of looking at viruses is to consider them to be programs written to create copies of themselves. These programs attach these copies onto host programs (infecting these programs). When one of these hosts is executed, the virus code (which was attached to the host) executes, and links copies of itself to even more hosts.

Similar to viruses, you can also find malicious code in Trojan Horses, worms, and logic bombs. Often the characteristics of both a virus and a worm can be found in the same beast; confusing the issue even further.

Before looking at specific virus types you might also want to consider the following general discussions:

- **Virus Behavior.** Infect, then attack; common behavior of most viruses.
- **Number of Viruses.** Lots and lots.
- **Virus Names.** It's not easy nor standardized.
- **How Serious Are Viruses?** Worms spreading due to user inattention are a serious threat.
- **What About Good Viruses?** The general consensus is that there are none.
- **Hardware Threats.** Viruses are not the only things that can cause damage. Consider some hardware problems.
- **Software Threats.** Viruses are not the only things that can cause damage. Consider some software problems.

Summary

- A virus is a program that reproduces its own code.
- Generally, the first thing a virus does is to reproduce (i.e., infect).
    - Viruses balance infection versus detection possibility.
    - Some viruses use a variety of techniques to hide themselves.
- On some defined trigger, some viruses will then activate.
    - Viruses need time to establish a beachhead, so even if they activate they often will wait before doing so.
    - Not all viruses activate, but all viruses steal system resources and often have bugs that might do destructive things.
- The categories of viruses are many and diverse. There have been many made and if you get one it should be taken seriously. Don't be fooled by claims of a good virus; there is no reason at the moment to create one.

Virus Behavior

Viruses come in a great many different forms, but they all potentially have two phases to their execution, the infection phase and the attack phase:

Infection Phase

> **Virus writers have to balance how and when their viruses infect against the possibility of being detected.**

Therefore, the spread of an infection may not be immediate.

When the virus executes it has the potential to infect other programs. What's often not clearly understood is precisely when it will infect the other programs. Some viruses infect other programs each time they are executed; other viruses infect only upon a certain trigger. This trigger could be anything; a day or time, an external event on your PC, a counter within the virus, etc. Virus writers want their programs to spread as far as possible before anyone notices them.

It is a serious mistake to execute a program a few times - find nothing infected and presume there are no viruses in the program. You can never be sure the virus simply hasn't yet triggered its infection phase!

Many viruses go resident in the memory of your PC in the same or similar way as terminate and stay resident (TSR) programs. (For those not old enough to remember TSRs, they were programs that executed under DOS but stayed in memory instead of ending.) This means the virus can wait for some external event before it infects additional programs. The virus may silently lurk in memory waiting for you to access a diskette, copy a file, or execute a program, before it infects anything. This makes viruses more difficult to analyze since it's hard to guess what trigger condition they use for their infection.

On older systems, standard (640K) memory is not the only memory vulnerable to viruses. It is possible to construct a virus which will locate itself in upper memory (the space between 640K and 1M) or in the High Memory Area (the small space between 1024K and 1088K). And, under Windows, a virus can effectively reside in **any** part of memory.

Resident viruses frequently take over portions of the system software on the PC to hide their existence. This technique is called stealth. Polymorphic techniques also help viruses to infect yet avoid detection.

Note that worms often take the opposite approach and spread as fast as possible. While this makes their detection virtually certain, it also has the effect of bringing down networks and denying access; one of the goals of many worms.

Attack Phase

Viruses need time to infect. Not all viruses attack, but all use system resources and often have bugs.

Many viruses do unpleasant things such as deleting files or changing random data on your disk, simulating typos or merely slowing your PC down; some viruses do less harmful things such as playing music or creating messages or animation on your screen. Just as the infection phase can be triggered by some event, the attack phase also has its own trigger.

Does this mean a virus without an attack phase is benign? **No.** Most viruses have bugs in them and these bugs often cause unintended negative side effects. In addition, even if the virus is perfect, it still steals system resources. (Also, see the "good" virus discussion.)

Viruses often delay revealing their presence by launching their attack only after they have had ample opportunity to spread. This means the attack could be delayed for days, weeks, months, or even years after the initial infection.

The attack phase is optional, many viruses simply reproduce and have no trigger for an attack phase. Does this mean that these are "good" viruses? **No!** Anything that writes itself to your disk without your permission is stealing storage and CPU cycles. (Also see the "good" virus discussion.) This is made worse since viruses that "just infect," with no attack phase, often damage the programs or disks they infect. This is not an intentional act of the virus, but simply a result of the fact that many viruses contain extremely poor quality code.

An an example, one of the most common past viruses, Stoned, is not intentionally harmful. Unfortunately, the author did not anticipate the use of anything other than 360K floppy disks. The original virus tried to hide its own code in an area of 1.2MB diskettes that resulted in corruption of the entire diskette (this bug was fixed in later versions of the virus).

Number of Viruses

> **There are currently over 50,000 computer viruses and that number is growing rapidly. Fortunately, only a small percentage of these are circulating widely.**

There are more MS-DOS/Windows viruses than all other types of viruses combined (by a large margin). Estimates of exactly how many there are vary widely and the number is constantly growing.

In 1990, estimates ranged from 200 to 500; then in 1991 estimates ranged from 600 to 1,000 different viruses. In late 1992, estimates were ranging from 1,000 to 2,300 viruses. In mid-1994, the numbers vary from 4,500 to over 7,500 viruses. In 1996 the number climbed over 10,000. 1998 saw 20,000 and 2000 topped 50,000. It's easy to say there are more now.

The confusion exists partly because it's difficult to agree on how to count viruses. New viruses frequently arise from someone taking an existing virus that does something like put a message out on your screen saying: "Your PC is now stoned" and changing it to say something like "Donald Duck is a lie!". Is this a new virus? Most experts say yes. But, this is a trivial change that can be done in less than two minutes resulting in yet another "new" virus.

Another problem comes from viruses that try to conceal themselves from scanners by mutating. In other words, every time the virus infects another file, it will try to use a different version of itself. These viruses are known as polymorphic viruses.

One example, the Whale (a huge clumsy 10,000 byte virus), creates 33 different versions of itself when it infects files. At least one person counts this as 33 different viruses on their list. Many of the large number of viruses known to exist have not been detected in the wild but probably exist only in someone's virus collection.

David M. Chess of IBM's High Integrity Computing Laboratory reported in the November 1991 Virus Bulletin that "about 30 different viruses and variants account for nearly all of the actual infections that we see in day-to-day operation." Now, about 180 different viruses (and some of these are members of a single family) account for all the viruses that actually spread in the wild. To keep track visit the Wildlist, a list which reports virus sightings.

How can there be so few viruses active when some experts report such high numbers? This is probably because most viruses are poorly written and cannot spread at all or cannot spread without betraying their presence. Although the actual number of viruses will probably continue to be hotly debated, what is clear is that the total number of viruses is increasing, although the active viruses not quite as rapidly as the numbers might suggest.

Summary

- By number, there are over 50,000 known computer viruses.
- Only a small percentage of this total number account for those viruses found in the wild, however. Most exist only in collections.

Virus Names

> **A virus' name is generally assigned by the first researcher to encounter the beast. The problem is that multiple researchers may encounter a new virus in parallel which often results in multiple names.**

What's in a name? When it comes to viruses it's a matter of identification to the general public. An anti-virus program does not really need the name of a virus as it identifies it by its characteristics. But, while giving a virus a name helps the public at large it also serves to confuse them since the names given to a particular beast can differ from anti-virus maker to anti-virus maker.

How? Why? Much as they would like to, the virus writers do not get to name their beasts. Some have tried by putting obvious text into the virus but most of the anti-virus companies tend to ignore such text (mostly to spite the virus writers). And, any virus writer that insists on a particular name has to identify themselves in the process--something they usually don't want to do. So, the anti-virus companies control the virus naming process. But, that leads to the naming problem.

Viruses come into various anti-virus companies around the world at various times and by various means. Each company analyzes the virus and assigns a name to it for tracking purposes. While there is cooperation between companies when new viruses are identified, that cooperation often takes a back seat to getting a product update out the door so the anti-virus company's customers are protected. This delay allows alternate names to enter the market. Over time these are often standardized or, at least, cross-referenced in listings; but that does not help when the beast makes its first appearance.

This problem/confusion will continue. One practical and well documented example of how it affects a real-world virus listing can be seen at the WildList site on the page...

> http://www.wildlist.org/naming.htm

One attempt at bringing some order to the naming problem is Ian Whalley's VGrep. VGrep attempts to collect all of the various virus names and then correlates them into a single searchable list. While useful, there is, again, the lag time necessary to collect and correlate the data.

So, get used to viruses having different names. As Shakespeare said...

> *What's in a name? That which we call a rose*
> *By any other name would smell as sweet...*

Summary

- Virus naming is a function of the anti-virus companies. This results in different names for new viruses.
- Different names can cause confusion for the public but not anti-virus software which looks at the virus, not its "name."
- There are different sites that attempt to correlate the various virus names for you.

2. Virus Types

Viruses come in many types; written using many different infection strategies.

Viruses come in a variety of types. Breaking them into categories is not easy as many viruses have multiple characteristics and so would fall into multiple categories. We're going to describe two different types of category systems: what they infect and how they infect. Because they are so common, we're also going to include a category specific to worms.

What They Infect

Viruses can infect a number of different portions of the computer's operating and file system. These include:

- System Sectors
- Files
- Macros
- Companion Files
- Disk Clusters
- Batch Files
- Source Code
- Worms using Visual Basic

How They Infect

Viruses are sometimes also categorized by how they infect. These categorizations often overlap the categories above and may even be included in the description (e.g., polymorphic file virus). These categories include:

- Polymorphic Viruses
- Stealth Viruses
- Fast and Slow Infectors
- Sparse Infectors
- Armored Viruses
- Multipartite Viruses
- Cavity (Spacefiller) Viruses
- Tunneling Viruses
- Camouflage Viruses
- NTFS ADS Viruses

And, in a special category, one might include:

- Virus Droppers Programs that place viruses onto your system but themselves may not be viruses (a special form of Trojan).

If you know, click on the virus topic you are interested in or read about each in sequence…

System Sector Viruses

> **System sectors (Master Boot Record and DOS Boot Record) are often targets for viruses. These boot viruses use all of the common viral techniques to infect and hide themselves. While mostly obtained from an infected disk left in the drive when the computer starts, they can also be "dropped" by some file infectors.**

System sectors are special areas on your disk containing programs that are executed when you boot (start) your PC. Every disk (even if it only contains data) has a system sector of some sort. Sectors are simply small areas on your disk that your hardware reads in single chunks. System sectors are invisible to normal programs but are vital for correct operation of your PC. They are a common target for viruses. There are two types of system sectors found on DOS/Windows PCs:

- **DOS Boot Sectors** (DBS)
- **Partition Sectors** (often called Master Boot Record or MBR)

System sector viruses modify the program in either the DOS boot sector or the Master Boot Record. Since there isn't much room in the system sector (only 512 bytes), these viruses usually have to hide their code somewhere else on the disk. These viruses sometimes cause problems when this spot already contains data that is then overwritten.

Some viruses, such as the Pakistani Brain virus, mark the spot where they hide their code as bad. This is one reason to be suspicious if any utility suddenly reports additional bad sectors on your disk and you don't know why (don't panic, bad sectors occur frequently for a wide variety of reasons). These viruses usually go resident in memory on your PC, infect the hard disk, and infect any floppy disk that you access. Simply looking at the directory of a floppy disk may cause it to be infected if one of these viruses is active in memory.

On Macintosh systems, some viruses will even infect a diskette immediately upon inserting a diskette into the floppy drive. (PCs generally do not access a disk automatically as the Macintosh does.)

Since viruses are active in memory (resident), they can hide their presence. If Brain is active on your PC, and you use a sector editor to look at the boot sector of an infected diskette, the virus will intercept the attempt to read the infected boot sector and instead return a saved image of the original boot sector. You will see the normal boot sector instead of the infected version. Viruses that do this are known as stealth viruses.

In addition to infecting diskettes, some system sector viruses also spread by infecting files. Viruses of this type are called multipartite (multiple part) viruses. Since they can infect both files and system sectors they have more avenues to spread. (**Note:** Some file viruses also infect system sectors to complete the circle.)

Summary

- System sectors (MBR and DBS) are often targets for viruses.
- Even data disks can be infected by these viruses.
- System sector viruses spread easily via floppy disk infections and, in some cases, by cross infecting files which then drop system sector viruses when run on clean computers.

Macro Viruses

> **Pure data files cannot propagate viruses, but with extensive macro languages in some programs the line between a "data" file and executable file can easily become blurred to the average user. While text E-mail messages can't contain viruses they may have attachments that do and some E-mail programs will automatically load and run these. Don't let them. Finally, be careful of programs that use other programs for reading E-mail.**

As indicated throughout this tutorial, in order for a virus to do anything, first a program of some type must execute. A virus, no matter what type, is still a program and it must load into memory and run in order to do anything. Simply reading it into memory is not sufficient. Pure data files are not viruses simply because, by their nature, they do not execute.

The problem, however, is that many modern programs contain some form of macro language; in some cases a very powerful macro language with commands that include opening, manipulating, and closing files. More and more, these programs allow a user to extend their capabilities by writing powerful macros and then attaching these to data files produced by that program. In many cases, in order to make things easy for users, the macros are set up to run automatically whenever the data file is loaded. It's in cases like this where the line between a data file and program starts to blur.

Note: There are many triggers (other than loading the document) that viral code can exploit. And, once running, various elements of the program's macro language can be exploited so that all future data files produced by that program version could contain the viral macro code.

Most scanners have default settings that check the most common executable files and data files from programs that have a macro language. So, when using those programs it's a good idea to not change the default extension so scanners can find the files they need to. Also, scanners can be set to check every file instead of just files that normally execute; but most do not do this by default--that would make the scanning process too long for most people.

In order to know when to turn full scanning on you need to know something about the software you use. In particular, you need to make yourself aware of any software that uses the sort of "automatic macro" feature described here. Never use a piece of software until you've explored its manual for some time just to see its full capabilities. If these include some sort of "programming" (macro) language, be aware there is an opportunity for problems. Common programs with macro capability that can be exploited by virus writers are Microsoft Word®, Excel® and other Office programs. Windows Help files can also contain macro code (but are rarely exploited because of the difficulty in doing so). And, the latest macro code to be exploited exists in the full version of the Acrobat program which reads and writes PDF files (the free reader is **not** affected; only the full version).

A second vulnerability exists on the Internet. Some E-mail programs and Internet browsers allow you to click on a data file or program that might be attached to a message or displayed on a web page and have that file or program load and/or run automatically. **You should not allow this to happen.** Always save the file or program to disk and then check it with anti-virus software before loading or executing it (or have an anti-virus program that "attaches" to your programs such that it checks files before the program loads them or checks E-mail as it comes in).

And, even more insidious are newer E-mail programs that allow one to use programs like Microsoft Word to read and write messages. You may not even know you are using Word. But, since the E-mail program does use Word, macros can be encoded into the message and be made to run on your system when you open the message to read it. It is very important that you know the characteristics of programs you use! Only then will you be able to determine if you are at risk.

Summary

- With macro languages the line between pure data files and executable files is blurring.
- An infected file might be attached to an E-mail. Don't automatically run attached files.
- Be careful of E-mail programs that use other programs with macros to display or create incoming mail.

Batch File Viruses

Batch files can be used to transmit binary executable code and either be or drop viruses.

While not often found, it is possible to write a batch file that contains a virus. In most cases the batch file is used to drop a memory or disk virus which then takes over when the computer is next started. These don't always work, but it is interesting to briefly go over the design so you can possibly recognize this type of virus if you happen to see one.

One batch file virus takes the following form (it's possible when this page displays you will receive a virus warning if you are running anti-virus software; don't worry, it's just triggering off the partial text below which has the virus code removed):

```
@ECHO OFF
:[ a label of specific form I won't mention ]
COPY %0.BAT C:\Q.COM>NUL
C:\Q
[ binary data ]
```

The first line causes batch file commands to not display on the screen so you won't see what's going on. The second line is a label as far as the batch file is concerned. In reality, this label is what makes the whole thing work so, of course, we're not going to show any examples. The third line copies the batch file itself to an executable file named Q.COM in the root directory of the C: drive. The output of the COPY command is directed to the NUL device so you see nothing on the screen that indicates this copy took place. Finally, the fourth line executes the newly created Q.COM file.

On the surface you would think that trying to rename a .BAT file to .COM and execute it would result in nothing but errors. Normally, that is the case but the label changes all that. The text up to the label converts to instructions the CPU can execute, but they do nothing. When the label is "executed" this changes. The CPU interprets the label as instructions that cause the CPU to look ahead to the binary instructions in the batch file. These binary instructions are the real virus (or virus dropper).

There are several batch file viruses, but each works in a manner similar to that described above. The labels and batch file instructions may differ; but the method of operation is similar.

Use the characteristics of the virus described above to look for batch file viruses. If there are obscure labels (lines starting with a colon) at the start of a batch file, use caution. Most batch file labels are fairly straightforward words or names. Secondly, if you see a batch file that is several thousand bytes long yet when you use the DOS command TYPE to display it to the screen you only see a few lines, that is another tip-off. Most batch file viruses insert an end-of-file mark (Control-Z) between the batch file portion and the binary instruction portion.

Batch file viruses are not common; but be aware they do exist and have been seen in the wild. Indeed, a new worm version surfaced in early June 2002: Cup. This beast is complicated and arrives attached to an E-mail. If executed, Cup creates, executes, and sometimes deletes the files WORLDCUP_SCORE.VBS, EYEBALL.REG, JAPAN.VBS, ENGLAND.VBS, IRELAND.VBS, URAGUAY.VBS and ARGENTINA.BAT. The first file mass mails a file called WORLDCUP.BAT to your Outlook address book. The .REG file assures the worm is run at system start by changing the Windows registry. The worm has other payloads in the various .VBS files. So, you see that batch file viruses/worms can be fairly complicated.

Summary

- Batch files can be used to transmit binary executable code and either be or drop viruses.
- To detect these viruses look for two signs:
  - An odd label at the start of the batch file
  - A batch file that is too large for the text in it.

Navigation
Virus Intro
Virus Types
- Companion
Tutorial Map
Anti-Virus Software

How viruses Infect

Viruses are sometimes also categorized by how they infect. These categorizations often overlap the categories above and may even be included in the description (e.g., polymorphic file virus). These categories include:

- **Polymorphic Viruses**
  Viruses that change their characteristics as they infect.
- **Stealth Viruses**
  Viruses that try to actively hide themselves from anti-virus or system software.
- **Fast and Slow Infectors**
  Viruses that infect in a particular way to try to avoid specific anti-virus software.
- **Sparse Infectors**
  Viruses that don't infect very often.
- **Armored Viruses**
  Viruses that are programmed to make disassembly difficult.
- **Multipartite Viruses**

Viruses that may fall into more than one of the top classes.

- **Cavity (Spacefiller) Viruses**
  Viruses that attempt to maintain a constant file size when infecting.
- **Tunneling Viruses**
  Viruses that try to "tunnel" under anti-virus software while infecting.
- **Camouflage Viruses**
  Viruses that attempted to appear as a benign program to scanners.
- **NTFS ADS Viruses**
  Viruses that ride on the alternate data streams in the NT File System.

Fast and Slow Infectors

> **A fast infector infects any file accessed, not just run. A slow infector only infects files as they are being created or modified.**

The term *fast* or *slow* when dealing with viruses pertains to how often and under what circumstances they spread the infection.

Typically, a virus will load itself into memory when an infected program is run. It sits there and waits for other programs to be run and infects them at that time.

A **fast infector** infects programs not just when they are run, but also when they are simply accessed. The purpose of this type of infection is to ride on the back of anti-virus software to infect files as they are being checked. By its nature, anti-virus software (a scanner, in particular) opens each file on a disk being checked in order to determine if a virus is present. A fast infector that has not been found in memory before the scanning starts will spread itself quickly throughout the disk.

A **slow infector** does just the opposite. A slow infector will only infect files when they are created or modified. Its purpose is to attempt to defeat integrity checking software by piggybacking on top of the process which legitimately changes a file. Because the user knows the file is being changed, they will be less likely to suspect the changes also represent an infection. By its nature (and because executable code is not usually changed) a slow infector does not spread rapidly and if the integrity checker has a scanning component it will likely be caught. Also, an integrity checker that is run on a computer booted from a known-clean floppy disk will be able to defeat a slow infector.

Summary

- A fast infector infects programs when they are accessed, not just when run. This type of virus is designed to ride on the back of anti-virus scanners and can quickly infect an entire disk if not found before the scan is performed.
- A slow infector infects programs only when they are created or modified. This type of virus is designed to defeat integrity checkers but can usually be found if the checker has a scanner component or is started properly.

Multipartite Viruses

> **Multipartite viruses have a dual personality. Some are file viruses that can infect system sectors; others are system sector infectors that can infect files.**

Some viruses can be all things to all machines. Depending on what needs to be infected, they can infect system sectors or they can infect files. These rather universal viruses are termed multipartite (multi-part).

Sometimes the multipartite virus drops a system sector infector; other times a system sector infector might also infect files.

Multipartite viruses are particularly nasty because of the number of ways they can spread. Fortunately, a good one is hard to write.

Summary

- Multipartite viruses have dual capabilities and typically infect both system sectors and files.

Camouflage Viruses

> **When scanners were less sophisticated it might have been possible for a virus to sneak by as scanners sometimes did not display some alarms, knowing them to be false. This type of virus would be extremely hard to write today.**

You don't hear much about this type of virus. Fortunately it is rare and, because of the way anti-virus programs have evolved, is unlikely to occur in the future.

When anti-virus scanners were based completely on signatures there was always the possibility of a false alarm when the signature was found in some uninfected file (a statistical possibility). Further, with several scanners circulating, each had their own signature database and when scanned by another product may indicate infection where there was none simply because of the inclusion of the virus identification string. If this happened often, the public would get understandably annoyed (and frightened). In response, a scanner might therefore implement logic that, under the right circumstances, would ignore a virus signature and not issue an alarm.

While this "skip it" logic would stop the false alarms, it opened a door for virus writers to attempt to camouflage their viruses so that they included the specific characteristics the anti-virus programs were checking for and thus have the anti-virus program ignore that particular virus. Fortunately, this never became a serious threat; but the possibility existed.

Today's scanners do much more than simply look for a virus signature string. In order to identify the specific virus variant they also check the virus code and even checksum the virus code to identify it. With these cross-checks it would be extremely difficult for a virus to camouflage itself and spoof a scanner.

Summary

- In the past it was possible for a virus to spoof a scanner by camouflaging itself to look like something the scanner was programmed to ignore.
- Because of scanner technology evolution this type of virus would be very difficult to write today.

Virus Droppers

> **A dropper is a program that, when run will attempt to install a regular virus onto your hard disk.**

Normally, you obtain a virus by either attempting to boot from an infected floppy disk, by running an infected file, or by loading an infected document with viral macro commands in it. There is another way you can pick up a virus: by encountering a virus dropper. These are rare, but now and again someone will attempt to be clever and try to program one.

Basically, a dropper is just what the name implies: a program designed to run and install (or "drop") a virus onto your system. The program itself is not infected nor is it a virus because it does not replicate. So, technically, a dropper should be considered a Trojan. Often, because the virus is hidden in the program code, a scanner will not detect the danger until after the virus is dropped onto your system. (It's technically possible to write a virus that also drops other viruses, and several have been tried. Most are very buggy, however.)

It's a technical point, but there is a class of dropper that only infects the computer's memory, not the disk. These are given the name injector by some virus researchers.

Summary

- A Trojan program that installs a virus onto your system is called a dropper.
- Fortunately, because of technical difficulties, droppers are hard to program and therefore rare.

That's it for the discussion of virus types. Before going on to protection let's take an interesting detour...

3.Virus Protection

Finding a virus on your system may not be easy; they often don't cooperate. Using anti-virus tools is important.

A virus may or may not present itself. Viruses attempt to spread before activating whatever malicious activity they may have been programmed to deliver. So, viruses will often try to hide themselves. Sometimes there are symptoms that can be observed by a trained casual observer who knows what to look for (but, don't count on it).

Virus authors often place a wide variety of indicators into their viruses (e.g., messages, music, graphic displays). These, however, typically only show up when the virus payload activates. With DOS systems, the unaccounted for reduction of the amount of RAM known to be in the computer is an important indicator resident viruses have a hard time getting around. But, under Windows, there is no clear indicator like that. The bottom line is that one must use anti-virus software to detect (and fix) most viruses.

Your main defense is to detect and identify specific virus attacks to your computer. There are three methods in general use. Each has pros and cons and are discussed via these links. Often, a given anti-virus software program will use some combination of the three techniques for maximum possibility of detection.

- Scanning
- Integrity Checking
- Interception

In a more general sense, check here for some ideas about using the above-referenced methods and other useful information:

- AV Product Use Guidelines
- File Extensions
- Safe Computing Practices (Safe Hex)
- Outlook and Outlook Express
- Disable Scripting
- Backup Strategy

Another line of defense is continuing education. Click below to see some sources of on-going information.

- On-going Virus Information

Summary

- Viruses, by design, are hard to find using standard tools. SCANDISK and MEM can help, but don't rely on them to find viruses and never rely on DOS commands to eliminate a virus.
- Anti-virus software helps using techniques of:
    - Scanning
    - Interception
    - Integrity Checking
- You can help by taking some common sense precautions and keeping educated

Scanning

Scanning looks for known viruses by a signature or characteristics that make new viruses similar to existing viruses. This requires that anti-virus makers and users keep products up to date.

Once a virus has been detected, it is possible to write scanning programs that look for telltale code (signature strings) characteristic of the virus. The writers of the scanner extract identifying strings from the virus. The scanner uses these signature strings to search memory, files, and system sectors. If the scanner finds a match, it announces that it has found a virus. This obviously detects only known, pre-existing, viruses. Many so-called "virus writers" create "new" viruses by modifying existing viruses. This takes only a few minutes but creates what appears to be a new virus. It happens all too often that these changes are simply to fool the scanners. (Please use the above as "concept" information. Writing a scanner today is quite a bit more complex.)

Note: Newer scanners often employ several detection techniques in addition to signature recognition. Among the most common of these is a form of code analysis. The scanner will actually examine the code at various locations in an executable file and look for code characteristic of a virus (e.g., a jump to a non-standard location, etc.). A second possibility is that the scanner will set up a virtual computer in RAM and actually test programs by running them in this virtual space and observing what they do. These techniques are often lumped under the general name "heuristic" scanning. Such scanners may also key off of code fragments that appear similar to, but not exactly the same as, known viruses.

The major advantage of scanners is that they allow you to check programs before they are executed. Scanners provide the easiest way to check new software for known or suspected viruses. Since they have been aggressively marketed and since they provide what appears to be a simple painless solution to viruses, scanners are the most widely-used anti-virus product.

Too many people seem to regard "anti-virus product" and "scanner" as synonymous terms. The peril here is that if too many people depend solely upon scanners, newly created viruses will spread totally unhindered causing considerable damage before the scanners catch up with the viruses. An example of this was the attack by the Maltese Amoeba (Irish) virus in the UK. This virus was not detected prior to its destructive activation on November 1, 1991. Prior to its attack, it had managed to spread quite widely and none of the existing (mostly scanner-based) products detected this virus.

According to the December 1991 Virus Bulletin:

> *Prior to November 2nd, 1991, no commercial or shareware scanner (of which VB has copies) detected the Maltese Amoeba virus. Tests showed that not ONE of the major commercial scanners in use ... detected this virus.*

This indicates the potential hazard of depending upon scanner technology for complete virus protection. (More recent examples have been fast-spreading viruses that also act like worms [e.g., Melissa]. Anti-virus software makers react rapidly to these threats but there is still some delay and users have to be constantly alert.)

Another major drawback to scanners is that it's dangerous to depend upon an old scanner. With the dramatic increase in the number of viruses appearing, it's risky to depend upon anything other than the most current scanner. Even that scanner is necessarily a step behind the latest crop of viruses since there's a lot that has to happen before the scanner is ready:

- The virus has to be detected somehow to begin with. Since the existing scanners won't detect the new virus, it will have some time to spread before someone detects it by other means.
- The newly-discovered virus must be sent to programmers to analyze and extract a suitable signature string or detection algorithm. This must then be tested for false positives on legitimate programs.
- The "string" must then be incorporated into the next release of the virus scanner.
- The virus scanner or detection database must be distributed to the customer.

In the case of retail software, the software must be sent to be packaged, to the distributors, and then on to the retail outlets. Commercial retail software takes so long to get to the shelves, that it is almost certainly out of date. Virtually all product makers today provide some way to obtain updates via the Internet in order to help speed up the update process.

If you depend upon a scanner, be sure to get the latest version directly from the maker. Also, be sure that you boot from a clean write-protected copy of DOS before running the scanner for the first time at least; there's a good chance that the scanner can detect a resident virus in memory, but if it misses the virus in memory, the scanner will wind up spreading the virus rather than detecting it. Every susceptible program on your disk could be infected in a matter of minutes this way! (See Fast and Slow Infectors.)

Ghost Positives

One possible defect of scanners you might run into are termed "ghost" positives.

When DOS/Windows reads from a disk it does not read exactly what is requested; it also reads a bit ahead so that when the next read request comes in DOS may just have the material needed in a memory buffer and it can be provided much faster. Likewise, when a scanner reads files it has to compare each with the detection database. These are stored in memory.

If, after scanning, the scanner does not clear its buffers in memory and you immediately run a second scanner then the second scanner may see the first scanner's strings in memory and if it uses the same string(s) could identify that virus as being in memory.

This is why it's important to run your scanner (or other anti-virus product) after a cold boot. One of the features of a cold boot is a complete memory check and this check overwrites all of memory, clearing out all false traces of viruses.

False Alarms

Despite the most extensive testing it is possible that a scanner will present false alarms (i.e., indicate a file as infected when it really is not). You will usually note this just after an update where a file you've had on your system suddenly shows up as infected. If it's a single file, previously clean, that exhibits this characteristic you can rest a bit easier; but you should nevertheless check with your anti-virus software maker.

Testing a Scanner

You don't need a virus to test the installation of a scanner. Most good scanners today are programmed to detect a standard test file called the EICAR test file. You can easily make this test file. Simply type or copy the following string into a text editor like Notepad:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-
FILE!$H+H*
```

Now save that file under the name EICAR.COM. This is an actual program that, when run, will display the text EICAR-STANDARD-ANTIVIRUS-TEST-FILE! and, when scanned, should activate your anti-virus program.

Note: This is **not** a virus. It is simply a file designed to activate the detection routines in scanners that support it. (Some suggest you need a "good" virus to test scanners. The problem is that to adequately test a scanner you need a virus "zoo" and have to install each virus in the zoo and test against it. This is something few users would want to do. The EICAR test file tests the installation of anti-virus software and that should be sufficient.)

Summary

- Scanning depends on prior knowledge of a virus in order to detect it. This is done by recognizing some sort of signature that represents the virus or some program characteristic that indicates a virus may be present.
- Scanners allow you to check programs before execution. That is their main advantage.
- Scanners need to be regularly updated. Don't depend on an old scanner.
- Some viruses attempt to defeat scanners by changing their code on the fly. Current scanners attempt to analyze code on the fly as a way of countering this.
- Never run two scanners in a row without cold booting to clear memory between. If you do, you may find "ghost" positives.

Integrity Checking

> **Integrity products record information about your system for later comparison in order to detect changes. Just detecting changes is not enough, however; the detection must have some "intelligence" behind it to avoid confusion.**

Integrity checking products work by reading your entire disk and recording integrity data that acts as a signature for the files and system sectors. An integrity check program with built-in intelligence is the only solution that can handle **all** the threats to your data as well as viruses. Integrity checkers also provide the only reliable way to discover what damage a virus has done.

So, why isn't everyone using an integrity checker? In fact, many anti-virus products now incorporate integrity checking techniques. One problem with many products is that they don't use these techniques in a comprehensive way. There are still too many things not being checked.

Some older integrity checkers were simply too slow or hard to use to be truly effective. A disadvantage of a bare-bones integrity checker is that it can't differentiate file corruption caused by a bug from corruption caused by a virus. Advanced integrity checkers that incorporate the capability to analyze the nature of the changes and recognize changes caused by a virus have become available. Some integrity checkers now use other anti-virus techniques along with integrity checking to improve their intelligence and ease of use.

If you choose an integrity checker, be sure it has all these features:

- It's easy to use with clear, unambiguous reports and built-in help.
- It hides complexity, so that complicated details of system file or system sector changes are only presented if they contain information the user must act upon.
- The product recognizes the various files on the PC so it can alert the user with special warnings if vital files have changed.
- It's fast. An integrity checker is of no use if it's too slow.
- It recognizes known viruses, so the user doesn't have to do all the work to determine if a change is due to a software conflict, or if it's due to a virus. This also helps protect the integrity checker against attacks by viruses directed at it.
- It's important that the integrity computation be more sophisticated than a mere checksum. Two sectors may get reversed in a file or other damage may occur that otherwise rearranges data in a file. A simple checksum will not detect these changes. A cryptographic computation technique is best.
- It's comprehensive. Some integrity checkers, in order to improve their speed, don't read each file in its entirety. They read only portions of larger files. They just spot check. This is unacceptable; it's important to know the file hasn't changed, not just that some of the file hasn't changed.
- It checks and restores both boot and partition sectors. Some programs check only files.
- For protection, it should have safety features built in (e.g., ability to define the signature information file name and store the information on a floppy disks).

While using an integrity checker is an excellent way to monitor changes to your system, with today's operating systems so many files change on a regular basis it's imperative that you also use a good up-to-date scanner along with the integrity checker or for the integrity checker to have that capability built in.

Summary

- Integrity checking products read the disk and create signature information to determine changes.
- Coupled with virus identification, using integrity checking should be able to detect most any virus with the bonus of also detecting data corruption.

Interception

Monitoring for system-level routines that perform destructive acts can help, but such monitoring is fairly easily bypassed. Do not depend on it alone.

Interceptors (also known as resident monitors) are particularly useful for deflecting logic bombs and Trojans. The interceptor monitors operating system requests that write to disk or do other things that the program considers threatening (such as installing itself as a resident program). If it finds such a request, the interceptor generally pops up and asks you if you want to allow the request to continue. There is, however, no reliable way to intercept direct branches into low level code or to intercept direct input and output instructions done by the virus itself. Some viruses even manage to disable the monitoring program itself. Indeed, for one widely-distributed anti-virus program several years back it only took eight bytes of code to turn its monitoring functions off.

It is important to realize that monitoring is a risky technique. Some products that use this technique are so annoying to use (due to their frequent messages popping up) that some users consider the cure worse than the disease!

Summary

- Interceptors are useful for some simple logic bombs and Trojans.
- It would be unwise to depend entirely upon behavior monitors as they are easily bypassed.

4. Best AV product:

AV Product Use Guidelines

First, understand how your anti-virus product works. Then, start with a known-clean computer and follow specific steps to assure good virus detection/protection. Do research on specific products before purchase.

Most modern anti-virus products use a combination of techniques. However, they still get almost all of their protection from their scanner component. It's vital to understand exactly how your product works so that you understand what type of protection you really have (you might want to review the comments about scanning, interception, and integrity checking on other tutorial pages). Here are some rules that will help you make sure that you get maximum protection out of whatever product you have:

- First, you should check your computer's setup information to make certain that the boot sequence starts with the floppy drive. If you don't, and it starts with the hard drive then any boot sector virus on your computer will gain control before you run the anti-virus program(s). To get to the BIOS setup you will typically have to press a key or keystroke combination during the time the BIOS is checking the computer's memory. Once in setup you can check the boot sequence (one of the techniques used to protect against boot sector viruses on floppy disks is to set the boot sequence to check the hard drive first--but if this is set then you won't be able to boot from a clean floppy as indicated below; thus, this check).
- Be sure to cold boot your PC from a write-protected diskette before virus checking, particularly if you suspect you have a virus. Most anti-virus products make this recommendation, but this rarely gets done because the recommendation is often buried in some obscure location in the documentation. If your PC's memory is infected with a virus that your scanner does not recognize, you could infect all the programs on your disk if you do not boot from a clean disk. Don't take this chance; boot from a write-protected diskette before you scan. (In some cases, the AV product might come with a bootable CD-ROM instead. If so, then set the BIOS default to boot from the CD and use that disc.)
- If you are using a product which depends mostly on its scanner component, make sure that you always have the latest version. Scanners are often frequently updated (one AV program vendor says they update files on the Internet hourly if needed).
- Before you execute or install any new software, check it first (yes, commercial software has come from the factory infected). If it comes with an install program, check again after you install the software; an install program will frequently change or decompress executable

programs. After you first execute brand new software do an additional check of your system to make sure everything is as it should be.
- If your product contains a scanner component, check all diskettes brought in from another location; even data diskettes! Inevitably someone will leave a data diskette in their A: drive, potentially spreading a boot sector virus if the diskette is infected (assuming you have not reset the boot sequence back to booting from the hard disk first).
- If the anti-virus software has a component that installs under Windows in order to scan all files before they are opened by all means install that component. This is a valuable service that is well worth the small amount of slowdown and resource use you will experience.

What's the best anti-virus product?

The simple answer is that there is no definite answer to the question! For one thing, a "good" anti-virus product integrates well with your particular system and system setup. If you are on a network with diskless workstations, for example, you might want to install the anti-virus software on the server. If you don't regularly exchange or download files you might find a less intrusive anti-virus product more to your liking. And so on.

Relying on magazine articles is also not the best way to decide upon an anti-virus product. Valid testing requires special setups to make certain products are being tested against real viruses under conditions those viruses might be found (e.g., it would not be a particularly useful test to place boot sector viruses into zip archives and then testing an anti-virus product against that archive).

One measure of anti-virus software is ICSA approval. To obtain this approval a scanner must detect all viruses on the current version of the Wild List in addition to 90% of the full NCSA test suite. You can obtain more information about this at: http://www.icsa.net/services/product_cert/

If you want to try an anti-virus product, many producers have evaluation versions at their web site.

Summary

- Understand your anti-virus product and what you can expect from it.
- Check setup to be certain you are booting from the floppy disk and then cold boot from a known-clean, write-protected diskette.
- Scan only with the latest version of any scanner.
- Check all new software and all data diskettes before use and again after the installation.
- Install any scan-on-use component your anti-virus product may have.
- Do a bit of research and look for certification when you purchase anti-virus software.

Disable Scripting

The Windows Scripting Host is used by few but makes many avenues of mischief available to malicious software. Consider removing or deactivating it.

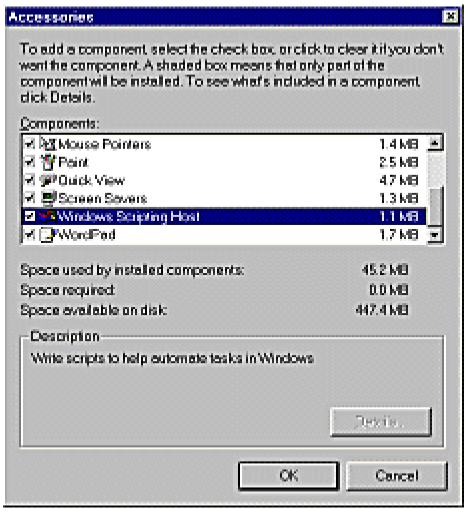
In order to run VisualBasic Scripts (VBS files) on your computer you must have the Windows Scripting Host (WSH) installed and working on your computer. While scripting allows you to closely integrate some application software, it also allows worms such as LoveLetter (as one example) to use your copy of Outlook to send itself to all the people in your address book (and other malicious things!).

In order to avoid these sorts of attacks it's often best to just disable the Windows Scripting Host. Most people don't need/use it. Following are instructions for removing WHS.

Windows98

Typically, WSH is installed if you choose a standard install of the OS, if you install the IE5 browser, or if you directly install WSH from Microsoft. To turn it off...

- Open the Add/Remove Control Panel application. Either "Start | Settings | Control Panel" or double click "My Computer" and "Control Panel" then double click "Add/Remove Programs."
- Click on the "Windows Setup" tab.
- Scroll to "Accessories" and double click that entry. An accessories windows that looks like the following should open...



Accessories

To add a component, select the check box, or click to clear it if you don't want the component. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Components:

☑	Mouse Pointers	1.4 MB
☑	Paint	2.5 MB
☑	Quick View	4.7 MB
☑	Screen Savers	1.3 MB
☑	Windows Scripting Host	1.1 MB
☑	WordPad	1.7 MB

Space used by installed components:	45.2 MB
Space required:	0.0 MB
Space available on disk:	447.4 MB

Description

Write scripts to help automate tasks in Windows

- Scroll down the accessories list until you find "Windows Scripting Host" and then click on the checkbox next to the entry to deselect it (i.e., remove the check mark in the box).

- Click OK to close the window(s) and OK again to close the "Add/Remove Programs" window.

(Windows 98 is the only OS Computer Knowledge has tried this process on. Following are brief instructions believed to work for other operating systems.)

Windows95

Basically, you have WSH installed if you've installed the IE5 browser or WSH itself. In order to stop it from running you have to disassociate the VBS extension with the WSH. Right click "My Computer" on the Desktop or in Windows Explorer. Select "Open." Click on the "View" menu and select "Options...." Now click on the "File Types" tab. Scroll down to "VBScript Script File" (if not found stop here and cancel out; you don't have scripting active). Click on the "VBScript Script File" and select "Remove." Confirm and then quit the File Types application.

WindowsNT 4.0

Basically, you have WSH installed if you've installed the IE5 browser or WSH itself. In order to stop it from running you have to disassociate the VBS extension with the WSH. Log on as an administrator. Right click "My Computer" on the Desktop or in Windows Explorer. Select "Open." Click on the "View" menu and select "Options...." Now click on the "File Types" tab. Scroll down to "VBScript Script File" (if not found stop here and cancel out; you don't have scripting active). Click on the "VBScript Script File" and select "Remove." Confirm and then quit the File Types application.

Windows 2000

WSH is normally installed. In order to stop it from running you have to disassociate the VBS extension with the WSH. Log on as an administrator. Right click "My Computer" on the Desktop or in Windows Explorer. Select "Open." Click on the "View" menu and select "Options...." Now click on the "File Types" tab. Scroll down to "VBScript Script File" (if not found stop here and cancel out; you don't have scripting active). Click on the "VBScript Script File" and select "Remove." Confirm and then quit the File Types application.

5. CONCLUSION :

Mostly I conclude our antivirus is important because viruses are increasing day by day.

First, understand how your anti-virus product works. Then ,starts with a known-clean computer and follow

Specific steps to assure good virus detection/protection.